# 埋め込みモデルベースの教師なしキーフレーズ抽出における長文に対する抽出精度の改善

藤原知樹
株式会社 ABEJA
tomoki.fujihara@abejainc.com

## 概要

キーフレーズ抽出は情報検索や要約に活用できる技術である．本稿の目的は埋め込みモデルを用いた教師なしキーフレーズ抽出における長文に対する抽出精度の改善である．埋め込みモデルベースの手法では，文章全体とフレーズとの類似度を直接算出するため，長文においてテキスト長の大きな差が原因で精度が低下しやすい．提案手法では文単位の埋め込みを用いて重要な文を絞り込むことで，既存手法に比べ長文へのキーフレーズ抽出精度が改善した．開発したキーフレーズ抽出ツールおよび日本語の評価データセットは https://github.com/flatton/keyphrase_extraction_tools にて公開予定である．

## 1 はじめに

キーワード抽出とは文章中での重要な単語を抽出する技術であり，情報検索や要約などに用いられる．その中でも，キーフレーズ抽出は単語単位ではなくフレーズ単位で重要な語句を抽出することを目的としている．キーフレーズ抽出手法には教師あり・教師なしのアプローチが存在するが，教師ありのアプローチは学習データを作成するコストがかかるため，より汎用的な教師なしのアプローチが研究されている．

従来手法としてはグラフベース [1, 2, 3, 4]，統計ベース [5, 6] のアプローチが提案されている．近年は大規模な言語モデルの発展に伴い，埋め込みモデルや生成モデルを用いた手法も研究されている．生成モデルを用いたアプローチではキーワード抽出だけでなくキーワード生成の研究 [7] も行われており，文章の内容を解釈して文章中に無いより簡潔で適切なキーフレーズを抽出することも可能である．

埋め込みモデルを用いたアプローチでは文章とフレーズの類似度を重要度として用いる．埋め込みモデルのアプローチは長文に対する抽出性能が低く，その原因として長文とフレーズではテキストの長さが大きく異なることが挙げられる [8, 9]．これに対して，フレーズの埋め込み方法を工夫する手法 [10, 11] がいくつか提案されているが，手法間で埋め込みモデルが異なる，出現位置や出現頻度による重みづけを独自に導入しているなどの細かな差異があり，埋め込み方法の違いによる効果は明確でない．

本稿では埋め込みモデルベースの教師なしキーフレーズ抽出における長文に対する抽出精度の改善を目的とし，文単位の埋め込みを用いて重要な文を絞り込む手法を提案する．提案手法と既存の長文対策の手法を比較した結果，既存手法に比べて長文への抽出精度が改善した．一方で，グラフベース，統計ベース，生成モデルベースの教師なしキーフレーズ抽出手法と比較した結果，短文への抽出精度は優れているが長文への抽出精度には依然課題が残るという結果になった．

## 2 教師なしキーフレーズ抽出

教師なしキーフレーズ抽出の手法としては主にグラフベース，統計ベース，埋め込みモデルベース，生成モデルベースのアプローチがある．キーフレーズ抽出手法間の主な違いは候補フレーズの選び方とキーフレーズの重要度の推定方法にある．グラフベース，統計ベース，埋め込みモデルベースのアプローチでは，通常キーフレーズの候補をあらかじめ作成する必要があり，主に品詞解析に基づく方法か n-gram が用いられている．生成モデルベースのアプローチの場合は事前にフレーズ候補を作成せずとも抽出可能である．以下，各種アプローチのキーフレーズ判定方法について説明する．

### 2.1 グラフベース

グラフベースのアプローチでは単語やフレーズなどをノード，共起関係をエッジとしたグラフを形成し，グラフに埋め込まれた単語間，フレーズ間の関係性から重要度を推定する．TextRank [1], SingleRank[2] では，トークン化された単語の系列に対して品詞やストップワードに基づきキーワード候補を選定し，出現位置に基づく共起関係からグラフを形成して PageRank [12] のアルゴリズムで重要度を算出する．また，TopicRank [3]，MultipartiteRank [4] では，品詞の並びをもとにキーフレーズ候補を選定し，さらにフレーズ間での語幹の一致率をもとにグループ分けすることでキーフレーズの多様性を高めている．

### 2.2 統計ベース

統計ベースの手法では単語・フレーズの出現回数や tf-idf 値などを用いて重要度を算出する．KP-Miner [5] は tf-idf，最初の出現位置，フレーズ中の単語数を特徴量として用いて重要度を算出する．出力されるキーフレーズはストップワードの単語を含まない n-gram である． YAKE! [6] は大文字小文字，出現位置，出現頻度，その単語と共起する単語の種類数，その単語が含まれる文の数を特徴量に用いる．出力されるキーフレーズは空白や特殊文字を含まない n-gram である．

### 2.3 生成モデルベース

生成モデルベースの手法ではフレーズの候補および重要度はモデル内部で暗黙的に計算される．生成によりキーフレーズを取得するため，指定したフレーズのリストからの選択させる，自動的に正規化されたキーフレーズを抽出する，事前知識をもとに文中にないキーフレーズを生成するといった利用が可能である [7].

### 2.4 埋め込みモデルベース

埋め込みモデルベースの手法では主にフレーズと文章の類似度を用いて重要度を算出する．ただし，類似度のみでは似た意味のキーフレーズが多く抽出されるという課題があり，EmbedRank++ [13] ではフレーズと文章の埋め込みの類似度でのランキングに加えて，Maximum Marginal Relevance (; MMR) [14] などのリランキングアルゴリズムによるキーフレーズの多様化を取り入れている．MMR などのリランキングアルゴリズムをサポートしたオープンソースなツールとして KeyBERT [15] がある．

もう一つの課題として，埋め込みベースのアプローチは長文に対する抽出精度が低い点が挙げられる [8, 9]． SIFRank [16] ではフレーズの出現位置や出現頻度による重みづけを行うことで抽出精度を改善している． AttentionRank [17] ではフレーズ自身の Self-Attention およびフレーズと文章の Cross-Attention から関連度を算出することで，SIFRank よりも長文に対して高い性能を示している． MDERank [10] では重要なフレーズほど文章から削除されるとその文章の意味に大きな影響を与えるという仮定のもと，フレーズ部分をマスクした文章と元の文章の距離を用いて重要度を算出している． PromptRank [11] では Encoder-Decoder モデルに文章のメイントピックを予測させるプロンプトを入力し，候補フレーズの生成確率を用いて重要度を算出している．

## 3 提案手法: 文フィルタリング

埋め込みモデルベースのアプローチが長文での性能が低下する原因として，フレーズと長文ではテキスト長が大きく異なるため，両者の類似度を正確に算出するのが困難な点が挙げられる [10, 11]．そこで提案手法では，文章とフレーズの類似度を算出する代わりに，文章と文，文とフレーズの類似度をそれぞれ算出することで，テキスト間での文字列の長さやコンテキストの差を小さくする．

提案手法では，まず文と文章の類似度を算出し，類似度の高い文を選出することで重要な文を抽出する．次に抽出された各文中のフレーズと各文の類似度を算出する．最後に文と文章の類似度およびフレーズと文の類似度からハイブリッドなランキングを作成し，上位のフレーズをキーフレーズとして出力する．文フィルタリングの処理は MMR や MEDRank などの既存手法と組み合わせることができる．

## 4 実験

埋め込みベースの教師なしキーフレーズ抽出の既存手法および提案手法について，抽出精度を比較した．また，グラフベース，統計ベース，生成モデルベースの教師なし抽出手法と抽出精度および処理速度を比較した．

### 4.1 実験条件

埋め込みモデルには `cl-nagoya/ruri-small`, `cl-nagoya/ruri-base`, および `cl-nagoya/ruri-large` [18] を用いた．グラフベース，統計ベースのモデルにはオープンソースのキーフレーズ抽出ツール PKE[19] で実装されている TextRank，SingleRank，TopicRank，MultipartiteRank，TfIdf，KPMiner，および YAKE を用いた．生成モデルベースの抽出では，Azure OpenAI の GPT4o (モデルバージョン：2024-11-20) を用いた．生成モデルベース以外の手法ではテキストを分ち書きする必要があるため GiNZA[20] を用いた．実験はすべてメモリ 16 GB，チップ Apple M2 の MacBook Pro 上で実施しており，埋め込みモデルによる encode 処理は `device="mps"` の条件で実行した．

### 4.2 評価データセット

評価データセットは ABEJA-CC-JA[21] からサンプリングした文字数約 200，2,000，20,000 文字の文章各 33 種類に著者がアノテーションしたものを用いた．アノテーション方式としては，文章をすべて読み，その文章の内容を把握する上で重要になる，または特徴的な語句や固有名詞などをキーフレーズとしてラベル付けした．また，別名など同じ意味で表記の異なるフレーズが存在する場合に，それぞれを独立したキーフレーズにすると正解ラベルを不用意に増やす恐れがあり，逆に一つのフレーズだけを正解とすると偏った評価になる可能性があると考えた．そのため，各テキストにはキーフレーズ集合のリストという形式でアノテーションを行い，各キーフレーズ集合は同じ意味のフレーズで構成され，その集合のリストをそのテキストの正解キーフレーズ群とした．テキストと正解ラベルのサンプルは付録 A に記載する．

### 4.3 評価指標

任意の長さの文字列全体の集合を $\mathcal{W}$ として，正解のキーフレーズ集合のリストを $\mathcal{Y} = \{Y_1, Y_2, \ldots, Y_m\}, Y_i \subset \mathcal{W}$ 予測キーフレーズの上位 $k$ 個を $\hat{Y}_k = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_k\}, \hat{Y}_k \subset \mathcal{W}$ として，次の 2 つの評価指標で抽出性能を評価した．

$$\text{Precision@k} = \frac{\left|\{Y_i \in \mathcal{Y} | Y_i \cap \hat{Y}_k \neq \varnothing\}\right|}{k}. \quad (1)$$

$$\text{Recall@k} = \frac{\left|\{Y_i \in \mathcal{Y} | Y_i \cap \hat{Y}_k \neq \varnothing\}\right|}{m}. \quad (2)$$

### 4.4 埋め込みベースのキーフレーズ抽出手法の比較

埋め込みベースのアプローチにおける既存手法と提案手法の効果を検証する（表 1）．200，2,000，20,000 文字のデータセット毎にスコアの平均値 ± 標準偏差を記載している．埋め込みモデルには `cl-nagoya/ruri-base` を用いた．

候補フレーズの選び方を比較すると，KeyBERT で用いられている n-gram に比べて，TopicRank などで用いられている品詞規則に基づく方法（grammar）の方がいずれの条件でも Precision，Recall が高い．評価方法として，文字列を正規化した上で正解のキーフレーズと完全一致した予測フレーズのみ正解とカウントされるため，n-gram ベースの抽出では文字の欠損や余分な文字を含めてしまうことがあり，スコアが低くなったと考えられる．

次に，grammar 条件を基準に，長文対策のアプローチとして MDERank で提案されているフレーズをマスクしたテキストをクエリとする手法（Mask），PromptRank のようにクエリプロンプトにコンテキストを追加する手法（Prompt），提案手法の文フィルタリング（Filter）を比較する．いずれの手法も 20,000 文字の文章に対してはスコアが向上しているが，200，2,000 文字の文章ではスコアが低下している． Filter w/ $min$ = 100 の条件では文単位に分割する際に，各文が 100 文字を超えるように文同士を結合して一つの文・チャンクとして扱う処理を施していた．これにより，20,000 文字の文章に対する抽出精度をさらに向上しつつ，200，2,000 文字の文章に対するスコアの低下を緩和できている． 20,000 文字の文章に対する処理速度は grammar 条件 0.10[$s$]，grammar + Filter w/ $min$ = 100 条件で 0.28[$s$] と約 3 倍に増加している．

### 4.5 各種教師なしキーフレーズ抽出手法との比較

各種教師なしキーフレーズ抽出手法との比較を行う（表 2）．埋め込みモデルベースの手法では grammar + Filter w/ $min$ = 100 の条件で固定し，埋め込みモデルのみ変更してモデルサイズの影響を分析する．抽出精度については GPT4o を用いた生成モデルベースの手法が最も高くなった．埋め込みモデルベースの手法は 200 文字での抽出精度はグラフ

**表 1**　埋め込みベースのキーフレーズ抽出手法の比較

| @k | 条件 | 200 文字 | | 2,000 文字 | | 20,000 文字 | |
|---|---|---|---|---|---|---|---|
| | | Precision (↑) | Recall (↑) | Precision (↑) | Recall (↑) | Precision (↑) | Recall (↑) |
| @5 | n-gram (1, 4) | 0.12 ± 0.11 | 0.07 ± 0.07 | 0.04 ± 0.08 | 0.02 ± 0.04 | 0.04 ± 0.08 | 0.01 ± 0.03 |
| | grammar | **0.61 ± 0.21** | **0.39 ± 0.12** | 0.44 ± 0.22 | **0.20 ± 0.10** | 0.17 ± 0.16 | 0.06 ± 0.06 |
| | + Mask | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 |
| | + Prompt | 0.53 ± 0.22 | 0.34 ± 0.16 | 0.41 ± 0.19 | 0.18 ± 0.09 | 0.22 ± 0.18 | 0.08 ± 0.07 |
| | + Filter (Ours) | 0.58 ± 0.24 | 0.37 ± 0.14 | 0.42 ± 0.24 | 0.19 ± 0.12 | 0.25 ± 0.20 | 0.10 ± 0.09 |
| | + Filter w/ *min* = 100 (Ours) | **0.61 ± 0.24** | 0.38 ± 0.14 | **0.45 ± 0.19** | **0.20 ± 0.09** | **0.30 ± 0.21** | **0.12 ± 0.09** |
| @10 | n-gram (1, 4) | 0.09 ± 0.08 | 0.12 ± 0.10 | 0.05 ± 0.06 | 0.05 ± 0.06 | 0.02 ± 0.04 | 0.02 ± 0.03 |
| | grammar | **0.47 ± 0.16** | **0.59 ± 0.16** | **0.37 ± 0.14** | **0.33 ± 0.12** | 0.16 ± 0.10 | 0.12 ± 0.08 |
| | + Mask | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 |
| | + Prompt | 0.39 ± 0.16 | 0.50 ± 0.20 | 0.31 ± 0.13 | 0.28 ± 0.12 | 0.18 ± 0.13 | 0.14 ± 0.11 |
| | + Filter (Ours) | 0.42 ± 0.14 | 0.52 ± 0.17 | 0.36 ± 0.12 | **0.33 ± 0.13** | 0.20 ± 0.11 | 0.16 ± 0.10 |
| | + Filter w/ *min* = 100 (Ours) | 0.45 ± 0.12 | 0.57 ± 0.14 | **0.37 ± 0.12** | **0.33 ± 0.10** | **0.25 ± 0.13** | **0.20 ± 0.12** |
| @25 | n-gram (1, 4) | 0.07 ± 0.04 | 0.21 ± 0.11 | 0.04 ± 0.03 | 0.08 ± 0.07 | 0.02 ± 0.03 | 0.03 ± 0.06 |
| | grammar | **0.24 ± 0.06** | **0.75 ± 0.17** | 0.23 ± 0.07 | 0.52 ± 0.13 | 0.12 ± 0.06 | 0.24 ± 0.14 |
| | + Mask | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 |
| | + Prompt | **0.24 ± 0.06** | 0.74 ± 0.16 | 0.20 ± 0.07 | 0.43 ± 0.13 | 0.11 ± 0.06 | 0.21 ± 0.14 |
| | + Filter (Ours) | **0.24 ± 0.07** | 0.74 ± 0.17 | 0.22 ± 0.06 | 0.49 ± 0.14 | 0.14 ± 0.06 | 0.26 ± 0.13 |
| | + Filter w/ *min* = 100 (Ours) | **0.24 ± 0.06** | **0.75 ± 0.17** | **0.24 ± 0.07** | **0.54 ± 0.15** | **0.15 ± 0.07** | **0.30 ± 0.16** |

**表 2**　各種教師なしキーフレーズ抽出手法との比較

| @k | モデル | 200 文字 | | | 20,000 文字 | | |
|---|---|---|---|---|---|---|---|
| | | Precision (↑) | Recall (↑) | ProcessTime (↓) | Precision (↑) | Recall (↑) | ProcessTime (↓) |
| @10 | TextRank | 0.35 ± 0.16 | 0.44 ± 0.19 | **0.01 ± 0.00** | 0.06 ± 0.08 | 0.05 ± 0.06 | **0.05 ± 0.01** |
| | SingleRank | 0.34 ± 0.18 | 0.44 ± 0.22 | **0.01 ± 0.00** | 0.11 ± 0.10 | 0.08 ± 0.08 | **0.05 ± 0.01** |
| | TopicRank | 0.25 ± 0.15 | 0.31 ± 0.18 | **0.01 ± 0.00** | 0.27 ± 0.17 | 0.20 ± 0.12 | **0.05 ± 0.01** |
| | MultipartiteRank | 0.28 ± 0.18 | 0.35 ± 0.23 | **0.01 ± 0.00** | 0.30 ± 0.16 | 0.22 ± 0.12 | 0.06 ± 0.01 |
| | Tf-Idf | 0.18 ± 0.11 | 0.24 ± 0.14 | 0.02 ± 0.00 | 0.26 ± 0.15 | 0.19 ± 0.11 | 0.08 ± 0.01 |
| | KP-Miner | 0.04 ± 0.06 | 0.05 ± 0.10 | 0.02 ± 0.00 | 0.22 ± 0.13 | 0.17 ± 0.11 | 0.08 ± 0.01 |
| | YAKE! | 0.15 ± 0.13 | 0.19 ± 0.15 | **0.01 ± 0.00** | 0.24 ± 0.15 | 0.17 ± 0.10 | **0.05 ± 0.01** |
| | GPT4o | **0.52 ± 0.16** | **0.65 ± 0.19** | 0.12 ± 0.11 | **0.45 ± 0.16** | **0.34 ± 0.15** | 1.02 ± 0.09 |
| | Ours (Ruri-small) | 0.45 ± 0.12 | 0.57 ± 0.15 | 0.04 ± 0.02 | 0.22 ± 0.13 | 0.18 ± 0.13 | 0.26 ± 0.07 |
| | Ours (Ruri-base) | 0.45 ± 0.12 | 0.57 ± 0.14 | 0.04 ± 0.02 | 0.25 ± 0.13 | 0.20 ± 0.12 | 0.28 ± 0.05 |
| | Ours (Ruri-large) | 0.43 ± 0.12 | 0.54 ± 0.16 | 0.08 ± 0.04 | 0.27 ± 0.12 | 0.21 ± 0.11 | 0.86 ± 0.23 |

ベース，統計ベースのモデルよりも優れているが，20,000 文字での抽出精度は MultipartiteRank にやや劣っており，長文への抽出精度改善は依然課題である．また埋め込みモデルのモデルサイズを大きくすると長文での抽出精度が向上する傾向にある．これは大きなモデルほど長い文章全体の意味をより埋め込むことができているためと考えられる．

処理速度については文章の長さに関わらずグラフベース，統計ベースの抽出モデルが高速である．埋め込みモデルベースの手法はモデルサイズを大きくすると処理速度も大きくなる傾向があり，Ruri-large を用いた条件では GPT4o による抽出と同程度まで速度が低下している．埋め込みモデルベースの手法では文章が長くなると，トークン長や候補フレーズが増加するため処理速度が増加する．そのため，候補フレーズの絞り込みや重要度算出のアルゴリズムの改善により処理速度を改善することも今後の課題である．

## 5　おわりに

本稿では埋め込みモデルを用いた教師なしキーフレーズ抽出における長文に対する抽出精度の改善を目的に，文フィルタリングを提案した．実験の結果，提案手法を導入することで埋め込みモデルベースの既存手法に比べ長文への抽出精度が改善した．一方で，各種教師なしキーフレーズ抽出手法と比較すると，短文への抽出精度は優れているものの長文への抽出精度は依然課題であり，処理速度の改善も今後の課題である．

# 参考文献


[1] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In Dekang Lin and Dekai Wu, editors, **Proc. EMNLP 2004**, pp. 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[2] Xiaojun Wan and Jianguo Xiao. CollabRank: Towards a collaborative approach to single-document keyphrase extraction. In Donia Scott and Hans Uszkoreit, editors, **Proc. COLING 2008**, pp. 969–976, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

[3] A. Bougouin, F. Boudin, and B. Daille. TopicRank: Graph-based topic ranking for keyphrase extraction. In Ruslan Mitkov and Jong C. Park, editors, **Proc. IJCNLP 2013**, pp. 543–551, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing.

[4] Florian Boudin. Unsupervised keyphrase extraction with multipartite graphs. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, **Proc. NAACL 2018**, pp. 667–672, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[5] Samhaa R. El-Beltagy and Ahmed Rafea. KP-Miner: Participation in SemEval-2. In Katrin Erk and Carlo Strapparava, editors, **Proc. SemEval 2010**, pp. 190–193, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[6] Ricardo Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt. Yake! collection-independent automatic keyword extractor. In Gabriella Pasi, Benjamin Piwowarski, Leif Azzopardi, and Allan Hanbury, editors, **Advances in Information Retrieval**, pp. 806–810, Cham, 2018. Springer International Publishing.

[7] R. Martínez-Cruz, A. J. López-López, and J." Portela. Chatgpt vs state-of-the-art models: a benchmarking study in keyphrase generation task. **Applied Intelligence**, Vol. 55, No. 1, p. 50, 2024.

[8] Mingyang Song, Yi Feng, and Liping Jing. A survey on recent advances in keyphrase extraction from pre-trained language models. In Andreas Vlachos and Isabelle Augenstein, editors, **Findings of the Association for Computational Linguistics: EACL 2023**, pp. 2153–2164, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[9] N. Giarelis and N. Karacapilidis. Deep learning and embeddings-based approaches for keyphrase extraction: a literature review. **Knowledge and Information Systems**, Vol. 66, No. 11, pp. 6493–6526, 2024.

[10] L. Zhang, Q. Chen, W. Wang, C. Deng, S. Zhang, B. Li, W. Wang, and X. Cao. MDERank: A masked document embedding rank approach for unsupervised keyphrase extraction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, **Findings of the Association for Computational Linguistics: ACL 2022**, pp. 396–409, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[11] A. Kong, S. Zhao, H. Chen, Q. Li, Y. Qin, R. Sun, and X. Bai. PromptRank: Unsupervised keyphrase extraction using prompt. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Proc. ACL 2023**, pp. 9788–9801, Toronto, Canada, July 2023. Association for Computational Linguistics.

[12] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[13] K. Bennani-Smires, C. Musat, A. Hossmann, M. Baeriswyl, and M. Jaggi. Simple unsupervised keyphrase extraction using sentence embeddings. In Anna Korhonen and Ivan Titov, editors, **Proc. CoNLL 2018**, pp. 221–229, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[14] Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In **Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**, SIGIR '98, p. 335–336, New York, NY, USA, 1998. Association for Computing Machinery.

[15] Maarten Grootendorst. KeyBERT: Minimal keyword extraction with BERT., 2020.

[16] Y. Sun, H. Qiu, Y. Zheng, Z. Wang, and C. Zhang. SIFRank: A new baseline for unsupervised keyphrase extraction based on pre-trained language model. **IEEE Access**, Vol. 8, pp. 10896–10906, 2020.

[17] Haoran Ding and Xiao Luo. AttentionRank: Unsupervised keyphrase extraction using self and cross attentions. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, **Proc. EMNLP 2021**, pp. 1919–1928, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[18] Hayato Tsukagoshi and Ryohei Sasano. Ruri: Japanese General Text Embeddings, 2024.

[19] Florian Boudin. pke: an open source python-based keyphrase extraction toolkit. In Hideo Watanabe, editor, **Proc. COLING 2016**, pp. 69–73, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[20] Mai Hiroshi and Masayuki. 短単位品詞の用法曖昧性解決と依存関係ラベリングの同時学習. 言語処理学会第 25 回年次大会, 2019.

[21] ABEJA inc. ABEJA-CC-JA, 2024. Accessed: 2024-12-13.

# A　評価データセットのサンプル

評価データセット中の文章と正解ラベルのサンプルを以下に記載する．メイントピックに該当するキーフレーズ集合を main_topic: set[str]，話の視点や切り口に該当するキーフレーズ集合を angle: set[str]，その他のキーフレーズ集合のリストを essential_terms: list[set[str]] としてラベル付与している．評価時はメイントピックか視点かその他かといった区別をせずに，三種類のラベルをまとめて一つの正解キーフレーズ集合のリスト（list[set[str]]）として扱っている．

text

葉に濃淡のモザイク，えそ条斑，斑紋，萎縮，ねじれを生じ，花に斑入を生じる．\n インゲンマメ黄斑モザイクウイルス (BYMV)，キュウリモザイクウイルス (CMV) によるが，BYMV によるものが多い．BYMV はアプラムシによって非永続型伝搬され，汁液伝染もする．CMV についてはユリモザイク病の項参照．\n 各病原ウイルスの性質により，アプラムシ防除，消毒液による農具，種子，指の消毒，無病球根の導入などを行う．

label

```
{
  "main_topic": ["モザイクウイルス"],
  "angle": ["病原ウイルス"],
  "essential_terms": [
     ["インゲンマメ黄斑モザイクウイルス", "BYMV"],
    ["キュウリモザイクウイルス", "CMV"],
    ["ユリモザイク病"],
    ["アプラムシ"],
    ["非永続型伝搬"],
    ["汁液伝染"],
    ["アプラムシ防除"],
    ["無病球根"]
  ]
}
```