A Tree-based Unsupervised Keyphrase Extraction Technique

Hyoin An and Yongqi Liu

The Ohio State University

May 2, 2022

Introduction

- **Automatic keyphrase extraction** is to extract quality keyphrases for a summarization of a document [4].
- Automatic keyphrase extraction techniques are used in various settings related to digital information processing applications.

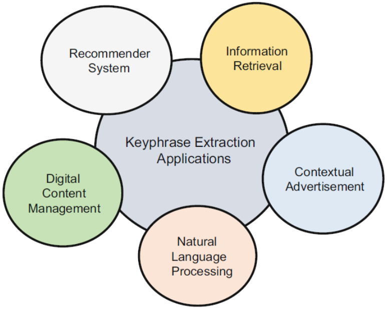

Recommender System
Information Retrieval
Keyphrase Extraction Applications
Digital Content Management
Contextual Advertisement
Natural Language Processing

Figure 1: Applications of keyphrase extraction (Figure 1 of Rabby et al. (2020) [4])

Key Results

- Utilizing these techniques could be challenging when a method is domain specific, a large amount of training data is required, and/or a method is computationally too extensive.
- Rabby et al. (2020) [4] propose **a new tree-based unsupervised method for keyphrase extraction (TeKET)**.
- A variant of tree, called Keyphrase Extraction (KePhEx) tree, is introduced which determines the final keyphrases from candidate keyphrases.
- Contributions:
  1. Propose a new keyphrase ranking approach
  2. Domain- and language-independent
  3. Require no training data

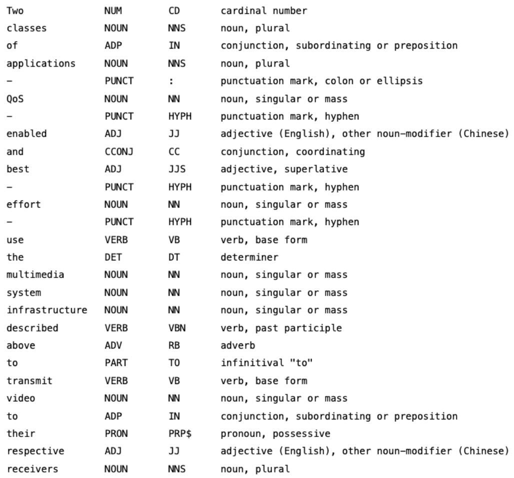
Pre-processing: POS Tagging

Part of Speech (POS) Tagging is the process of allocating the part of speech tag or other philological class sign to each and every word in a sentence [2].

(Example 1) "Two classes of applications - QoS-enabled and best-effort - use the multimedia system infrastructure described above to transmit video to their respective receivers."

The POS tagging can automatically classify each word to its class, like "applications" as noun and "use" as verb.

Pre-processing: POS Tagging



```
Two             NUM     CD      cardinal number
classes         NOUN    NNS     noun, plural
of              ADP     IN      conjunction, subordinating or preposition
applications    NOUN    NNS     noun, plural
–               PUNCT   :       punctuation mark, colon or ellipsis
QoS             NOUN    NN      noun, singular or mass
–               PUNCT   HYPH    punctuation mark, hyphen
enabled         ADJ     JJ      adjective (English), other noun-modifier (Chinese)
and             CCONJ   CC      conjunction, coordinating
best            ADJ     JJS     adjective, superlative
–               PUNCT   HYPH    punctuation mark, hyphen
effort          NOUN    NN      noun, singular or mass
–               PUNCT   HYPH    punctuation mark, hyphen
use             VERB    VB      verb, base form
the             DET     DT      determiner
multimedia      NOUN    NN      noun, singular or mass
system          NOUN    NN      noun, singular or mass
infrastructure  NOUN    NN      noun, singular or mass
described       VERB    VBN     verb, past participle
above           ADV     RB      adverb
to              PART    TO      infinitival "to"
transmit        VERB    VB      verb, base form
video           NOUN    NN      noun, singular or mass
to              ADP     IN      conjunction, subordinating or preposition
their           PRON    PRP$    pronoun, possessive
respective      ADJ     JJ      adjective (English), other noun-modifier (Chinese)
receivers       NOUN    NNS     noun, plural
```

Figure 2: POS tagging result of Example 1

Pre-processing: Terms

- **Noun phrases:** “Noun + Adjective” or “Adjective + Noun” form
- **Candidate keyphrases:** Noun phrases that are extracted from a document and used to construct KePhEx trees
- **Root word:** A (noun) word that is used as a root in a KePhEx tree
- **Similar keyphrases:** a collection of candidate keyphrases that share the same root word
- **TF-IDF:** Term Frequency (“aboutness”) - Inverse Document Frequency (“informativeness”)
- **Mu (μ):** Cohesiveness Index (with respect to the root)
- **Mamu:** Minimum allowable mu (μ) that is used to prune a tree

Pre-processing: Cleaning Process

Cleaning process: Any candidate keyphrase

1. that contains non-alphabetic characters
2. that contains single alphabetic word(s)
3. whose frequency is less than the *lsaf* factor*

is removed from the corpus.

**lsaf* factor = least seen allowable frequency factor

Stemming process: To incorporate different forms of the same stem of a word, we take the stemmed form of words.

(Example 2) manage, management $\rightarrow$ *manag*
autocorrelation $\rightarrow$ *autocorrel*

Tree-based KeyPhrase Extraction Technique (TeKET)

- A KeyPhrase Extraction (KePhEx) tree is grown for each root word.
- Similar keyphrases would be used to form and update the KePhEx tree.
- Each node in the tree has three attributes:
  1. Word (noun or adjective)
  2. TF-IDF value
  3. Mu (μ)
- TeKET is implemented in Python, and the authors' codes can be found in [3].

Step (1) Add Nodes

1. Choose a root word.
2. Go through the similar keyphrases one by one and see if a word in that phrase is qualified to be added into the tree as a new node.
3. Initialize the μ value of a new node to be 1.

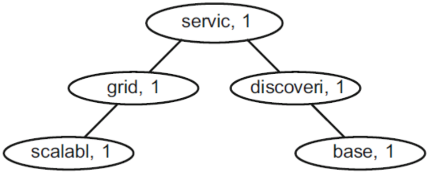


Figure 3: Example of a newly created tree using one similar keyphrase - “scalabl grid servic discoveri base”, where the root word is “servic” (Figure 4 of Rabby et al. (2020) [4])

Step (2) Update μ values

Update all the μ values in the tree simultaneously for each similar keyphrase:

- Increase the μ values of the words "in the path" of a keyphrase by 1.
- Decrease the μ values of all other words in the tree by 1.

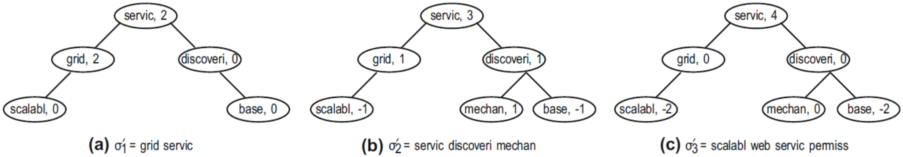
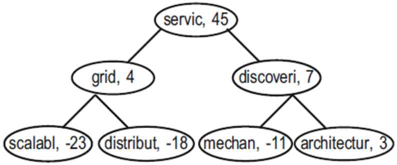


Figure 4: Several steps of tree processing for various similar keyphrases (Adapted from Figure 5 of Rabby et al. (2020) [4])

Step (3) Pruning with Mamu

All the nodes with μ values less than Mamu are pruned from the tree.



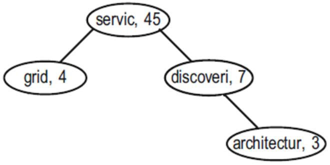


Figure 5: Pruning the constructed tree with mamu = 2 (Adapted from Figure 5 and 6 of Rabby et al. (2020) [4])

Step (4) Extract Final Keyphrases from the Tree

Once we obtain a tree for a given root word, we can extract the final keyphrases from the tree:

- one from each root to leaf path in the left subtree ("left final keyphrases")
- one from each root to leaf path in the right subtree ("right final keyphrases")
- combinations of a left and a right final keyphrase ("combined final keyphrases")

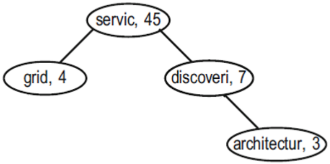


Figure 6: Final KePhEx tree with root word "servic" (Figure 6 of Rabby et al. (2020) [4])

Step (5) Ranking

- We rank the final keyphrases using weights and choose the top N phrases.
- The weight of a keyphrase p, ω_p, is calculated as:

$$\omega_p = \sum_{k=1}^{N} (tf)_k \times \sum_{k=1}^{N} \mu_k,$$

where N is the total number of words in p.

Implementation: Set up

Data set: SemEval-2010 corpus [1, 5]

This corpus is composed of total 244 documents, covering topics from

- (**C**) Distributed Systems (59)
- (**H**) Information Search and Retrieval (64)
- (**I**) Distributed Artificial Intelligence - Multiagent Systems (60)
- (**J**) Social and Behavioral Sciences - Economics (61)

Gold standard keyphrases: keyphrases selected by authors and readers

Implementation: Evaluation metric

Let

- $\kappa_{correct}$ = the number of correctly matched keyphrases with gold standard keyphrases
- $\kappa_{extract}$ = the number of extracted keyphrases from a document
- $\kappa_{standard}$ = the number of keyphrases in gold standard keyphrases list

Then we use the following evaluation metrics.

$$\textbf{Precision} : \rho = \frac{\kappa_{correct}}{\kappa_{extract}}$$

$$\textbf{Recall} : \zeta = \frac{\kappa_{correct}}{\kappa_{standard}}$$

$$\textbf{F1-Score} : \phi = \frac{2 \times \rho \times \zeta}{\rho + \zeta}$$

Implementation: Results

Top 15 keyphrases (H-49)

Extracted: 'retriev time', 'spatial autocorrel', 'languag model score', 'document novelti', 'languag model', 'inform storag', 'high spatial autocorrel', 'acm press', 'languag model retriev', 'anyth autocorrel', 'autocorrel manifest', 'acm intern confer', 'cluster hypothesi', 'score diffus', 'averag precis'

Standard: 'perform predict', 'inform retriev', 'spatial autocorrel', 'autocorrel', 'cluster hypothesi', 'zero relev judgment', 'relationship of predictor+predictor relationship', 'predict power of predictor+predictor predict power', 'languag model score', 'rank of queri+queri rank', 'regular'

Precision: 0.2 Recall: 0.27 F1 Score: 0.23

Implementation: Results

Top 15 keyphrases (H-37)

Extracted: ‘contentbas spam’, ‘content-bas spam detect’, ‘large-scal content-bas spam detect’, ‘spam detect’, ‘data mine’, ‘svm’, ‘spam comment’, ‘blog comment spam detect’, ‘compar perform’, ‘activ set’, ‘spam filter’, ‘blog identif’, ‘benchmark data set’, ‘comment spam detect experi’, ‘expens comput’

Standard: ‘support vector machin’ , ‘content-base filter’, ‘spam filter’, ‘blog’, ‘splog’, ‘link analysi’, ‘machin learn techniqu’, ‘link spam’, ‘content-base spam detect’, ‘bayesian method’, ‘increment updat’, ‘logist regress’, ‘hyperplan’, ‘featur map’, ‘spam filter’

Precision: 0.07 Recall: 0.07 F1 Score: 0.07

Discussion

Summary

- This method does not require training data or any prior knowledge in the domain or in the language.
- The TeKET algorithm runs fast once pre-processing is properly done.

Connection to the class

- This is an unsupervised machine learning method.
- It shows a different usage of binary trees.

Future work

- Being able to match two different words with the same meaning (e.g. 'svm' and 'support vector machine', 'content-bas' and 'content-base') may improve the performance.
- Using the general tree instead of binary tree may allow more various combinations of the keyphrases.
- Ranking root words may prevent important-but-less-frequent phrases from being eliminated.

References

[1] Su Nam Kim et al. "SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles". In: *Language Resources and Evaluation* 47 (Aug. 2010), pp. 21–26. DOI: 10.1007/s10579-012-9210-3.

[2] Deepika Kumawat and Vinesh Jain. "POS Tagging Approaches: A Comparison". In: *International Journal of Computer Applications* 118.6 (May 2015). Full text available, pp. 32–38.

[3] Gollam Rabby et al. *Automatic Keyphrase Extraction*. https://drive.google.com/drive/folders/1e2UrDtYqRAjAE5hso4oXobX_Djuo_VUW. 2020.

[4] Gollam Rabby et al. "TeKET: a Tree-Based Unsupervised Keyphrase Extraction Technique". In: *Cognitive Computation* 12 (2020), pp. 811–833.

[5] SIGLEX. *SemEval-2010 Corpus*. https://semeval2.fbk.eu/semeval2.php?location=data. 2010.