# Rule-Based HierarchicalRank: An Unsupervised Approach to Visible Tag Extraction from Semi-structured Chinese Text

Jicheng Lei[1], Jiali Yu[2], Chunhui He[3](✉), Chong Zhang[3], Bin Ge[3], and Yiping Bao[4]

[1] CETC Big Data Research Institute Co., Ltd., Chengdu, China
[2] Tus-Holdings Co., Ltd., Beijing, China
[3] Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, Hunan, People's Republic of China
xtuhch@163.com
[4] Guizhou Wingscloud Co. Ltd., Guiyang, China

**Abstract.** The large and growing amounts of semi-structured Chinese text present both challenges and opportunities to enhance text mining and knowledge discovery. One such challenge is to automatically extract a small set of visible tag from a document that can accurately reveal the document's topic and can facilitate fast information processing. Unfortunately, at this stage, there is still a certain gap between the existing methods and truly engineering application.

In order to narrow this gap, we propose Rule-Based HierarchicalRank (RBH), an unsupervised method for visible tag extraction from semi-structured Chinese text via a documents' title and non-title two levels. In different level, we use inconsistent methods to extract the candidate visible tags. The experiment results show that the performance of the RBH method is far better than all the baseline methods on visible tag extraction task on two distinct experiment datasets. Specifically, On Paper-Dataset, the rule-based HierarchicalRank methods' precision and F1-score achieves 18.6% and 14.1%, while TOP K = 5. In addition, on Event-Dataset, the best precision of our method is higher 7% than the state-of-the-art method PositionRank with TOP K = 1. Furthermore, the best Recall of RBH achieves 37.7% when TOP K = 5.

**Keywords:** Tag extraction · RBH · Text mining · Knowledge discovery · PageRank

## 1 Introduction

With the development of the internet and information technology, and using text to store data or information is very common in the modern life. These texts are usually divided into three forms: structured, semi-structured and unstructured. Such as a basic table information, academic papers and Weibo content. Among these three forms, semi-structured text is the most commonly used. Because on the one hand, it can store not only structured data, but also unstructured information. Semi-structured text usually

A. C. Nayak and A. Sharma (Eds.): PRICAI 2019, LNAI 11672, pp. 191–205, 2019.
https://doi.org/10.1007/978-3-030-29894-4_15

contains a lot of text information. Because it is not fully structured, most of the unstructured information can't be directly used. It needs to exploit multiple advanced analysis techniques to further complete the structure task to mining the data potential value. Extracting useful tag information from these rapidly-growing semi-structured text has become very challenging.

Text tag it is a general concept in current academe. In order to describe it more details, we divided it into visible tag and hidden tag two categories, which are described as follows:

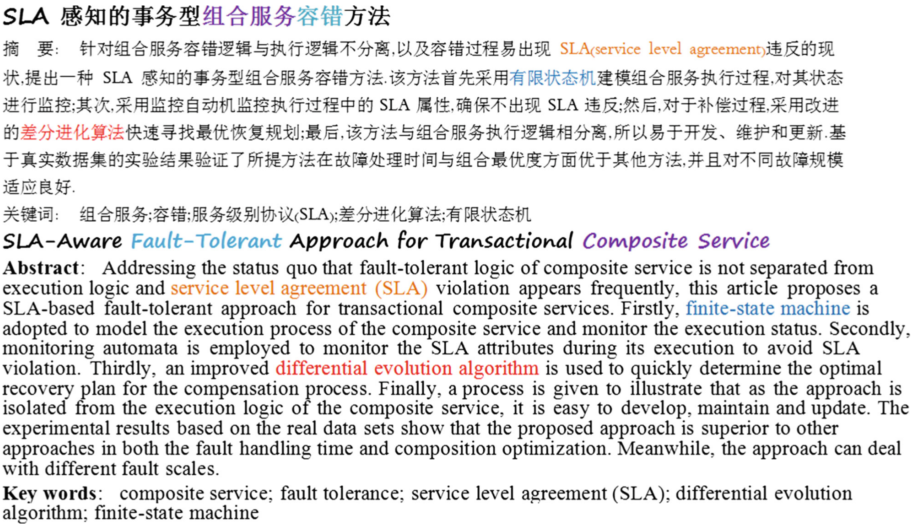
**Visible tag:** it is a visible topic-word for the original text, such as an entity name or a keyphrase.

**Hidden tag:** it is an invisible topic-word for the original text, such as a metaphysical category.

Refining research tasks by classification is a common way. For different classification, we can use different method to extract tags. In this paper, we mainly study visible tag extraction instead of hidden tag extraction.

According to the definition of visible tag, we can know that keyword or keyphrase extraction at most of the time will be very similar to the visible tag extraction task. Via extracting tag can well summarize the topic of a document and can as the basis support for efficient information processing and application task. Such as scientific article summarization [1] or text classification [2], and information retrieval [3] or recommendation [4]. In the field of tag extraction, many methods have been proposed in recently. Related works are mainly divided into two categories: supervised [5] and unsupervised [6].

Kim [7] has pointed out that the supervision method in the field of visible tag extraction is better than the unsupervised method. However, the shortcoming of the supervision method is that it requires a large amount of annotation data to complete the training of the model. This labeling cost is too high to support the automatic extraction of tag for the open-domain task. For unsupervised method, Florescu [8] proposed using the global positional information that appears in the text to improve the extraction of key-phrases and achieve better experimental results on scholarly documents. Huang [9] point out selecting special POS (for example, noun, verb, adj) feature and length of the word can improve keyphrase extraction quality. In addition, via deeply analysis of semi-structured text, it is found that the title of the document is much shorter than the abstract or content, but the part of title often included some tags. Figure 1 shows a semi-structured text (Chinese/English) sample [10], and it contains title, abstract, and keywords of the document.



SLA 感知的事务型组合服务容错方法

摘　要:　针对组合服务容错逻辑与执行逻辑不分离,以及容错过程易出现 SLA(service level agreement)违反的现状,提出一种 SLA 感知的事务型组合服务容错方法.该方法首先采用有限状态机建模组合服务执行过程,对其状态进行监控;其次,采用监控自动机监控执行过程中的 SLA 属性,确保不出现 SLA 违反;然后,对于补偿过程,采用改进的差分进化算法快速寻找最优恢复规划;最后,该方法与组合服务执行逻辑相分离,所以易于开发、维护和更新.基于真实数据集的实验结果验证了所提方法在故障处理时间与组合最优度方面优于其他方法,并且对不同故障规模适应良好.

关键词:　组合服务;容错;服务级别协议(SLA);差分进化算法;有限状态机

SLA-Aware Fault-Tolerant Approach for Transactional Composite Service

**Abstract**: Addressing the status quo that fault-tolerant logic of composite service is not separated from execution logic and service level agreement (SLA) violation appears frequently, this article proposes a SLA-based fault-tolerant approach for transactional composite services. Firstly, finite-state machine is adopted to model the execution process of the composite service and monitor the execution status. Secondly, monitoring automata is employed to monitor the SLA attributes during its execution to avoid SLA violation. Thirdly, an improved differential evolution algorithm is used to quickly determine the optimal recovery plan for the compensation process. Finally, a process is given to illustrate that as the approach is isolated from the execution logic of the composite service, it is easy to develop, maintain and update. The experimental results based on the real data sets show that the proposed approach is superior to other approaches in both the fault handling time and composition optimization. Meanwhile, the approach can deal with different fault scales.

**Key words**: composite service; fault tolerance; service level agreement (SLA); differential evolution algorithm; finite-state machine

**Fig. 1.** The sample of a semi-structured (Chinese/English) text

Nguyen [11] has finished a preliminary study for the type of sample with Fig. 1, by selecting some features such as the distribution of candidate phrases in different sections of a research paper, and the acronym status of a phrase to improve extraction quality.

For the visible tag extraction task, although the advantages are obvious and some research progress has been made. However, the state-of-the-art method still can't support engineering application of the Chinese text visible tag extraction task. In order to improve this situation, via using hierarchical strategies, we propose rule-based HierarchicalRank method to achieve visible tag extraction task from semi-structured Chinese text. In this paper, our contributions are as follows:

(a) Based on our knowledge, for the first time, we propose to segment the text tag as visible tag and hidden tag two categories, and it via the dynamic extension method to automatic extract the visible tags between the title and non-title part in a document.
(b) We propose an unsupervised rule-based and hierarchical extraction model, called HierarchicalRank, that using different extraction strategies at different field, and combine a words' occurrences into a biased PageRank [12] to extract visible tag from semi-structured Chinese text.
(c) We summarize some effective general rules in Chinese tag extraction task, which can assist in the extraction of keyphrase or keyword.
(d) we improved the PositionRank method, and via introduce words' length to compute the weight of the word.

The rest of the paper is organized as follows. The related work is summarized in the next section. rule-based HierarchicalRank method is described in Sect. 3. And then, give the experimental results and analysis in Sect. 4. In Sect. 5, we did a simple discussion. Finally, we conclude the paper and future work in Sect. 6.

## 2 Related Work

For visible tag extraction task, many approaches have been proposed [13, 14]. Using supervised method to complete the extraction of key-phrases, mainly KEA [15], GenEx [16], BDT [17]. In 2012, Chuang [18] proposed a model that incorporates a set of statistical and linguistic features for identifying descriptive terms in a text. In addition, some researches to map the corresponding tag and non-tag in the document into a binary classification problem by labeling the data, and train the relevant classifier according to the labeled data to realize automatic extraction of tags. a method was proposed for visible tag extraction based on the Naive Bayes [15]. Using SVM classifier and combined N-gram language model to extract tags from meeting transcripts has been proven to improve performance [19]. Recently, proposed a neural network architecture based on a Bi-LSTM or RNN that is able to detect the main topic on the well-known INSPEC datasets [20].

In addition, utilizing unsupervised method to extract keyword or keyphrase is also very popular. For this case, tag extraction task is seen as a statistical and ranking problem [21]. A typical approach in the early date was to use TF-IDF to implement visible tag extraction [22, 23]. At present, it is more common to use co-occurrence and graph theory to construct a graph-based ranking algorithm to automatically extract tags. TextRank [24] is a classic method, based on which a series of methods have been derived and have achieved well performances in visible tag extraction in different fields. For example, ExpandRank [25] and PositionRank [8]. In addition, PTR [26] and WAM [27] are also belong to efficient methods on some special tasks.

Usually, different extraction methods are also used for different type of the text. Hu [28] through combined Skip-gram model to propose PKEA algorithm to extract patent keywords. Naidu [29] proposed an algorithm that automatically extract keywords for text summarization in Telugu e-newspaper dataset. In order to improve the accuracy of keyword extraction, Yuan [30] put forward a framework of keyword extraction based on meta-learning. Biswas [31] proposed an unsupervised graph-based visible tag extraction method from Tweets content, called KWG which uses Node-Edge rank centrality measure to calculate the importance of nodes closeness centrality measure to break the ties among the nodes.

In contrast to the above approaches, we propose HierarchicalRank, aimed at capturing both word's POS information and highly frequent weights in a document via hierarchical extraction strategies. The strong contribution of this paper is the design of hierarchical extraction strategies, which is different from existing methods that use the same level to extract visible tags. Our method assigns priority extraction strategy to the title in a document instead of using a uniform distribution over all content.

## 3 Rule-Based HierarchicalRank Method

In this section, we describe a fully unsupervised method HierarchicalRank. Considering that visible tags contain not only keyword or keyphrase, but also some representative entity names. So, in the hierarchical segmentation phase, we segment the title and non-title content as two parts in a contains title's semi-structured text. Particularly, if the semi-structured text being processed is missing a title, the algorithm will be automatically set the title to Null-value. After segmentation as described above, the length of the title will be shorter, while the content of non-title will be longer. For this case, for the title of text, we prefer to use a rule-based approach for automatic extraction of visible tags. For the non-title part, an unsupervised method will be used for the extraction of visible tags. Finally, the extracted results of the two parts are combined according to the visible tag selecting strategies to merge the TOP-K topic words as the visible tags of the semi-structured text. Figure 2 shows the flow chart of the rule-based HierarchicalRank method.

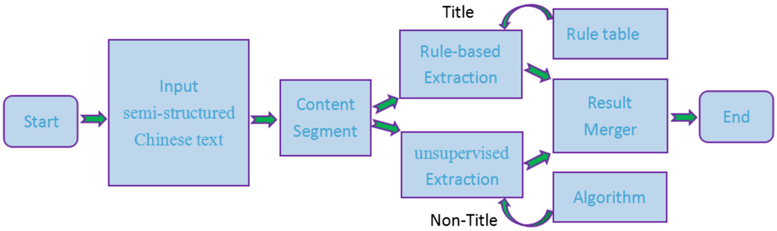


**Fig. 2.** The flow chart of the rule-based HierarchicalRank method

## 3.1 Rule Setting

In order to extract higher quality visible tags from the title of semi-structured text. Based on two semi-structured Chinese text experiment datasets, we made in-depth statistics and analysis for the title with all of the documents, and found that more than 84% of the title contained one or two visible tags. According to the analysis results from the Chinese text part-of-speech (POS) information, it is found that the visible tag in the title include the following eight categories: (1) Person Name; (2) Institution Name; (3) English Proper Noun; (4) Chinese Proper Noun; (5) Subject Proper Noun; (6) Geographic Location Noun; (7) Biological Proper Noun; (8) Compound word of Nouns, Verbs and Adjectives. Based on these statistics results, we set an eight-level rules table for the extraction of visible tags in the title. The detailed level as shown in Table 1:

**Table 1.** Rule table level divide status

| Rule-Level | Rule-Content (POS) | Rule-Level | Rule-Content (POS) |
|---|---|---|---|
| **I** | Person Name (nr) | **V** | Geographic Location Noun (ns) |
| **II** | Institution Name (nt\|nz\|ni) | **VI** | Subject Proper Noun (g) |
| **III** | English Proper Noun (nx) | **VII** | Biological Proper Noun ([nb\|nf\|nh]) |
| **IV** | Chinese Proper Noun (nn) | **VIII** | Compound word ([n+v]\|[adj+n]\|[n+n]\|[v+n]) |

In Table 1, we list all the general rules for the tag extraction from the title. Specially, the priority is decremented from **I** to **VIII**. The end of each rule represents part of speech, for example, 'nr' in rule **I** represent the word belong to a person name, and (nt|nz|ni) in Table 1 indicates OR relationship.

## 3.2 Candidate Tags Hierarchical Extraction

According to Fig. 2, we can know that candidate tags are derived from both the title and non-title two parts. Because the title length is very shorter and the length of the non-title is longer, a hierarchical ranking approach is applied to the different content. For the title, we using a rule-based extraction approach, and the unsupervised approach will be used in the non-title part.

### 3.2.1 Title Candidate Tags Extraction

For the title, by combining the rules have listed in Table 1, it can efficiently and easily extract TOP K no duplicate candidate visible tags. In the experiment, we select TOP K (K = 2) candidate tags from the title, and the K-value should be adjusted in different application scenarios. Here, using the title in Fig. 1 as the visible tags extraction sample. The original title is 'SLA 感知的事务型组合服务容错方法 (SLA-Aware Fault-Tolerant Approach for Transactional Composite Service)'. The result after the word segmentation is '[SLA/nx, 感知/v, 的/ude1, 事务型/b, 组合/v, 服务/n, 容错/b, 方法/n]' . Finally, Matching the rules **III (nx)** and **VIII (v+n)** in Table 1, we easily get two tags ['SLA' ,'组合服务(composite service)' ] as the titles' candidate visible tags.

### 3.2.2 Non-titles Candidate Tags Extraction

Considering that the non-title content is relatively longer than the title, it is not appropriate to use a rule-based method to extract the tag. Therefore, we propose to use an unsupervised method to complete the extraction of the tag. Based on the current state-of-the-art method PositionRank [8], we have made some fine-tuning and improvements to make it better adapt to the hierarchical extraction architecture.

The part of fine tuning: PositionRank method is to extract the keyphrase by using both the title and non-title content as input of the algorithm, but now we only use the non-title part as the input of the improved PositionRank algorithm to extract the visible tag. Improvements: (1) the original PositionRank not well support Chinese text processing, and we introduced the Chinese words segmentation module jieba[1] tokenizer to support Chinese text processing; (2) the PositionRank using the regular expression [(adjective)*(noun)+] to match phrase, for Chinese, we expand it to [(noun)*(verb)+|(adjective)*(noun)+|(noun)*(noun)+|(verb)*(noun)+]; (3) the PositionRank using all position information of the word to calculate the words' weight, we add length of the word to compute the weight. The improved PositionRank method detail described as followed.

[1] https://pypi.org/project/jieba/.

Suppose D is a target document for tags extraction. For non-title of D, via using jieba to finish words segment, and combine both POS filter and words' co-occurrence to build an undirected graph G = (V, E) for D. Two nodes $v_i$ and $v_j$ are connected by an edge $(v_i, v_j) \in E$ if the word corresponding to these nodes co-occur within a window of w contiguous tokens in the content of D. The weight of an edge $(v_i, v_j) \in E$ is computed based on the co-occurrence count on the two words within a window of w successive tokens in D. Let M as its adjacency matrix. An element $m_{ij} \in M$ is set to the weight of edge $(v_i, v_j)$ if there exist an edge between nodes $v_i$ and $v_j$, and is set to zero otherwise. M is the normalized form of matrix with $m_{ij} \in M$ defined as:

$$\mathrm{m}_{ij} = \begin{cases} m_{ij}/\sum_{j=1}^{|V|} m_{ij}, & if \ \sum_{j=1}^{|V|} m_{ij} \neq 0 \\ 0, & otherwise \end{cases} \tag{1}$$

Where |V| is the number of nodes. The rank score of a node $v_i$ is recursively computed by summing the normalized scores of node $v_j$, which are linked to $v_i$. Let S denote the vector of rank scores, for all $v_i \in E$. The initial values of S are set to 1/|V|. The rank score of each node at step T+1, can than be computed recursively using:

$$S(T+1) = M \cdot S(T) \tag{2}$$

To ensure that the PageRank does not get stuck into cycles of the graph, a damping factor $\alpha$ is added to allow the 'teleport' operation to another node in the graph. Hence, the computation of S as followed:

$$S = \alpha \cdot M \cdot S + (1 - \alpha) \cdot P \tag{3}$$

Where S is the principal eigenvector and P is a vector of length |V| with all elements 1/|V|.

The idea of improved PositionRank is to assign big weights with the word that both appeared early and have a big length of the word in a document. Specifically, we want to assign a higher weight to a word appeared in third position as compared to a word found on the tenth position with the same length of word. If the same word appears multiple times in the target document, then we sum all its position weights and product with the length of the word as the word total weight. For example, if the length of the word = 2, and it appears in the 5th and 8th, its weight is:

$$W = 2 * \left(\frac{1}{5} + \frac{1}{8}\right) = 0.65$$

Then, the vector P is set to the normalized weight for each candidate word as follows:

$$P = \left[\frac{P_1}{P_1 + P_2 + \ldots + P_{|V|}}, \frac{P_2}{P_1 + P_2 + \ldots + P_{|V|}}, \ldots, \frac{P_{|\mathrm{V}|}}{P_1 + P_2 + \ldots + P_{|V|}}\right] \tag{4}$$

The rank score of a vertex $\mathrm{v_i}$, i.e., $\mathrm{S(v_i)}$, can be obtained in an algebraic way recursively computing the following equation:

$$S(v_i) = (1 - \alpha) \cdot P_i + \alpha \cdot \sum_{v_j \in Adj(v_i)} \frac{w_{ji}}{O(v_j)} S(v_j) \tag{5}$$

Where $\mathrm{P_i}$ is the weight found in the vector P for vertex $\mathrm{v_i}$, and $\mathrm{O(v_j)} = \sum_{(\mathrm{v_k}) \in Adj(v_j)} w_{jk}$.

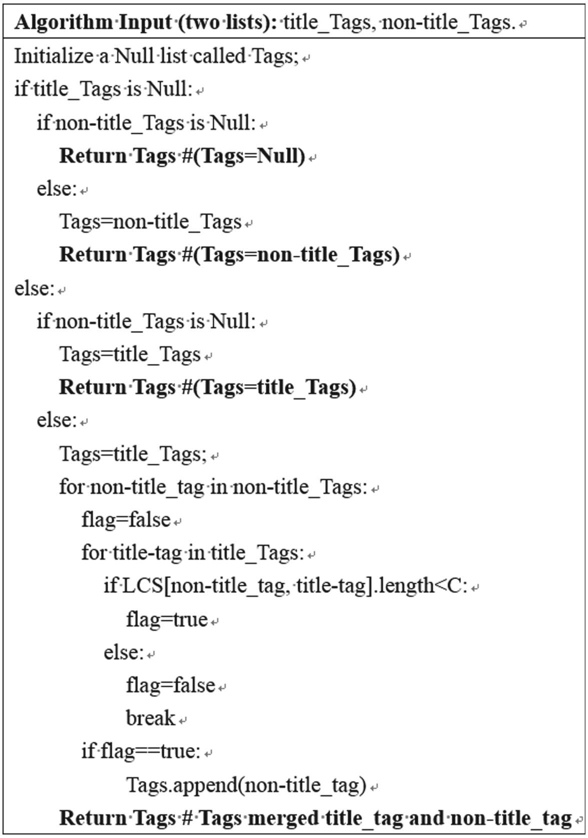
When all candidate words have finished the computation of rank score, we according to appearing contiguous positions in the original document to concatenated the words into a phrase. Then, using the expand regular expression to match all phrases with the phrases' length more than L (L is the least length of a phrase). Finally, via ranking the value of sum (scores) of individual words that include the phrase to extract the TOP K candidate tags.

In our experiments, we setting $\alpha = 0.85$, co-occur window size = 6, candidate tag number $\mathrm{K} = 6, \mathrm{L} = 3$, and the words' rank scores are recursively computed until the difference between two consecutive iteration is less than 0.001 or a number of 200 iterations is reached.

### 3.3 Candidate Tags Merger

According to Sect. 3.2, for a semi-structured Chinese document, by using HierarchicalRank method, we can get title_Tags (TOP 2) and non-title_Tags (TOP 6) two candidate tag lists. Follow the extraction rule, we can know that title_tags contains no more than two non-repeating tags, and non-title_tags contains no more than six non-repeating tags. Considering that they use an independent extraction method, there may exist duplicate or similar tags within two tags' lists.

In order to merge the candidate tags, we use the Longest Common Sequence (LCS) [32] value of the strings to filter similar tags. The detail merge algorithm is as followed.



```
Algorithm Input (two lists): title_Tags, non-title_Tags.
Initialize a Null list called Tags;
if title_Tags is Null:
    if non-title_Tags is Null:
        Return Tags #(Tags=Null)
    else:
        Tags=non-title_Tags
        Return Tags #(Tags=non-title_Tags)
else:
    if non-title_Tags is Null:
        Tags=title_Tags
        Return Tags #(Tags=title_Tags)
    else:
        Tags=title_Tags;
        for non-title_tag in non-title_Tags:
            flag=false
            for title-tag in title_Tags:
                if LCS[non-title_tag, title-tag].length<C:
                    flag=true
                else:
                    flag=false
                    break
            if flag==true:
                    Tags.append(non-title_tag)
        Return Tags # Tags merged title_tag and non-title_tag
```

Using this algorithm, we can easily to get all the visible tags, and for Chinese, we suggest C = 2.

# 4 Experimental Results and Analysis

## 4.1 Datasets and Evaluation Metrics

**Datasets Introduction.** In order to evaluate the performance of HierarchicalRank, we carried out experiments based on two Chinese datasets. The first dataset consists of all the Chinese research papers published by the Journal of Software[2] in 2018. The second dataset consists of the mainly China hot social events published by zhiweidata[3] from January 1st to January 15th, 2019. For the first dataset, we use the title and abstract of

[2] http://www.jos.org.cn/jos/ch/index.aspx.

[3] http://ef.zhiweidata.com/#!/down.

each paper to extract visible tags. The author-input keywords are used as gold-standard for evaluation. For the second dataset, we use the title and description content of each event to extract visible tag. The original visible events' tags are used as gold-standard for evaluation. All two datasets are summarized in Table 2, which show the number of documents in each dataset, the total number of visible tags (Vt), and the average number of visible tags per document (AvgVt).

**Table 2.** A summary of the datasets

| Dataset | #Docs | Vt | AvgVt |
|---|---|---|---|
| Paper dataset (Journal of Software 2018) | 56 | 267 | 4.8 |
| Event dataset (zhiweidata) | 50 | 71 | 1.4 |

**Evaluation Metrics.** We use the mean reciprocal rank (MRR) curve to illustrate our experimental results. MRR value revealed the averaged ranking of the first correct prediction and is defined as:

$$MRR = \frac{1}{|D|}\sum\nolimits_{d \in D} \frac{1}{rd} \tag{6}$$

where D is the collection of documents and rd is the rank at which the first correct keyphrase of document d was found. In addition, we also summarize the results in terms of Precision, Recall, and F1-score in a table to contrast HierarchicalRank with previous methods since these metrics are widely used in previous works.

## 4.2 Core Parameter Setting and Analysis

For the rule-based HierarchicalRank extraction methods, the number of best candidate tags in the title is an important parameter. Considering that the length of the title is very short, and the number of best candidate tags in the title is recommended between 1 and 3. An effective method of processing is to sample and statistic the extracted text and calculate the AvgVt value of this type. For example, the first dataset AvgVt is 4.8, and the second dataset AvgVt is 1.4. Then, set the number of best candidate tags to 1, if the AvgVt <= 2, and the number of best candidate tags set 2, if the AvgVt > 2 and the AvgVt <= 6, in addition, if the AvgVt > 6, we can set 3 as the number of best candidate tags. For the choice of the number of non-title best candidate tags and words co-occur window size, we can refer to PositionRank [8] method to set it.

Figure 3 shows the MRR curve of rule-based HierarchicalRank and TextRank for different values of the number of best candidate tags in the title, on all two datasets.

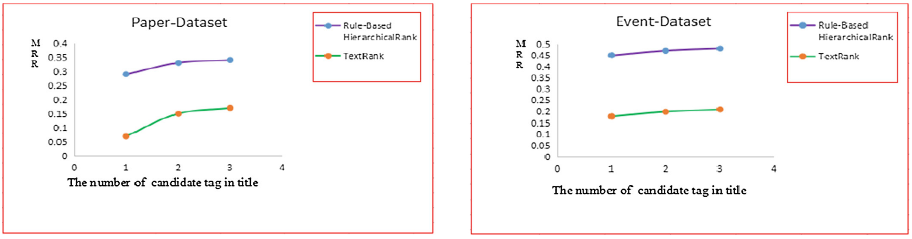


**Fig. 3.** MRR curves with different values for the number of best candidate tags in title

According to Fig. 3, the MRR curve shows that the performance of our method is better than TextRank algorithm for title-tag extraction on all two experiment datasets.

In addition, for improved PositionRank, we add the words' length to compute word weights. In order to evaluate the performance of improved PositionRank algorithm. We combining the original PositionRank and the improved PositionRank algorithm to do comparative experiment in all two datasets. Figure 4 shows the MRR curve for the experiment.

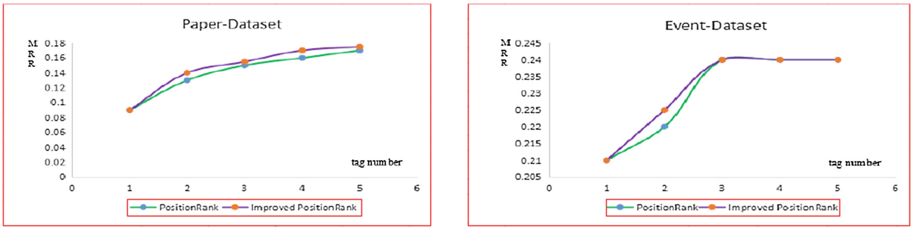


**Fig. 4.** MRR curves of the PositionRank and improved PositionRank algorithm

From Fig. 4, through the MRR curve, we can know that the performance of the improved PositionRank algorithm is slightly better than the original PositionRank algorithm in complete tags extraction task for all two experimental datasets.

Another core parameter of the HierarchicalRank is the total number of extracted visible tags (title_tags + non-title_tags) in a document, and this parameter is usually sensitive to the average length of the document. In our experiments, we set the total number of tag in a document between 1 and 5. Figure 5 shows the MRR curve of rule-based HierarchicalRank and another four unsupervised baselines method, PositionRank, Improved PositionRank, TextRank, and TF-IDF for different total number of the tags within all two datasets.

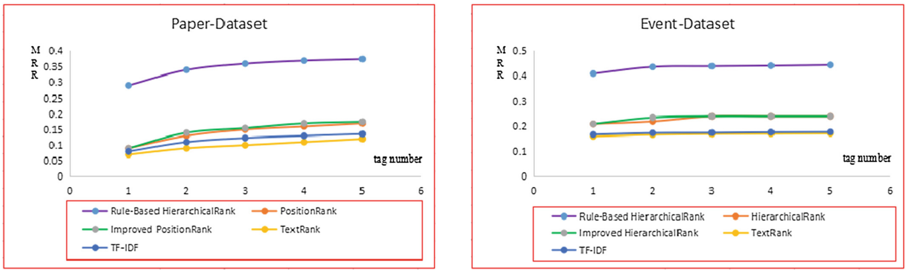


**Fig. 5.** MRR curves of the rule-based HierarchicalRank and baselines algorithm

Via Fig. 5, shows the MRR curve, we can know that the performance of the rule-based HierarchicalRank algorithm is far better than all the baselines method on visible tags extraction task for all two experiment datasets.

### 4.3 Overall Performance

In order to consistent with these prior works on visible tag extraction report results also in terms of precision (P), Recall (R), F1-score (F1). We also calculate the P, R and F1 in our experiment. In Table 3, we show the results of the comparison of rule-based HierarchicalRank with all baselines, in terms of P, R and F1 for TOP K = 1, 3, 5 predicted number of total tags, on all two datasets.

**Table 3.** rule-based HierarchicalRank against baselines in terms of P, R and F1

| Dataset | Unsupervised methods | TOP #1 (%) | | | TOP #3 (%) | | | TOP #5 (%) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | P | R | F1 | P | R | F1 | P | R | F1 |
| Paper-Dataset | rule-based HierarchicalRank | **18.6** | **6.0** | **9.1** | **15.5** | **9.7** | **12.0** | **14.3** | **13.9** | **14.1** |
| | Improved PositionRank | 9.0 | 2.1 | 3.4 | 9.0 | 5.8 | 7.1 | 7.3 | 6.9 | 7.1 |
| | PositionRank | 8.9 | 1.9 | 3.1 | 8.9 | 5.6 | 6.9 | 7.0 | 6.7 | 6.9 |
| | TextRank | 7.1 | 1.5 | 2.5 | 4.8 | 3.0 | 3.7 | 4.0 | 3.7 | 3.9 |
| | TF-IDF | 7.8 | 1.7 | 2.8 | 5.2 | 3.3 | 4.0 | 4.5 | 3.9 | 4.2 |
| Event-Dataset | rule-based HierarchicalRank | **36.0** | **23.7** | **28.6** | **21.0** | **37.7** | **27.0** | **20.6** | **37.7** | **26.6** |
| | Improved PositionRank | 29.7 | 16.6 | 21.3 | 17.1 | 16.7 | 16.9 | 16.5 | 16.7 | 16.6 |
| | PositionRank | 29.0 | 16.0 | 20.6 | 16.4 | 16.0 | 16.2 | 16.0 | 16.0 | 16.0 |
| | TextRank | 16.0 | 8.3 | 10.9 | 11.4 | 11.1 | 11.3 | 11.1 | 11.1 | 11.1 |
| | TF-IDF | 17.2 | 8.6 | 11.5 | 12.1 | 11.6 | 11.8 | 11.2 | 11.6 | 11.4 |

As show from Table 3, the rule-based HierarchicalRank method outperforms all baselines, on all two datasets. The performance of the improved PositionRank algorithm is slightly better than the PositionRank, but it is not obviously on the Paper-Dataset. The rule-based HierarchicalRank method in Paper-Dataset achieves 18.6% Precision and

14.1% F1-score, when TOP K = 5, it is double the PositionRank method on Precision and F1-score. On Event-Dataset, the best Precision of our method higher 7% than PositionRank with TOP K = 1, and the best Recall achieves 37.7% when TOP K = 5.

## 5 Discussion

The main purpose of the rule-based HierarchicalRank method proposed in this paper is to solve the visible tag extraction task of semi-structured text, but it can also adapt the keyphrase or keyword extraction task by adjusting the corresponding parameters. In addition, the method also can be used to processed have title and content unstructured text, such as Chinese Weibo text.

Considering the setting of the rules, one of the shortcomings of this method is that only the visible tag extraction task of Chinese text is currently supported. If reader use this method to process non-Chinese text, it is a new task to redesign the rules according to interrelated text corpus.

In the processing of Chinese text, the accuracy of word segmentation and part-of-speech (POS) annotation often have a greater impact on the performance of the algorithm. In order to improve the performance of the rule-based HierarchicalRank method, when dealing with Chinese text, we suggest that increase the accuracy of word segmentation by introducing the domain dictionary to improve the quality of tag extraction.

For the setting of the core parameters of the method, the relevant solutions are given for the papers of our experimental datasets. If the algorithm is used in other application scenarios, it is recommended that to adjust and test the parameters according to the specific data feature, it will give fuller play to the performance of the algorithm.

This paper only studies the visible tags extraction task. For the hidden tags extracting task, and it may be solved in the future by using supervised learning method or deep learning method.

## 6 Conclusion and Future Work

We proposed a novel hierarchical unsupervised method, called HierarchicalRank, which segment both the title and non-title content of the semi-structured Chinese text to extract the visible tags. In addition, we improved the PositionRank algorithm via introduce words' length to compute the weights. To our knowledge, we are the first to propose using hierarchical method in unsupervised visible tag extraction. Specifically, in title level, the experimental results show that the quality of the extraction of the visible tag can be effectively improved via combining some general rules.

In a addition, the experimental results on two datasets show that rule-based HierarchicalRank method achieves better performance than the currently state-of-the-art method. In the future, it would be interesting to explore the performance of HierarchicalRank on other types of unstructured text, e.g., web news. Finally, combined HierarchicalRank to extract hidden tag belong to another very important direction in text mining.

**Acknowledgment.** This paper was supported by the National Natural Science Foundation of China (NSFC) via grant No. 61872446 and Natural Science Foundation of Hunan Province, China via grant No. 2018JJ2475 and No. 2018JJ2476.

## References

1. Abujbara, A., Arbor, A.: Coherent Citation-Based Summarization of Scientific Papers. Meeting of the Association for Computational Linguistics: Human Language Technologies. DBLP (2011)
2. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proceedings of the International Conference on Machine Learning (1997)
3. Liu, T.Y.: Learning to rank for information retrieval. ACM SIGIR Forum **41**(2), 904 (2010)
4. Li, Y., Nie, J., Yi, Z., Wang, B., Yan, B., Weng, F.: Contextual recommendation based on text mining. In: International Conference on Computational Linguistics: Posters (2010)
5. Caragea, C., Bulgarov, F.A., Godea, A., Gollapalli, S.D.: Citation-enhanced keyphrase extraction from research papers: a supervised approach (2014)
6. Wang, M., Zhao, B., Huang, Y.: PTR: phrase-based topical ranking for automatic keyphrase extraction in scientific publications. In: Hirose, A., Ozawa, S., Doya, K., Ikeda, K., Lee, M., Liu, D. (eds.) ICONIP 2016. LNCS, vol. 9950, pp. 120–128. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46681-1_15
7. Kim, S.N.: Automatic keyphrase extraction from scientific articles. Lang. Resour. Eval. **47** (3), 723–742 (2013)
8. Florescu, C., Caragea, C.: PositionRank: an unsupervised approach to keyphrase extraction from scholarly documents. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 1105–1115 (2017)
9. Huang, C.M., Wu, C.Y.: Effects of word assignment in LDA for news topic discovery. In: IEEE International Congress on Big Data (BigData Congress), pp. 374–380. IEEE (2015)
10. Zhang, J.N., Wang, S.G., Sun, Q.B., Yang, F.C.: SLA-Aware fault-tolerant approach for transactional composite service. J. Softw. **29**(12), 3614–3634 (2018). http://www.jos.org.cn/1000-9825/5313.htm. (in Chinese)
11. Nguyen, T.D., Kan, M.-Y.: Keyphrase extraction in scientific publications. In: Goh, D.H.-L., Cao, T.H., Sølvberg, I.T., Rasmussen, E. (eds.) ICADL 2007. LNCS, vol. 4822, pp. 317–326. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77094-7_41
12. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the web. Stanford InfoLab (1999)
13. Hasan, K.S., 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), vol. 1, pp. 1262–1273 (2014)
14. Merrouni, Z.A., Frikh, B., Ouhbi, B.: Automatic keyphrase extraction: an overview of the state of the art. In: 4th IEEE International Colloquium on Information Science and Technology (CiSt), pp. 306–313. IEEE (2016)
15. Frank, E., Paynter, G.W., Witten, I.H., et al.: Domain-specific keyphrase extraction. In: International Joint Conference on Artificial Intelligence (1999)
16. Turney, P.D.: Learning algorithms for keyphrase extraction. Inf. Retrieval **2**(4), 303–336 (2002)

17. Lopez, P., Romary, L.: HUMB: automatic key term extraction from scientific articles in GROBID. In: Proceedings of International Workshop on Semantic Evaluation, pp. 248–251 (2010)
18. Chuang, J., Manning, C.D., Heer, J.: "Without the clutter of unimportant words": ldescriptive keyphrases for text visualization. ACM Trans. Comput. Hum. Interact. **19**(3), 1–29 (2012)
19. Sheeba, J.I., Vivekanandan, K.: Improved keyword and keyphrase extraction from meeting transcripts. Int. J. Comput. Appl. **52**(13), 11–15 (2013)
20. Basaldella, M., Antolli, E., Serra, G., Tasso, C.: Bidirectional LSTM recurrent neural network for keyphrase extraction. In: Serra, G., Tasso, C. (eds.) IRCDL 2018. CCIS, vol. 806, pp. 180–187. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73165-0_18
21. Alqaryouti, O., Khwileh, H., Farouk, T., Nabhan, A., Shaalan, K.: Graph-based keyword extraction. In: Shaalan, K., Hassanien, A.E., Tolba, F. (eds.) Intelligent Natural Language Processing: Trends and Applications. SCI, vol. 740, pp. 159–172. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-67056-0_9
22. Zhang, Y., Zincirheywood, N., Milios, E.: Narrative text classification for automatic key phrase extraction in web document corpora (2005)
23. Li, J., Zhang, K.: Keyword extraction based on tf/idf for Chinese news document. Wuhan Univ. J. Nat. Sci. **12**(5), 917–921 (2007)
24. Mihalcea, R., Tarau, P.: TextRank: bringing order into texts. In: EMNLP, pp. 404–411 (2004)
25. Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: National Conference on Artificial Intelligence. AAAI Press (2008)
26. Liu, Z., Huang, W., Zheng, Y., Sun, M.: Automatic keyphrase extraction via topic decomposition. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9–11 October 2010, MIT Stata Center, Massachusetts, A meeting of SIGDAT, a Special Interest Group of the ACL. Association for Computational Linguistics (2010)
27. Liu, Z., Chen, X., Zheng, Y., Sun, M.: Automatic keyphrase extraction by bridging vocabulary gap. In: Fifteenth Conference on Computational Natural Language Learning. Association for Computational Linguistics (2011)
28. Hu, J., Li, S., Yao, Y., Yu, L., Yang, G., Hu, J.: Patent keyword extraction algorithm based on distributed representation for patent classification. Entropy **20**(2), 104 (2018)
29. Naidu, R., Bharti, S.K., Babu, K.S., Mohapatra, R.K.: Text summarization with automatic *Keyword* extraction in Telugu e-Newspapers. In: Satapathy, S.C., Bhateja, V., Das, S. (eds.) Smart Computing and Informatics. SIST, vol. 77, pp. 555–564. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-5544-7_54
30. Yuan, M., Zou, C.: Text keyword extraction based on meta-learning strategy. In: International Conference on Big Data and Artificial Intelligence (BDAI), pp. 78–81. IEEE (2018)
31. Biswas, S.K.: Keyword extraction from tweets using weighted graph. In: Mallick, P.K., Balas, V.E., Bhoi, A.K., Zobaa, A.F. (eds.) Cognitive Informatics and Soft Computing. AISC, vol. 768, pp. 475–483. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-0617-4_47
32. Ge, B., He, C.H., Hu, S.Z., Guo, C.: Chinese news hot subtopic discovery and recommendation method based on key phrase and the LDA model. DEStech Transactions on Engineering and Technology Research, ECAR (2018)