Keyword Extraction Model Based on Syntactic Analysis and Topic Distribution (Postprint)

Authors: Wang Hao, Liu Dan, Liu Shuo

Date: 2022-05-10T11:22:57Z

Abstract

To address the issues of the TextRank algorithm ignoring syntactic information and thematic information when extracting document keywords, we propose a document keyword extraction model based on syntactic analysis and thematic distribution. The model extracts document keywords through a two-stage progressive process at the paragraph and document levels. First, at the paragraph level, paragraph keywords are extracted by incorporating word co-occurrence, syntactic, and semantic information; then, paragraphs are clustered according to their themes to form paragraph topic sets; finally, document keywords are extracted based on the distribution characteristics of paragraph topics. On a public news dataset, the extraction performance of the model improved by approximately 10% compared to the original TextRank. Experimental results demonstrate that the method achieves significant performance improvement, confirming the importance of syntactic and thematic information.

Full Text

Preamble

Vol. 39 No. 9
Application Research of Computers
ChinaXiv Partner Journal

Keyword Extraction Model Based on Syntactic Analysis and Topic Distribution

Wang Hao, Liu Dan†, Liu Shuo
(Research Institute of Electronic Science & Technology, University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract: To address the limitations of TextRank in ignoring syntactic and thematic information when extracting document keywords, this paper proposes a document keyword extraction model based on syntactic analysis and topic distribution. The model employs a two-stage progressive extraction process at the paragraph and document levels. First, it extracts paragraph keywords by integrating word co-occurrence, grammatical, and semantic information. Then, it clusters paragraphs according to their topics to form paragraph topic sets. Finally, it extracts document keywords based on the distribution characteristics of paragraph topics. Experiments on a public news dataset demonstrate that the proposed model improves extraction performance by approximately 10% compared to the original TextRank. The results confirm significant improvements in extraction effectiveness, validating the importance of grammatical and thematic information.

Keywords: keyword extraction; TextRank; dependency grammar; semantic distance; paragraph topic

0 Introduction

Keywords provide high-level summarization of document content and concise expression of themes. Keyword extraction technology finds extensive industrial application, with unsupervised methods being particularly favored for their versatility. TextRank represents the most representative graph-based unsupervised extraction algorithm, constructing a word graph where nodes represent words and calculating node weights to extract keywords. However, it neglects semantic and syntactic information as well as textual theme information, resulting in suboptimal performance for long texts and multi-topic documents.

This paper proposes S-TAKE (Syntactic analysis and Paragraph Topic based Article Keyword Extraction Model), a keyword extraction model founded on syntactic analysis and document themes. The model uses paragraphs as the basic textual unit for keyword extraction and implements a two-stage process from paragraph to document level. When extracting paragraph keywords, syntactic analysis incorporates grammatical information into the word graph to mitigate TextRank' s bias toward high-frequency words; word embeddings incorporate semantic information into the transition matrix to address TextRank' s ignorance of semantic relationships between words; and using paragraphs as the basic extraction unit resolves TextRank' s difficulties with long texts. When filtering document keywords, paragraph themes are introduced to form thematic keyword sets, and document keywords are selected based on theme importance and other factors, solving TextRank' s problem of ignoring textual themes. The main innovations are:

a) During word graph construction, syntactic analysis introduces grammatical information and word embeddings introduce semantic information, improving upon TextRank' s single-feature limitation, over-reliance on high-frequency words, and lack of grammatical/semantic considerations.

b) Using paragraphs as the basic extraction unit reduces computational complexity, enhances intra-paragraph thematic coherence, and improves performance on long texts where original TextRank performs poorly.

c) Clustering paragraphs by theme to form thematic keyword sets and selecting document keywords based on theme importance addresses TextRank' s disregard for textual themes.

Experimental results demonstrate significant improvements in precision, recall, and F-score compared to original TextRank and other combinations described herein.

1 Related Work

Keyword extraction constitutes a fundamental task in text processing. Since Luhn proposed frequency-based keyword extraction, researchers have developed numerous approaches, categorized as supervised or unsupervised based on corpus usage.

Supervised methods employ classification or sequence labeling for keyword extraction, commonly using classifiers such as Naive Bayes, SVM, CRF, and MLP, or neural networks for sequence labeling. These methods achieve good performance but require annotated corpora, with effectiveness dependent on training data, imposing significant application constraints.

Unsupervised methods quantify word importance for keyword extraction without requiring annotated corpora, offering high versatility. They include statistical, topic model, and graph-based approaches. Statistical methods measure importance through statistical information but are sensitive to writing style and ignore semantic relationships. Topic model approaches cluster words by theme and select central words as keywords, considering thematic factors but with theme distribution and word clusters heavily influenced by corpus characteristics, creating divergence between cluster centers and actual keywords. Graph-based methods treat words as nodes and edges as relationships, extracting keywords through node weight calculation, exemplified by TextRank. However, TextRank only utilizes word co-occurrence information, with node weights overly influenced by word frequency, prompting numerous improvement models.

The most common improvements incorporate statistical features into TextRank. Sun et al. constructed new transition probabilities using a gravity model considering word influence, distance, and co-occurrence. Xia et al. defined word coverage, position, and clustering influences to weight the transition matrix. Meng et al. defined word span based on first and last occurrence distances, combining it with word position for transition matrix weighting. Ai et al. comprehensively considered word position, part-of-speech, and distribution to modify transition matrix weights. Biswas et al. determined node weights depend primarily on frequency, centrality, and neighbor positions. Niu et al. identified main influencing factors as word coverage, length, frequency, span, and position. Li et al. constructed transition matrices using the mean of TF-IDF values and information entropy. Mao et al. used normalized Google distance for word pair weights and incorporated WordNet information. However, statistical features are text-dependent, and these improvements ignore semantic and syntactic information while neglecting theme impacts on keywords.

Consequently, some improvements combine TextRank with other models, primarily topic models and word representation models. When integrating topic models, some research clusters candidate keywords by theme, constructing word graphs based on word clusters and text information, represented by TopicRank, Topical PageRank, and Multipartiterank. Other research weights transition matrices based on word similarity under themes. When integrating word representation models, semantic information optimizes the transition matrix. For instance, Yu et al. used Word2Vec vector similarity to measure semantic distance, combining statistical information for transition matrix weighting. Xia used word vectors for word clustering to improve inter-node transition probability calculation. Wang et al. introduced Doc2Vec to address weak global representativeness of local information. However, these improvements ignore grammatical information and fail to consider text theme distribution impacts on keywords when using topic models.

2 S-TAKE Model

This paper proposes S-TAKE, a document keyword extraction model based on syntactic analysis and document themes. Using paragraphs as the basic textual unit, the model employs a two-stage paragraph-to-document extraction process comprising "paragraph keyword extraction" and "document keyword filtering." For document D, the process first obtains its paragraph collection {P , P , ···, P }. According to the paragraph keyword extraction algorithm, paragraph word graphs G and transition matrices C are constructed to calculate node weights and obtain paragraph keyword sets KW . Then, paragraph text generates paragraph theme vectors PT , and paragraphs are clustered by theme according to the document keyword filtering algorithm. Paragraph keywords are combined to form thematic keyword sets TK , and document keyword set DK is derived by filtering based on theme importance, word frequency, and other factors. The model architecture is illustrated in Figure 1.

2.1 Paragraph Keyword Extraction Algorithm

Documents typically contain multiple themes. Traditional keyword extraction methods construct word graphs using entire documents, ignoring multi-theme characteristics and resulting in inconsistent themes within word graphs, yielding poor document-level performance. Paragraphs, as basic document components, exhibit high intra-paragraph thematic consistency, and document keywords are contained within paragraph keywords. Therefore, we propose using paragraphs as the basic textual unit for keyword extraction.

The model extracts paragraph keywords based on TextRank. First, paragraph word graph G = V, E is constructed, where vertex set V is obtained by filtering paragraph tokenization results, and edge set E is obtained through syntactic analysis incorporating grammatical information combined with word co-occurrence. Then, word embedding models capture semantic information, and different weights are assigned to edges based on semantic similarity to form transition matrix C. Finally, PageRank formulas calculate node weights PR using word graph structure and transition matrix, and keywords are selected based on node weights.

2.1.1 Graph Construction Based on Syntactic Analysis The word graph G = V, E consists of vertex set V corresponding to candidate keywords and edge set E corresponding to related keyword pairs.

1) **Vertex Set V Acquisition**
   Word graph vertices correspond to words in the text. Due to keyword characteristics and Chinese writing conventions, tokenization results require filtering to construct vertex set V. Filtering non-keywords explicitly reduces graph scale, improves construction quality, and optimizes extraction performance.

Keywords reflect document themes and must be content words with actual meaning. Filtering operates primarily based on part-of-speech and stopword lists. The model considers nouns, verbs, numerals, adjectives, and adverbs as potential keyword parts-of-speech, filtering out other parts-of-speech and stopwords to form the candidate keyword set, i.e., vertex set V.

2) **Edge Set E Acquisition**
   TextRank uses word contextual features (co-occurrence) as the sole criterion for word relationships, resulting in single-dimensional features sensitive to writing style. Beyond contextual features, words possess grammar information independent of writing style. Grammatical information is manifested through dependency relationships between words, typically obtained via syntactic analysis and represented as triples: (w , w , r , S , R), where w and w are words with dependency relationship directed from w to w ; r is the arc value representing dependency type; S is the analyzed sentence; and R is the dependency type set. When a dependency relationship exists between words and both belong to candidate keyword set V, corresponding vertices are considered connected. If edge set E does not contain this edge, it is added:

Edges obtained through syntactic analysis reflect grammatical associations, demonstrating robustness to writing style variations. Grammatical associations are unaffected by word distance, capturing long-distance relationships. However, a sentence contains only (word count - 1) dependencies, and edge quantity further decreases after word filtering, making word graphs overly sparse when using only syntactic analysis. Additionally, since sentence cores are typically verbs, exclusive use of syntactic analysis overemphasizes verb importance.

Therefore, the model considers both grammatical and co-occurrence information dimensions when constructing word graphs, performing union operations on edge sets obtained through both approaches. We propose a word graph construction algorithm integrating grammatical and co-occurrence information:

Algorithm 1: Word Graph Construction Based on Syntactic Analysis
Input: Paragraph text P
Output: Paragraph word graph G

```
a) Initialize word graph G , G  =  V, E , V =  , E =
b) Initialize variable len(sliding window SW) = w
c) Segment P into sentence list {S , S , ···, S }
d) FOR i = 1 to n:
e) Tokenize S  into word list {w  , w  , ···, w  }
f) Initialize filtered sentence SV =
g) FOR w   in S :
h) IF w    filter dictionary:
i)   Continue
j)   IF w    vertex set V: Add v_w  → V
k) Obtain sentence dependency set D = {d  , d  , ···, d      }
l) FOR d   in D:
m) IF V_wp, V_wq   vertex set V and e =  V_wp, V_wq   edge set E:
n)   Add  V_wp, V_wq  → E
o) FOR j = 1 to len(SV):
p) FOR k = 1 to w:
```

```
q)  IF v_w , v_w     edge set E: Add  v_w , v_w    → E
```

The resulting word graph simultaneously considers grammatical relationships and sequential co-occurrence, addressing TextRank' s lack of grammatical information and long-distance word associations while avoiding sparsity and verb overemphasis from pure dependency parsing.

2.1.2 Transfer Matrix Construction Based on Semantic Weighting

The transfer matrix represents core elements for paragraph keyword extraction, with elements indicating inter-node transition probabilities expressible as edge weight ratios. TextRank assigns uniform weights to all edges, assuming equal transition probabilities from a node to its neighbors, but actual transitions exhibit preferences. Since word graph vertices correspond to text words and different edges connect different words, weights can be assigned by measuring relationships between connected words.

The most direct method for measuring word relationships uses semantic information to calculate semantic distance. Semantic information is typically represented through word vectors, including static vectors like Word2Vec and dynamic vectors like BERT. Therefore, we incorporate semantic information via word vectors to weight the transfer matrix.

Using paragraphs as the basic unit for graph construction leverages strong thematic cohesion—each paragraph corresponds to one theme, with keywords under the same theme being semantically close. When measuring transition probability, higher semantic similarity yields higher transition probability. Vector representations commonly use cosine distance to measure semantic proximity:

where x , x are word vectors and s is the cosine distance in [-1, 1]. Larger s indicates greater vector similarity and closer semantic meaning.

While considering semantic information, edge occurrence frequency must also be incorporated. Edge frequency represents association counts between related words—higher frequency indicates stronger contextual relevance. We construct weight matrix W based on cosine distance and occurrence frequency:

Weighting the initial transfer matrix C using W yields the actual transfer matrix:

Algorithm 2: Transfer Matrix Generation Based on Semantic Weighting
Input: Paragraph text P, word graph structure G
Output: Corresponding transfer matrix C

a) Construct two |V|×|V| matrices based on G ' s vertex set size: initial transfer matrix C and weight matrix W

b) Initialize C according to G ' s edge set E

```
c) Segment P into sentence list {s , s , ···, s }
d) FOR i = 1 to n:
e) Tokenize S  into word list {w  , w  , ···, w  }
f) FOR e in edges contained in sentence S :
g) Obtain vector representations x_w , x_w  for edge-connected nodes V , V_q corresponding
h) Calculate edge weight S   based on x_w , x_w
i) Add weight S   to corresponding elements w_pq and w_qp in weight matrix
j) Multiply initial weight matrix C  and weight matrix W element-wise to
   obtain transfer matrix C
```

The resulting transfer matrix simultaneously considers semantic relationships and word pair occurrence frequency, producing a matrix better aligned with actual Chinese expression patterns.

2.1.3 PR Values and Keyword Selection With word graph G and transfer matrix C obtained, PageRank' s PR formula calculates node weights:

where PR(v) represents node v ' s weight; d is the damping factor; In(v) is node v ' s incoming node set; Out(v) is node v ' s outgoing node set; and c represents transition probability from node v to v in matrix C. Node weights require iterative calculation until stable. Since all node weights must be updated simultaneously each iteration, matrix operations are employed. Using column vector R to represent all nodes' PR values at time t, the calculation formula at time t+1 is:

where C is the transfer matrix and m is the node count in word graph G. Iteration continues until weights stabilize or reach a predetermined count. At stability, R = R . The final PR matrix R contains each node' s ultimate PR value. Nodes are sorted in descending PR order, and top-K words are selected as keywords.

2.2 Paragraph Clustering and Keyword Filtering Based on Themes

Original TextRank and its improvements construct word graphs at the document level, destroying original textual and thematic structures while ignoring subtheme information. Chinese document themes typically exhibit hierarchical structures, with each paragraph usually elaborating one theme. More important themes receive more textual description, i.e., more paragraphs.

Therefore, based on extracted paragraph keywords, the model proposes a document keyword filtering algorithm based on themes. The algorithm first generates paragraph theme vectors PT from paragraph P ’ s text and clusters paragraphs by theme using these vectors. Keywords from same-theme paragraphs are merged to form thematic keyword lists. This approach fully considers textual structure and theme information, addressing original TextRank’s and other methods’ ignorance of textual and thematic structures. Finally, thematic keywords are filtered based on word frequency and theme importance to obtain document keyword set DK.

Algorithm 3: Document Keyword Filtering Based on Thematic Clustering
Input: Paragraph P text, paragraph keyword set KW
Output: Document keyword set DK

```
a) FOR i = 1 to count(P ):
b) Generate paragraph theme vector PT from paragraph P text
c) Cluster paragraphs by theme based on PT to form theme set {T , T , ···, T }
d) Merge paragraph keywords from same-theme paragraphs to form thematic keyword sets
e) FOR i = 1 to m:
f) Calculate theme importance IT based on paragraph count corresponding to theme
g) Sort words in KWT by frequency in theme-corresponding paragraphs in descending order
h) Select top IT ×K keywords from KWT and add to document keyword set DK
i) IF count(DK) < K:
j) Sort all remaining thematic keywords by document frequency in descending order
k) Select top K-count(DK) keywords not in DK and add to DK
```

First, Sentence-Transformer constructs embedding representations for each paragraph. Based on BERT, Sentence-Transformer has length limitations on input text, employing truncation for texts exceeding limits.

Using the obtained embedding as paragraph theme vector PT , K-means clusters paragraphs by theme to form theme-based paragraph sets. Since document themes are generally limited, K is set to 3 for K-means.

Merging paragraph keyword lists under the same "theme" forms thematic keyword lists KWT . Keyword occurrence frequency in theme-corresponding paragraphs is counted—higher frequency indicates greater theme representativeness. Paragraph keyword lists are sorted in descending frequency order.

Different themes have varying importance to the text. Themes with more corresponding paragraphs are more important and should occupy larger proportions in the document keyword list. Therefore, theme weights IT are assigned based on corresponding paragraph counts:

where count(•) counts elements in parentheses. Based on weights, IT ×K keywords are selected from each theme to contribute to document keywords. Duplicate keywords are merged, and remaining keywords are supplemented according to frequency to form the final document keyword list DK.

---

3 Experiments

3.1 Experimental Data and Environment

Experiments selected two original datasets, filtered to construct experimental data.

Original Dataset 1 is the Southern Weekend news dataset constructed by Xia et al. Randomly selecting 300 articles exceeding 1000 characters and splitting original keywords by basic words formed the nz_news dataset, containing 1090 unsplit keywords and 1467 split keywords, averaging 2766.790 characters, 3.633 unsplit keywords, and 4.890 split keywords per article.

Original Dataset 2 comprises news data crawled from various portals with indivisible keywords. Randomly selecting 300 articles between 500-1000 characters formed the random_news dataset, containing 4642 keywords, averaging 729.197 characters and 15.473 keywords per article. Sample data are shown in Figure 2.

The experimental environment is detailed in Table 1.

Experiments adopt precision (P), recall (R), and F-score (F) as evaluation metrics. Let A represent the correct keyword set provided by the test dataset and E represent the extracted keyword set. Evaluation formulas are:

3.3 Results and Analysis

Experiment 1: Performance Comparison Across Different Keyword Counts
To validate extraction effectiveness and the impact of different keyword quantities, experiments were conducted on nz_news and random_news datasets using methods M1-M5 to extract 3, 5, and 7 keywords.

Experiment 2: Impact of Sliding Window Length on Keyword Extraction

Co-occurrence window length determines co-occurrence pair count and significantly affects graph construction. To verify its impact, methods M1 and M5 extracted 10 keywords using window lengths 2-6 on random_news dataset.

Method M1 (original TextRank) shows decreasing extraction effectiveness on random_news as window length increases, consistent with its original paper' s conclusion of using window length 2. Method M5 achieves best performance at window length 4, with gradual decline thereafter. Results suggest that dependency relationships mitigate the impact of increasing co-occurrence windows to some extent.

Method M6 (S-TAKE Model) Validation

Method M6 represents the S-TAKE model. To validate its effectiveness, it is compared with method M5 on nz_news dataset (random_news lacks paragraph information). With co-occurrence window length 3, experiments extract 5, 7, and 10 keywords, comparing performance on "split keywords" and "unsplit keywords."

When keyword count is 5, S-TAKE outperforms M5 on "unsplit keywords" but underperforms on "split keywords." At keyword counts 7 and 10, S-TAKE comprehensively outperforms M5.

Using the corpus excerpt in Figure 2(a) and "split keywords" as the standard, original TextRank extracting 7 keywords yields: [Japan, defendant, corporation, court, China, Mitsui, merchant ship, report, vessel, according to law]. S-TAKE yields: [Japan, Shinzo Abe, corporation, China, merchant ship, Mitsui, Yasukuni Shrine, according to law, vessel, report]. For top-3 keywords, TextRank hits 2 while S-TAKE hits 3; for top-7, TextRank hits 4 while S-TAKE hits 5; for top-10, TextRank still hits only 4 while S-TAKE hits 6. Using "unsplit keywords" as the standard with 10 extracted keywords, original TextRank hits only 2 while the proposed method hits 3. Both methods struggle with "compound" keywords.

Analysis reveals that split keywords generally represent refined expressions of themes, often co-occurring with theme words in keyword lists (e.g., "pension-pension," "medical insurance-insurance"), with theme words having greater weights. When ignoring themes and extracting few keywords, multiple words from the same theme are easily selected, making split keywords more likely to be extracted. Thus M5 outperforms S-TAKE on "split keywords" when extraction counts are low. However, S-TAKE considers thematic elements and extracts theme words from other document themes, yielding better performance on "unsplit keywords."

4 Conclusion

This paper enhances TextRank by incorporating syntactic and semantic information, improving keyword extraction capability. Based on Chinese writing characteristics, it proposes constructing word graphs at the paragraph level and obtaining document keywords through paragraph theme clustering, addressing TextRank's ignorance of textual structure and theme information. Experimental results demonstrate significant S-TAKE performance improvements over original TextRank, proving the importance of grammatical and semantic information, the significance of theme information, and validating the paragraph theme clustering approach.

The research also raises new questions: how to better model paragraph themes to reduce errors, how to assign different weights to different dependency relationships, and how to weight forward/backward directions of the same dependency edge. Future work will continue investigating these aspects.

References

[1] Luhn H P. A Statistical Approach to Mechanized Encoding and Searching of Literary Information [J]. IBM Journal of Research and Development, 1957, 1 (4): 309–317.

[2] Mihalcea R, Tarau P. TextRank: Bringing Order into Text [C]. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. Barcelona, Spain: Association for Computational Linguistics, 2004: 404–411.

[3] Sun Fuquan, Zhang Jingjing, Liu Bingyu, et al. Improved TextRank keyword extraction algorithm based on gravity [J]. Computer Applications and Software, 2020, 37 (07): 216-220, 295.

[4] Xia Tian. Study on Keyword Extraction Using Word Position Weighted TextRank [J]. Data Analysis and Knowledge Discovery, 2013 (9): 30-34.

[5] Meng Caixia, Zhang Yan, Li Nannan. Research on the improvement method of keyword extraction based on TextRank [J]. Computer and Digital Engineering, 2020, 48 (12): 3022-3026.

[6] Ai Jinyong. A Study on TextRank Keyword Extraction Method for Tibetan Texts Incorporating Multiple Features [J]. Information Research, 2020 (07): 1-6.

[7] Biswas S K, Bordoloi M, Shreya J. A graph based keyword extraction model using collective node weight [J]. Expert Systems with Applications, 2018, 97: 51-59.

[8] Niu Yongjie, Jiang Ning. Research on influence factors of keyword extraction algorithm TextRank [J]. Electronic Design Engineering, 2020, 28 (12): 1-5.

[9] Li Zhiqiang, Pan Suhan, Dai Juan, et al. An improved TextRank keyword extraction algorithm [J]. Computer Technology and Development, 2020, 30 (03):

77-81.

[10] Mao Xiangke, Huang Shaobin, Li Rongsheng, et al. Automatic Keywords Extraction Based on Co-Occurrence and Semantic Relationships Between Words [J]. IEEE Access, 2020, PP (99): 1-1.

[11] Bougouin A, Boudin F, Daille B. TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction [C]. In: Proceedings of the Sixth International Joint Conference on Natural Language Processing. Nagoya, Japan: Asian Federation of Natural Language Processing/ACL, 2013.

[12] Liu Zhiyuan, Huang Wenyi, Zheng Yabin, et al. Automatic Keyphrase Extraction via Topic Decomposition [C]. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Massachusetts, USA: Association for Computational Linguistics, 2010.

[13] Boudin F. Unsupervised key phrase extraction with multipartite graphs [C]// Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT, Association for Computational Linguistics, New Orleans: June 1-6, 2018, 2: 667-672.

[14] Sterckx L, Demeester T, Deleu J, et al. Creation and evaluation of large Keyphrase extraction collections with multiple opinions [J]. Language Resources and Evaluation, 2017, 52: 503-532.

[15] Zhang Binglei. Research on Chinese short text classification based on TextRank and LDA [J]. China Computer & Communication, 2021, 33 (06): 12-14.

[16] Yu Bengong, Zhang Hongmei, Cao Yumeng. Improved TextRank Keyword Extraction Method Based on Multivariate Features Weighted [J]. Digital Library Forum, 2020 (03): 41-50.

[17] Xia Tian. Extracting Keywords with Modified TextRank Model [J]. Data Analysis and Knowledge Discovery, 2017, 1 (2): 28-34.

[18] Wang Wei, Li Xiangshun, Yu Sheng. Chinese Text Keyword Extraction Based on Doc2vec And TextRank [C]// 2020 Chinese Control And Decision Conference (CCDC). 2020.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.