CEUR-WS.org/Vol-3331/paper03.pdf

# Multi-Feature Fusion TextRank Algorithm for Sentence-Oriented Keyword Extraction

Shuo-shuo Meng, Guo-sheng Hao*, Zi-hao Yang

*School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, China*

**Abstract**

Most keyword extraction algorithms mainly focus on the extraction from a document but not a sentence. A document hosts more information than a sentence, therefore keyword extraction from a sentence is a challenging task. In addition, keyword extraction from a sentence has potential application in many fields, such as question answering systems, text search, recommendation systems, etc. Therefore, this paper proposes the multi-feature fusion TextRank algorithm for sentence-oriented keyword extraction by integrating knowledge of features in the initial score of keywords and calculation of the probability transfer matrix. The initial scores of candidate keywords are adjusted by fusing the term frequency and part of speech features in the sentence. And the probability transfer matrix to calculate the scores is tuned by using the semantic and syntactic features among the candidate keywords. Based on the scores of candidate keywords, the top *K* words are selected as the keywords of the sentence. The experiments show that our method outperforms in the indices of *P*, *R*, and *F*.

**Keywords**

sentence; keyword extraction; TextRank; feature fusion

## 1. Introduction

Keyword extraction is one of the popular topics in the field of natural language processing[1]. It is also widely applied in our daily life since keywords help us save time by transferring the main ideas of documents through fewer words.

Keyword extraction from a document but not from a sentence is widely studied, such as from an abstract of a paper[2], news[3], patent texts[4], etc. In document keyword extraction, the knowledge in titles[3], paragraphs[5], and the location of words[6] can be used, and some deep learning methods as the end-to-end extraction are also presented[2].

Keyword extraction methods are divided into supervised and unsupervised types[1]. In supervised type, keyword extraction is regarded as a binary classification or multi-classification task[1]. In unsupervised type, they can be summarized into three kinds: keyword extraction based on statistical features, the topic model and the graph model.

Among the keyword extraction based on statistical features, TF-IDF (Term Frequency-inverse Document Frequency)[7] is well-known for its simplicity and efficiency. The topic model based methods extract keywords according to the topic distribution of documents[8]. Among the graph model based keyword extraction, a popular method is TextRank algorithm[9]. Inspired by PageRank algorithm[10], TextRank algorithm includes three steps: (1) construct a graph model according to co-occurrence relationship between words, (2) adjust the scores iteratively, (3) and select the top *K* words with the highest score as keywords. The improvement of TextRank mainly focused on two aspects: the scores initialization of candidate keywords, and the construction of the probability transfer matrix.

In the improvement from the scores initialization of candidate keywords, many features in document are introduced, such as term frequency[11], the length of words, the position of words, the part of speech[12], the narrative table[11], the importance of words in the document's title[3]. For the improvement

AHPCAI2022@2nd International Conference on Algorithms, High Performance Computing and Artificial Intelligence

EMAIL: 2020200454@jsnu.edu.cn (Shuo-Shuo Meng), *hgskd@jsnu.edu.cn (Guo-Sheng Hao), 2020200445@jsnu.edu.cn (Zi- Hao Yang)

© 2022 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

from the construction of the probability transfer matrix, the included features are as follows: the similarity[13] calculated according to Word2Vec, the reduction of the sparsity of matrix[5] based Doc2vec.

Compared with a document, a sentence is shorter, and it does not have adequate structure information.The challenge of sentence keyword extraction is the sparsity of text semantics [5]. Most of the existing keyword extraction algorithms are studied with the background of documents. This paper designs an algorithm suitable for sentence keyword extraction.

However, the TextRank algorithm has the following shortcomings when applied to keyword extraction from a sentence: (1) It does not consider the inherent features of candidate keywords, such as term frequency and part of speech. (2) It is difficult to obtain deeper relations among words because its probability transfer matrix only makes use of the co-occurrence relationships among words.

This paper proposes the multi-feature fusion (MFF) TextRank algorithm for sentence-oriented keyword extraction to address the above shortcomings. For shortcoming (1), the word term frequency and part of speech features in the sentence are considered to assign initial scores to candidate keywords. For shortcoming (2), except for the co-occurrence relationship, we also take the semantic and syntactic features into consideration to obtain deeper relationships between words.

The rest of the paper is organized as follows: Section 1 points out the shortcomings of the state of the arts, and reviews the related work on keyword extraction. Section 2 gives the main idea and the key elements of this paper. The experiments are shown in Section 3 and this paper is concluded in Section 4.

## 2. Muti-Feature Fusion Textrank Algorithm For Sentence-Oriented Keyword Extraction

This section firstly introduces the main idea of the algorithm. Secondly, a method was proposed to improve candidate keyword scores. Thirdly, the algorithm to improve the construction of the probability transfer matrix is given. Finally, the multi-feature fusion TextRank algorithm for sentence-oriented keyword extraction is presented.

### 2.1. Main idea of the algorithm

Compared with a document, a sentence has less information that can be made use of. Therefore, when applying TextRank in sentence-oriented keyword extraction, the information will have to be fully used. In this paper, we propose two ideas to make use of the information of the sentence: (1) When assign initial scores to the candidate keywords, the term frequency and part of speech features are fused into the calculation, and (2) When construct probability transfer matrix, both semantic and syntactic features are considered by using the summarization of the semantic similarity matrix and dependency relevance matrix. Among them, the semantic similarity matrix is composed of the semantic similarity among the candidate keywords trained by Word2vec.

The above two ideas are embedded in the scores initialization of candidate keywords and the construction of probability transfer matrix separately, as shown in Figure 1, which gives the framework of MFF TextRank. Six parts are included in this framework, and they are organized according to their relationship in the keyword extraction. Firstly, based on candidate keyword sets and co-occurrence relationships, an undirected graph should be constructed. Secondly, the initial scores of the keywords are assigned. Thirdly, the probability transfer matrix is constructed. Fourthly, iteratively calculate candidate keywords' scores. Fifthly, rank candidate keywords according to their scores, and at last take the top *K* candidate keywords with high scores as the result.

The model of TextRank algorithm can be formally expressed as:

$$s_t(v_i) = (1-d) + d * \sum_{v_j \in In(v_i)} \frac{w_{ji}}{\sum_{v_k \in Out(v_j)} w_{jk}} s_{t-1}(v_j) \tag{1}$$

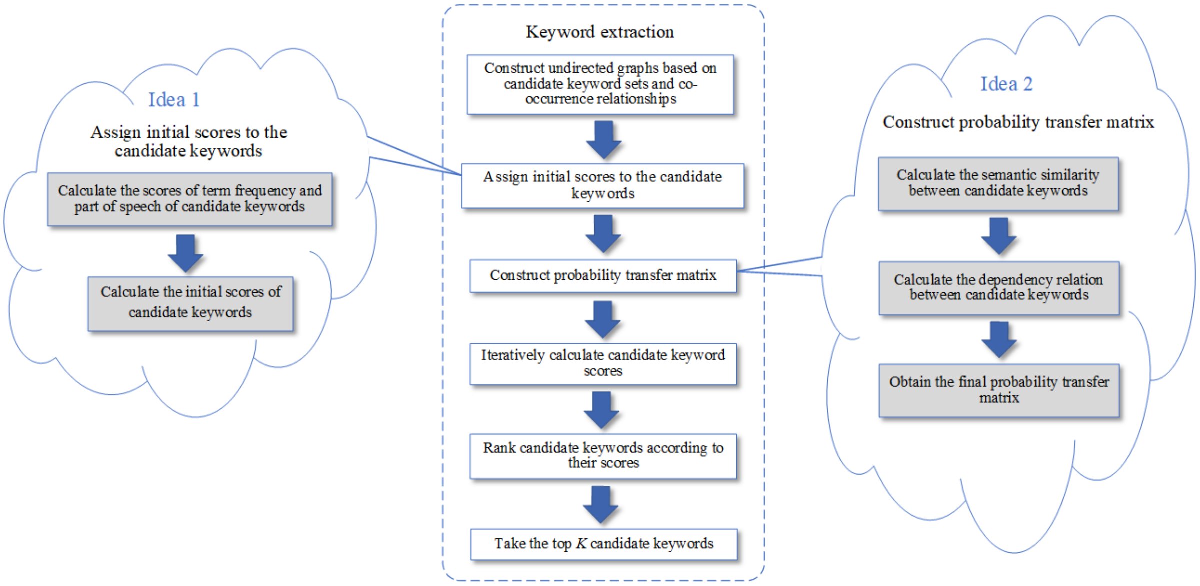
Idea 1
Assign initial scores to the candidate keywords
Calculate the scores of term frequency and part of speech of candidate keywords
Calculate the initial scores of candidate keywords
Keyword extraction
Construct undirected graphs based on candidate keyword sets and co-occurrence relationships
Assign initial scores to the candidate keywords
Construct probability transfer matrix
Iteratively calculate candidate keyword scores
Rank candidate keywords according to their scores
Take the top K candidate keywords
Idea 2
Construct probability transfer matrix
Calculate the semantic similarity between candidate keywords
Calculate the dependency relation between candidate keywords
Obtain the final probability transfer matrix

**Fig.1** MFF TextRank Framework

where $In(v_i)$ denotes the set of all nodes (words) that have edges heading to node $v_i$; $Out(v_j)$ denotes the set of nodes that have edges tailing from the node $v_j$; $s_t(v_i)$ denotes the TextRank score of the node $v_i$ in $t$-th iteration; $w_{ji}$ denotes the weight of the edge between the node $v_j$ and $v_i$; $w_{ji}$ will be taken as the element value in the probability transfer probability, and $d \in [0,1]$ is a damping factor, and it is generally set to 0.85 [9]. The vector form of TextRank can be rewritten as:

$$\boldsymbol{s}_t = (1-d) * \boldsymbol{e} + d * \boldsymbol{M} * \boldsymbol{s}_{t\text{-}1} \tag{2}$$

where $\boldsymbol{s}_t$ denotes a vector form of the scores of all keywords; $\boldsymbol{M}$ denotes the probability transfer matrix, $\boldsymbol{e}$ is a unit-vector. Eq. (2) presents the iteration of TextRank, i.e. the iterative computation will continue until the termination condition, such as $|\boldsymbol{s}_t\text{-}\boldsymbol{s}_{t\text{-}1}|<\varepsilon$, is satisfied, where $\varepsilon$ is the given threshold. Then all candidate keyword scores are ranked, and the top $K$ words are selected as the keywords of the sentence.

The ideas to make use of the knowledge in a sentence to initialize the score of $s_0(v_i)$, and to calculate the probability transfer matrix are explained separately as below.

## 2.2. Scores initialization of candidate keywords

The canonical TextRank algorithm assigns the initial scores of each candidate keyword to 1 or $1/N$ ($N$ is the number of candidate keywords) by default, and it ignores the knowledge contained in each candidate keyword. In this paper, we integrate various knowledge and assign different initial scores to each candidate keyword accordingly. The initial score is calculated as:

$$s_0(v_i) = s^1(v_i) * s^2(v_i) \tag{3}$$

where $v_i$ is the $i$-th candidate keyword; $s^1(v_i)$ denotes the term frequency of $v_i$; $s^2(v_i)$ denotes the score of part of speech of $v_i$. The higher $s^1(v_i)$ is, the more important this word is. Similarly, since the part of speech of the candidate keyword is different, the score should be different. $s^1(v_i)$ and $s^2(v_i)$ are described as follows.From the perspective of part of speech, keywords in the sentence are often nouns, verbs, and adjectives[14], therefore $s^2(v_i)$ is calculated according to the corresponding part of speech as follows: 2 for a noun, 1.5 for a verb, 1 for an adjective, and 0.5 for others.

The initial scores of all candidate keywords can be represented by a vector $\boldsymbol{S_0}$ with dimension $n$, as shown in equation (4).

$$\boldsymbol{S}_0 = \left(s_0(v_1), s_0(v_2), \cdots, s_0(v_i), \cdots, s_0(v_n)\right) \tag{4}$$

## 2.3. Construction of probability transfer matrix

Based on the canonical TextRank algorithm, in terms of semantics, this paper calculates the semantic similarity based on Word2vec. In terms of syntax, the relevance between words is calculated based on dependency parsing. They are described separately as below.

In terms of semantic similarity, if it is high, the weight between the two candidate keywords is high. For two candidate keywords, $v_i$, $v_j$, their vectors $\mathbf{v_i}$, $\mathbf{v_j}$ can be gained based on Word2vec. The semantic similarity $m_{ij}$ between them can be calculated according to cosine similarity, and the semantic similarity matrix is shown in equation (5).

$$M_{\alpha} = \left[ m_{ij} \right] \tag{5}$$

In terms of syntax, if the dependency relevance[15] between two candidate keywords is high, their weight will be high. Although the Word2vec-based TextRank algorithm[13] can achieve good results on some publicly available datasets, it may not always be valid when it comes to a single sentence. Therefore, from the syntactic perspective, we calculate the relevance by dependency parsing. This paper takes LTP[16] (Language Technology Platform) as the tool for dependency parsing.

Therefore, based on the length of the dependency relation path, the dependency relevance is calculated according to:

$$l_{ij}=1/len(v_i,v_j) \tag{6}$$

where $len(v_i,v_j)$ denotes the length of dependency relation path between $v_i$ and $v_j$. The dependency relevance matrix is shown in equation (7).

$$M_{\beta} = \left[ l_{ij} \right] \tag{7}$$

The probability transfer matrix $M$ is shown in equation (8).

$$M = \left[ p_{ij} \right] = [m_{ij} + l_{ij}] = M_{\alpha} + M_{\beta} \tag{8}$$

where $p_{ij}$ denotes the transfer probability from node $v_i$ to node $v_j$, and there is $\sum_{j=1}^{n} p_{ji} = 1$.

## 2.4. Multi Feature Fusion TextRank algorithm

With the initial words' scores in (4) and the probability transfer matrix in (8), the MFF (Multi Feature Fusion TextRank algorithm iteratively calculates the scores of nodes according to (2). Then the top $K$ candidate words can be selected as keywords.

The MFF TextRank algorithm is shown in Algorithm 1. The initial scores are assigned to the candidate keywords according to the word frequency and lexical features. Then the probability transfer matrix is constructed by considering both the semantic relationship characteristics between words and the syntactic relation among words to make the extracted keywords more accurate.

**Algorithm 1** Multi-Feature Fusion TextRank

```
Input: A sentence; the number of keywords to be extracted, $K$
Output: Top $K$ keywords
Step1: Pre-process the sentence: segment, remove the stop-words, and construct a candidate keyword set;
Step2: Calculate the initial scores for each candidate keyword $v_i$:
    (1) Calculate the term frequency $s^1(v_i)$;
    (2) Calculate the score of part of speech $s^2(v_i)$;
    (3) Get the initial scores $s_0(v_i)=s^1(v_i)*s^2(v_i)$;
Step3: Construct the graph with all candidate keywords as nodes, and if two candidate keywords appear in a co-occurrence
    window, there is an edge between them;
Step4: Construct the probability transfer matrix according to the semantic similarity and dependency relevance:
    (1) Calculate the semantic similarity between candidate keywords based on vectors gained with Word2vec and construct
    the probability transfer matrix $M_\alpha$;
    (2) Calculate the dependency relevance and construct the probability transfer matrix $M_\beta$;
```

(3) Obtain the final probability transfer matrix $M=M_{\alpha}+M_{\beta}$;
Step5: Iteratively calculate candidate keywords' scores until the termination condition is satisfied;
Step6: Sort the candidate keywords in descending order of score and extract the top $K$ words as the keywords.

## 3. Experiments and Results Analysis

The dataset in the experiments is from SogouCA[17], which is in Chinese and its size is about 1.4GB and covers 18 fields, including military, sports, society, entertainment, etc. The preprocessing of the dataset includes: (1) words segment and removing stop-words; (2) training a word vector model based on Gensim, which contains the Word2vec tool, and we obtained it with a size of about 160 MB.

We cross-labeled 500 sentences of hot topics randomly crawled from Baidu Knows (https://zhidao.baidu.com/) and Zhihu (https://www.zhihu.com/). In order to validate the reliability of the keyword extraction results, the keywords are extracted and manually cross-labeled. In the analysis of the experimental results, the extracted keywords are compared with the manually labeled keywords. The indices used in the experiments include the accuracy rate $P$, the recall rate $R$, and the *F-measure*.

The benchmark algorithms for experimental comparison include the TF-IDF algorithm (A1), the TextRank algorithm (A2), the TextRank algorithm with improved initial scores of candidate keywords (A3), the TextRank algorithm with improved probability transfer matrix by dependency parsing (A4), the TextRank algorithm with improved probability transfer matrix by Word2vec (A5), and the MFF TextRank (A6). When the number of extracted keywords is 1, 2, 3, and 4, the indices $P$, $R$, and *F-measure* are calculated, and the experimental results are shown in Table 1 and Figure 2.

**Table 1** Experimental results

| Algorithm | N=1 | | | N=2 | | | N=3 | | | N=4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F | P | R | F | P | R | F | P | R | F |
| A1 | 0.744 | 0.200 | 0.315 | 0.706 | 0.380 | 0.494 | 0.685 | 0.553 | 0.612 | 0.654 | 0.697 | 0.675 |
| A2 | 0.853 | 0.230 | 0.362 | 0.806 | 0.432 | 0.562 | 0.795 | 0.625 | 0.700 | 0.779 | 0.779 | 0.779 |
| A3 | 0.867 | 0.233 | 0.368 | 0.849 | 0.454 | 0.592 | 0.835 | 0.644 | 0.727 | 0.818 | 0.785 | 0.801 |
| A4 | 0.855 | 0.230 | 0.363 | 0.860 | 0.460 | 0.600 | 0.839 | 0.647 | 0.731 | 0.820 | 0.787 | 0.803 |
| A5 | 0.884 | 0.238 | 0.375 | 0.881 | 0.471 | 0.614 | 0.852 | 0.657 | 0.742 | 0.826 | 0.793 | 0.809 |
| A6 | **0.894** | **0.240** | **0.379** | **0.891** | **0.477** | **0.621** | **0.872** | **0.672** | **0.759** | **0.830** | **0.797** | **0.813** |

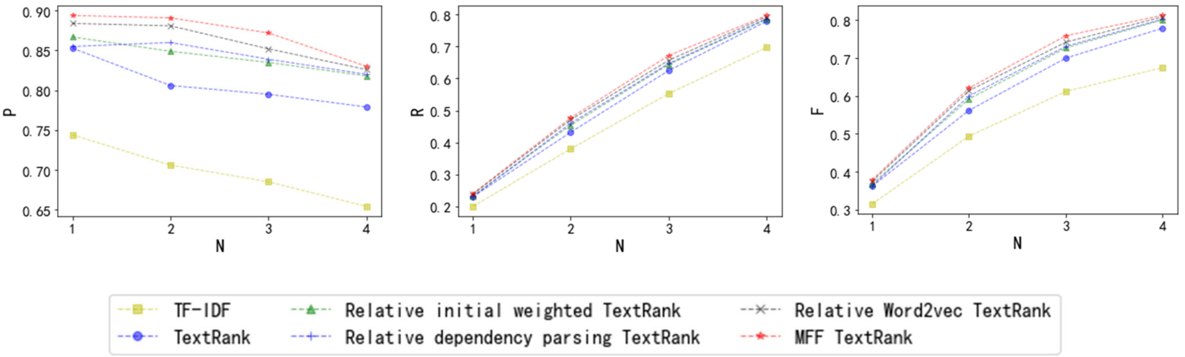

**Fig.2** Comparison of six algorithms

It can be seen from Figure 2 that the TextRank algorithm, which extracts keywords through co-occurrence relationships of words in the sentence, outperforms the TF-IDF algorithm, which selects keywords based on word frequency. After assigning the initial scores of candidate keywords based on term frequency and part of speech in TextRank, the performance of the algorithm is improved. It is further improved after integrating the dependency relevance in the probability transfer matrix in TextRank. After integrating the semantic similarity in the probability transfer matrix in TextRank, the algorithm performance becomes better. The MFF TextRank outperforms the above five algorithms in terms of the accuracy rate $P$, the recall rate $R$, and the *F-measure*.

The reasons that our algorithm outperforms other algorithms mainly include that: (1) More knowledge of word are integrated in the initial scores of candidate keywords, and the knowledge includes frequency and part of speech are introduced. (2) More knowledge of relationship between/among words are integrated into the construct of probability transfer matrices, and the knowledge includes the semantic relationship and the dependency relevance.

## 4. Conclusion

This paper proposes a multi-feature fusion TextRank algorithm for sentence-orient keyword extraction. To address the shortcomings that the canonical TextRank algorithm ignores the knowledge of keywords and the relationship between/among keywords, the TextRank algorithm is improved from two aspects: (1) The initial scores of candidate keywords are assigned by fusing the knowledge of the term frequency and part of speech. (2) The probability transfer matrix are calculated by fusing the knowledge of semantic relation between words, and the dependency relevance among words. The experiments show that our algorithm has better results in sentence keyword extraction.

Although the algorithm in this paper outperforms the other five algorithms in terms of *P*, *R*, and *F*, there is still room for improvement in terms of time complexity, and it is the future work to integrate suitable features to make the TextRank algorithm extract sentence keywords with higher performance.

## 5. Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (No. 62077029, 61673196, 62277030), Society Development Foundation of Xuzhou under Grant No. KC19213, Jiangsu Normal University Postgraduate Research and Practice Innovation Program Project (2021XKT1391).

## 6. References

[1] Firoozeh, N., Nazarenko, A., Alizon, F., Daille, B. (2020) Keyword extraction: Issues and methods., Natural Language Engineering., 26(3): 259-291.

[2] Yang, D.-H., Y, Wu, Y.-X., Fan, C.-X. (2020) Chinese Short Text Keyphrase Extraction Model Based on Attention. Computer Science., 47(1): 193-198.

[3] Lu, Y., Zhang, P.-Z., Zhang, C. (2019) Research on news keyword extraction technology based on TF-IDF and TextRank. In: Proceedings of the 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS). Beijing, pp. 452-455.

[4] Hu, J., Li, S., Yao, Yu, Y., L.-Y., Yang, G.-C., Hu, J.-J. (2018) Patent keyword extraction algorithm based on distributed representation for patent classification. Entropy., 20(2): 104.

[5] Li, J., Huang, G.-M., Fan, C.-L., Sun Z.-L., Zhu H.-T. (2019) Keyword extraction for short text via word2vec, doc2vec, and textrank. Turkish Journal of Electrical Engineering & Computer Sciences., 27(3): 1794-1805.

[6] Hu, Q., Hao, X.-Y., Zhang X.-Z., Chen Y.-W. (2016) Research on the Strategy of Keyword Extraction. Journal Of Taiyuan University Of Technology., 47(02): 228-232.

[7] Salton, G., Buckley, C. (1988) Term-weighting approaches in automatic text retrieval. Information processing & management., 24(5): 513-523.

[8] Blei, D. M., Ng, A. Y., Jordan, M. I. (2003) Latent dirichlet allocation. Journal of machine Learning research., pp. 3(Jan): 993-1022.

[9] Mihalcea, R., Tarau, P. (2004) Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing. Barcelona, pp. 404-411.

[10] Page, L., Brin, S., Motwani, R., Winograd, T. (1999) The PageRank citation ranking: Bringing order to the web. Stanford Digital Libraries Working Paper.

[11] Zhang, Z.-F., Liu, P.-P., Li X.-S. (2021) Keywords extraction algorithm of railway literature based on improved TextRank. Journal Of Beijing Jiaotong University., 45(02): 80-86.

[12] Xu, L. Text Keyword Extraction Method Based on Weighted TextRank. (2019) Computer Science., 46(S1): 142-145.

[13] Wen, Y., Yuan, H., Zhang, P. (2016) Research on keyword extraction based on word2vec weighted textrank. In: Proceedings of the 2016 2nd IEEE International Conference on Computer and

Communications (ICCC). Chengdu, pp. 2109-2113.

[14] Zhang, J.-E. (2013) Method for the Extraction of Chinese Text Keywords Based on Multi-Feature Fusion. Information studies: Theory & Application., 36(10): 105-108.

[15] Zhang, W.-N., Ming, Z.-Y., Zhang, Y., Nie, L.-Q., Liu, T., Chua, T.-S. (2012) The use of dependency relation graph to enhance the term weighting in question retrieval. In: Proceedings of the Proceedings of COLING. Mumbai, pp. 3105-3120.

[16] Che, W.-X., Feng, Y.-L., Qin, L.-B., Liu, T. (2010) Ltp: A chinese language technology platform. In: Proceedings of the Coling 2010: Demonstrations. Beijing, pp. 13-16.

[17] Wang, C., Zhangm M., Ma, S., Ru, L. (2008) Automatic online news issue construction in web environment. In: Proceedings of the Proceedings of the 17th international conference on World Wide Web. Beijing, pp. 457-466.