An answer summarization method based on keyword extraction

Qiaoqing Fan[1,a] and Yu Fang[1]

[1]*Department of Computer Science, Tongji University, Shanghai, China*

Abstract. In order to reduce the redundancy of answer summary generated from community q&a dataset without topic tags, we propose an answer summarization algorithm based on keyword extraction. We combine tf-idf with word vector to change the influence transferred ratio equation in TextRank. And then during summarizing, we take the ratio of the number of sentences containing any keyword to the total number of candidate sentences as an adaptive factor for AMMR. Meanwhile we reuse the scores of keywords generated by TextRank as a weight factor for sentence similarity computing. Experimental results show that the proposed answer summarization is better than the traditional MMR and AMMR.

1 Introduction

Due to the users of community question answering (QA) service have different language habits and knowledge levels, the focuses of different users' answers are various especially in consulting type and advising type questions. Different users' answers can be a complement to each other. So we can take advantage of multi-document summarization to merge these answers, but we have to pay more attention on reducing redundancy because these answers are highly similar to each other. This paper takes the keywords of a number of related answers as topic information and propose an answer summarization method based on keyword extraction from the perspective of reducing redundancy.

The key point of TextRank [1] is to construct an undirected graph where the influence of a node is transferred evenly. The work of this paper is to combine tf-idf[b] with word vector to change the influence transferred ratio equation in TextRank considering the impact of semantic and term frequency(TF) on keyword extraction. The answer summarization algorithm in this paper is based on Adaptive Maximum Marginal Relevance (AMMR) [2] and we take the keywords as the topic information for AMMR. We also reuse the influence scores of keywords derived from TextRank as a weight factor for sentence similarity computing to improve the quality of result.

The remainder of this paper is organized as follows. Section (2) provides a review of related works. Section (3) describes our keyword extraction method. Section (4) describes the answer summarization method based on keyword extraction. Section (5) shows the experimental result. The last section outlines conclusions and future directions.

[a] Qiaoqing Fan: 1433238@tongji.edu.cn

[b] https://en.wikipedia.org/wiki/Tf–idf

2 Related works

Research on keyword extraction has been ongoing for years. Mihalcea R et al. proposed TextRank based on PageRank. And then on the basis of TextRank, Li et al. [3] explored the use of tags for improving the performance of webpage keyword extraction task, Xia et al. [4] used the influence of word location to construct a word graph and calculate the probability transition matrix. In addition, there are some machine learning methods for keyword extraction, typically LDA [5]. The biggest difference between TextRank and LDA is that training corpus ahead is not required. Although LDA analyse the linkages between words form semantic level, it is required to retrain the model when new corpus comes. So TextRank is more simple in use. TextRank select only lexical units of a certain part of speech to construct an influence-evenly-transferred word graph. According to Matthew Effect[a], a word should get much more attention from neighbouring synonyms to highlight the importance of itself, so do high frequency words. Therefore, this paper will take term frequency and word sense into account for keyword extraction task.

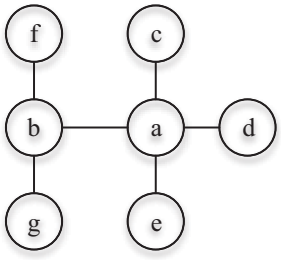
The summarization is defined into two ways according to the number of documents to be summarized, respectively are Single Document Summarization and Multiple Document Summarization. Extractive summarization and abstractive summarization approach are used [6]. The research object of this paper is Extractive Multiple Document Summarization.

Answer summarization was firstly proposed by Liu et al. [7]. Yin et al. developed a hierarchical clustering method to group similar questions and a ranking based summarization for representing an answer [8]. Wang et at. proposed a topic-centric answer summarization method called Adaptive Maximum Marginal Relevance(AMMR) in 2013 [2]. AMMR takes the e-mail headers and webpage tags as topic information to automatically adjust the weighting of topic relativity and redundancy when picking relevant sentences from candidate answers. However, not all QA datasets are labelled with such tags or have headers. Therefore, this paper proposes a keyword-extraction-centric answer summarization method based on Wang' work.

3 Keyword extractions

TextRank is derived from PageRank [9]. It splits text into a number of units. In keyword extraction the units are a set of words. All these words that are added into a graph and an edge are added between those words that co-occur within a window of n words. After the graph is constructed, the ranking algorithm described in [1] is run on the graph for several iterations until it converges. Once a final score is obtained for each unit, select the top m units with highest scores as keywords.

However, TextRank only construct an influence-evenly-transferred word graph. Considering that Figure 1 is a word graph of a certain document. In TextRank, the influence word b contributes to word a is one third of b because three nodes are linked with b. In fact, one word should transfer much more influence to the other if the other word is closer to it in semantic space. So we need an method to evaluate the semantic relationship between these words.



[a] https://en.wikipedia.org/wiki/Matthew_effect

Figure 1. A word graph of a document

Word2vec provides such a kind of evaluation method. It was proposed by Mikolov et al. in 2013 to compute continuous vector representations of words from very large datasets [10,11]. The vectors can be used to compute semantic similarity since it captures the context of a word during the process of training.

Formally, let $W=\{w_0,w_1,\ldots\ldots,w_{N-1}\}$ be the word set of document d. N is the number of words in W. $W^{'}\in W$ is the word set extracted from W, including nouns, verbs and words in the user dictionary. P is the numer of words in $W^{'}$. Each word in W can be represented in K-dimensional vector after training the corpus with word2vec. Let $\mathbf{vec_a}=[v_0,v_2,\ldots v_k\ldots,v_{K-1}],0\ll a<N,0\ll k<K$ denotes the vector of w_a. Then the semantic similarity of w_a and w_b can be measured by cosine similarity:

$$sem_{ab}=\frac{\sum_{k=0}^{K-1} vec_a[k]*vec_b[k]}{\sqrt{\sum_{k=0}^{K-1} vec_a[k]^2}\sqrt{\sum_{k=0}^{K-1} vec_b[k]^2}} \tag{1}$$

And the influence transferred ratio euqation in TextRank is re-weighted by the semantic similarity and re-normalized as shown below:

$$\mathrm{S}(b,a)=\frac{sem_{ab}}{\sum_{c\in Out(b)} sem_{cb}}S(b) \tag{2}$$

Let $Out(b)$ be the index set of words that w_b points to(successors) in word graph, $S(b)$ is the influence score of w_b. In addition, considering TF is lost in TextRank and a word should transfer much more influence to its adjacent high-frequency words. So we also introduce tf-idf for the influence transferred ratio equation (IDF is used to inhibit the high-frequency words which are widely exist in corpus). The variant of IDF we adopt is the same as BM25[a]. The final influence transferred ratio equation is shown below:

$$\mathrm{S}(b,a)=\frac{sem_{ab}*tfidf_a}{\sum_{c\in Out(b)} sem_{cb}*tfidf_c}S(b) \tag{3}$$

And the influence score of w_a is :

$$\mathrm{S(a)}=(1-\mathrm{d})+\mathrm{d}*\sum_{b\in In(a)}\mathrm{S}(b,a) \tag{4}$$

$In(a)$ is the index set of words that point to w_a.

Let m be the number of keywords to be extracted. The whole process to extract keywords is shown below:

1. Do word segmentation, part-of-speech tagging and removing stop words on document set to generate W。
2. Filter other words out except nouns, verbs and other words (in user dictionary) from W to generate W'.
3. Construct word graph G for W'.
4. The algorithm described above is run on G for several iterations until it converges.
5. Once a final score is obtained for each word in the graph, words are sorted in reversed order of their score, and the top m words in the ranking list are returned as keywords.

4 Answer summarization

Maximum Marginal Relevance (MMR) was proposed by Carbonell et al. in 1998 [12]. Generally, the diversity distribution over text content is different from one the other. Fixed linear interpolation of MMR cannot be well adapted to this, for selecting a sentence from candidate set whether tends to enhance diversity or topic relevance to user requirements. So Wang proposed AMMR from the perspective of topic cohesion of candidate sentences, which is based on an adaptive factor. The equation is shown below:

[a] https://en.wikipedia.org/wiki/Okapi_BM25

$$\text{AMMR} = \text{Arg}\max_{s_i \in R \setminus S}[(1 - \frac{M_T}{\text{M}}) * (Sim_1(s_i, q) - \frac{M_T}{\text{M}} * max_{s_j \in S} Sim_2(s_i, s_j)] \quad (5)$$

Here q is a query or user profile, R is the set of candidate summary sentences. S is the subset of sentences in R already selected. $R \setminus S$ is the set difference. Sim_1 and Sim_2 are the similarity metric used in sentences retrieval and relevance ranking between sentences. M is the number of the sentences in R. M_T is the number of the sentences containing any topic word (tags or headers instead in [2]) in R. It means that the topic cohesion of S is more compact when the ratio $\frac{M_T}{M}$ is larger and AMMR selecting a sentence from R will focus on reducing redundancy. Similarly, if $\frac{M_T}{M}$ is smaller, it means that the topic cohesion of S is less compact, AMMR will focus on enhancing topic relevance.

Keywords can specifically reflect topic information of a text because keyword contains key information of it. Here the M_T in AMMR is set to the number of the sentences containing any keyword where M is not changed. We can see that different answers have different M_T value. So it is an adaptive procedure and $\frac{M_T}{\text{M}}$ is the adaptive factor.

The similarity metric Sim_1 and Sim_2 also have an important influence on result quality beside the adaptive factor above. The similarity metric of this paper is the same as [12] used:

$$\text{sim}_{sem}(s_1, s_2) = \frac{\sum_{w_a \epsilon s_1} sim_m(w_a, s_2) + \sum_{w_b \epsilon s_2} sim_m(w_b, s_1)}{|s_1| + |s_2|} \quad (6)$$

$sim_m(w_a, s_1)$ is the word in s_1 that has the highest semantic similarity to the word w_a. $|s_1|$ is the number of words in s_1. The equation (1) is continue being used to compute word semantic similarity.

Two sentences are similar or not always depends on the nouns and verbs in them. Therefore, in order to highlight the impact of these words in computing sentences semantic similarity, this paper further makes use of the influence score of keyword derived from TextRank and proposes the following similarity calculation equation:

$$\text{sim}_{semk}(s_1, s_2) = \frac{\sum_{w_a \epsilon s_1} sim_m(w_a, s_2) * Score(w_a) + \sum_{w_b \epsilon s_2} sim_m(w_b, s_1) * Score(w_b)}{\sum_{w_a \epsilon s_1} Score(w) + \sum_{w_b \epsilon s_2} Score(w_b)} \quad (7)$$

$Score(w_a)$ is the influence score of w_a, it is derived from TextRank, just the value $S(a)$ (see equation 4). However, a part of words doesn't have such a score due to the algorithm we used to extract keywords only focus on nouns, verbs, and user dictionary words. So we do a smooth operation for these words. The meaning of m, W', W, P and N is not changed. $W'' = W - W'$, means the set difference of W and W'. Let $score[P]$ be the array of influence score in reversed order. The influence score of all words $Score(w_i)$ can be represented as :

$$Score(w_i) = \begin{cases} S(i) & w_i \epsilon W' \\ \frac{\sum_{j=m}^{P-1} score[j]}{P-m} & w_i \epsilon W'' \end{cases} \quad (8)$$

From the equation we can see that we used the average score of non-keywords to present the score of word in W''. This smooth operation further highlights the impact of keywords on sentence similarity computing.

Let l be the number of summary sentences to be extracted. The whole process of our answer summarization is:

1. Use the method we introduced in section (3) to extract keywords from candidate answer texts.

2. Set the value $\frac{\text{M}_T}{M}$ for AMMR and compute influence score for all words in candidate answer texts through equation (8).

3. Score all the sentences in candidate answer texts for l iterations using equation (5), at each iteration, sentence with highest score is selected and added into summary list。

5 Experimental results and analysis

The tokenizer we used is ICTCLAS, a Chinese word segmentation system. The data set used in the experiments is a collection of 44254 QA pairs from the 120ask website. In order to ensure the quality of word segmentation, we collected 132,371 medical terminologies from Sogou thesaurus, covering medicine name, disease name, anatomy, pharmaceutical company name, etc. Then after the text participle and removing stop words, we trained word2vec model with skip-gram using hierarchical softmax. The dimension of word vector was 50 and the window size was 5.

To evaluate our system, we selected 100 questions from the dataset, and got four to five most relevant questions and corresponding answers by Lucene[a] of each question. All these answers of each question were merged into one text as a candidate summary sentences set. Each candidate summary sentences set has an average of 19 sentences, and the 100 candidate summary sentences sets were distributed to five trained annotators to extract keywords and summaries. Finally, each candidate summary text has 4 keywords and 9 summary sentences on average.

5.1 Result of Keyword Extraction

All the other words were filtered out beside nouns, verbs, and user dictionary words. The d value of TextRank is set to 0.85 and window size is 10. ST-TextRank and S-TextRank are the keyword extraction method this paper introduced. The difference between them is that ST-TextRank adopts equation (3) as influence transferred ratio equation, but S-TextRank adopts equation (2). We compared our method with traditional TextRank and [8] (denoted as P-TextRank). P-TextRank achieved the best result after several attempts when set λ to 80, α to 0.6, β to 0.15 and γ to 0.25. The evaluation metrics we used are Precision, Recall and F-measure and the experimental results are compared in the extraction of 3, 5 and 10 key words. Results are shown in Table 1.

Table 1. Comparison of the performances of different keyword extraction algorithms

Algorithm	Numbers of keywords	Precision	Recall	F-measure
ST-TextRank	3	0.520	0.359	0.425
	5	0.406	0.463	0.433
	10	0.266	0.601	0.369
S-TextRank	3	0.433	0.304	0.357
	5	0.350	0.402	0.374
	10	0.258	0.582	0.358
P-TextRank	3	0.510	0.362	0.423
	5	0.400	0.468	0.431
	10	0.280	0.643	0.390
TextRank	3	0.407	0.290	0.339
	5	0.354	0.415	0.382
	10	0.255	0.583	0.355

From Table 1 we can observe that S-TextRank is better than TextRank, it proves the validity of word vector we use in influence transferred ratio equation. ST-TextRank outperforms all the other algorithms in all metrics when the number of keywords is 3, it proves the validity of the combination of word vector and tf-idf in influence transferred ratio equation. Although the result P-TextRank is not as good as ST-TextRank, but better than S-TextRank, evenly better than ST-TextRank when the

[a] https://lucene.apache.org/

number of keywords is 5. We think the main reason is that it is difficult for P-TextRank to coordinate all the weight factors to make the model achieve best.

5.2 Result of answer summarization

We followed the same evaluation metrics for answer summarization as Wang adopted in [2] (Accuracy, Redundancy and Summary Quality). Accuracy is a proportion of all correctly predicted sentences compared to all sentences. Redundancy reflects how much information an answer summary contains. Summary Quality is the difference between Accuracy and Redundancy. Here equation (6) was adopted as similarity computation for Redundancy. We compared our method with MMR in different λ values and Wang's AMMR . For the reason that Wang took the topic tags and e-mail headers as the topic information and our QA data set has not been labelled with such tags, or has headers. So a similar approach was adopted, we took the words from question after segmentation as headers when we implemented the algorithm (denoted as Q-AMMR). QF-AMMR is the implementation of Q-AMMR with removing stop words from headers which just contain nouns, verbs and user dictionary words. KW-AMMR and K-AMMR are the algorithms based on keyword extraction we introduced in this paper. They are different in sentence similarity computation, where KW-AMMR takes equation (7) and K-AMMR takes equation (6). In addition, according to table 1, ST-TextRank got best when the number of keywords was set to 3, therefore we chose 3 as the number of keywords for answer summarization. The experimental results are shown in table 2.

From table 2, we can see that KW-AMMR and K-AMMR outperform all the other algorithms, it proves the validity of our thought to take keywords as topic information in AMMR. KW-AMMR outperforms K-AMMR not only in precision, but also redundancy, it proves the usefulness of the equation (7) in similarity calculation for reducing redundancy. Although the redundancy of KW-AMMR is higher than MMR when the λ is set to 0.1 (or 0.2), but the accuracy is too low. This just shows the efficiency of adaptive factor used in AMMR compared with MRR in different λ values from 0.1 to 0.9.

We also notice that although the accuracy of Q-AMMR and QF-AMMR are maintained at a high level, the redundancy and summary quality do not achieve a better result. This is mainly caused by the following reasons: a) the M_t is too small (even zero) after segmenting the question into words and removing stop words; b) when the users write down their answers, a lot of synonyms and other words are expanded, which are widely exist in other sentences, and such a class of words that can reflect the topic information are not captured during keywords extraction. To summarize, the words of question do not have the ability to fully reflect the topic information, so there is a high rate of redundancy in QF-AMMR and Q-AMMR.

Table 2. Comparison of the performances of different answer summarization algorithms

Algorithm		Precision	Redundancy	Summary Quality
KW-AMMR		0.618	0.1361	0.4819
K-AMMR		0.604	0.1362	0.4678
Q-AMMR		0.582	0.1510	0.431
QF-AMMR		0.574	0.1568	0.4172
MMR	λ=0.1	0.554	0.1089	0.4451
	λ=0.2	0.564	0.1104	0.4536
	λ=0.3	0.562	0.1122	0.4498
	λ=0.4	0.580	0.1207	0.4593
	λ=0.5	0.574	0.1367	0.4373
	λ=0.6	0.570	0.1543	0.4157

λ=0.7	0.586	0.1633	0.4227
λ=0.8	0.574	0.1688	0.4052
λ=0.9	0.570	0.1735	0.3965

6 Conclusion

An answer summary method is proposed this paper. We combine tf-idf with word vector to change the influence transferred ratio equation in TextRank and take the ratio of the number of sentences containing any keyword to the total number of candidate sentences as an adaptive factor for AMMR. Also, we reuse the scores of keywords generated from TextRank as a weight factor for sentence similarity computing. The experimental results show that the method based on keyword extraction in this paper has achieved great results in improving accuracy and reducing redundancy.

The next step we will focus on the dynamic determination of keywords' number, rather than a fixed value and optimizing the quality of the answer summary, for instance providing context information for the summary and making the sentences more fluent.

References

1. R. Mihalcea, P. Tarau, TextRank: Bringing order into texts, *Conference on Empirical Methods in Natural Language Processing*, 404(2004)
2. B. Wang, Research on the semantic mining of question-answer pairs in web communities, Harbin Institute of Technology (Doctoral dissertation, 2013)
3. P. Li, B. Wang, Z. Shi, et al, Tag-TextRank: A Webpage Keyword Extraction Method Based on Tags, Journal of Computer Research and Development, **49**, 2344 (2012)
4. T. Xia, Study on Keyword Extraction Using Word Position Weighted TextRank, New Technology of Library & Information Service, 30 (2013)
5. L. Zhang, H. Wang, Research on Sentence Extraction in Text Summarization, Journal of Chinese Information Processing, **26**, 97 (2012)
6. Y. Liu, S. Li, Y. Cao, et al, Understanding and summarizing answers in community-based question answering services, Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1, Association for Computational Linguistics, **1**, 497 (2008)
7. Y. Yin, Y. Zhang, Liu X, et al, HealthQA: A Chinese QA Summary System for Smart Health, *ICSH*, 51 (2014)
8. L. Page, S. Brin, Motwani R, et al, The PageRank citation ranking: bringing order to the web, Stanford InfoLab, **9**,1 (1999)
9. T. Mikolov, K. Chen, G. Corrado, et al, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301 (2013)
10. T. Mikolov, I. Sutskever, K. Chen, et al, Distributed representations of words and phrases and their compositionality, Advances in neural information processing systems, **26**, 3111 (2013)
11. J. Carbonell, J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries", *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, 335 (1998)
12. R. Malik, L. Subramaniam, S. Kaushik, Automatically Selecting Answer Templates to Respond to Customer Emails, IJCAI, **7**, 1659 (2007)