A Novel Document Summarizatioin Algorithm

Wang Chao Yao Yujian

Nov 2013

Abstract

We propose a novel automatic document summarization algorithm *SenRank* that finds relation between sentences by counting word overlaps and shared references, then makes use of PageRank scores derived from such relation to generate a summary. Experiments show that it can produce summaries of reasonable quality. Moreover, the quality of the summary depends on the relative weight of co-references information.

1 Introduction

The explosive growth of the Internet has resulted in the overwhelming abundance of documents online, making the information filtering process increasingly difficult for human users. As a result, it is necessary to develop a fast way to generate a short piece of text covering all main topics and relevant points of a document. Automatic document summarization can be helpful for such processes.

In this paper, we propose a novel automatic document summarization algorithm called *SenRank*. In short, SenRank treats the article as a graph of ideas and uses PageRank on sentences to extract main text.

Then we use the ROUGE(Recall-Oriented Understudy for Gisting Evalution) toolkit to test SenRank. The results show that the quality of generated summary by SenRank is reasonably good.

2 Formulation

2.1 Assumptions

We start the formulation with the following assumptions:

1. Articles consist of ideas

2. Ideas flow
3. Each sentence presents some ideas
4. Ideas are expressed through words and references
5. Overlap of words signifies relation, thus flow, of ideas

We can then assume that a sentence is significant and expressing the main ideas, thus *summarizing*, if ideas from many other sentences flow to it.

Thus, intuitively, by assembling a set of *summarizing* sentences, we can produce a good summary. Further, to shorten the length of the summary, we should try to minimize overlaps among sentences in this set.

2.2 Construction of the article graph

With these assumptions, we formulate the article as a graph of sentences.

Formally, we can build a complete graph $\mathcal{G} = (V, E)$, where V is the set of sentences, and $\forall(u, v) \in E, weight(e) = relation(u, v)$.

Further, we define $relation(u, v)$ to be the number of word overlaps (where only a subset of all words are considered) plus the number of shared references (by making use of co-reference). Moreover, we assume that $relation(u, v) = relation(v, u)$, as both word-overlap and co-references has no direction.

With the graph, we can then compute the probability of ideas flowing to a sentence with the PageRank *(Page et al. (1999))* algorithm. Then we can rank the sentences with their PageRank score and aggregate the top few non-overlapping sentences as the summary.

3 Algorithm

With the graph formulation, SenRank works by following the steps detailed in this section.

3.1 Tokenization and derivation of co-references

We use Stanford Corenlp *(Recasens, Marneffe, and Potts (2013), Lee et al. (2013), Lee et al. (2011), Raghunathan et al. (2010))* to tokenize the article and then generate the co-reference data.

3.2 Parsing co-references

The co-reference data obtained from Stanford Corenlp is a list of tuples: $\{(c_1, c_2, ..., c_n), (d_1, d_2, ..., d_n)...\}$, where c_i, d_i refers to a sentence in the article.

Each tuple represents the occurrences of one entity in sentences. So $(c_1, c_2, ..., c_n)$ means that a particular entity appears in Sentences $c_1, c_2, ...c_n$. Moreover, there can be some i, j such that $i \neq j$ and $c_i = c_j$ in some tuples, i.e., if an entity appears for multiple times in a sentence, that sentence will appear for multiple times in the tuple.

SenRank then use this co-reference data to build an adjacency matrix

$$A_{ij} = a_{ij} + a_{ji}$$

where a_{ij} is the number of times Sentence i appearing in tuples that also contain Sentence j.

For example, we may have obtained the co-reference data $(1, 1, 2, 3), (1, 3)$. Then $a_{12} = 2$, $a_{21} = 1$, thus $A_{12} = A_{21} = 1 + 2 = 3$

3.3 Counting word overlaps

SenRank then process all sentences and form another adjacency matrix

$$B_{ij} = \sum_{w \in W} w_i w_j$$

where W is the set of words that are considered for word overlaps and w_i refers to the number of times Word w appears in Sentence i.

3.4 Building the article graph

With A and B, SenRank builds the final adjacency matrix

$$C = k_a A + k_b B$$

where k_a and k_b are some constants that can be tweaked for optimal performance.

3.5 Running PageRank

SenRank then runs the PageRank algorithm on C to obtain PageRank scores for all sentences.

3.6 Generating a summary

Finally SenRank generates a summary with the following algorithm:

Firstly we define m as the threshold such that $weight(u, v) > m$ means that u and v are so similar that they cannot both appear in the summary.

Then the algorithm proceeds as the following:

1. S is the set of all sentences
2. R is the set of sentences in the summary, initially $R = \emptyset$
3. Loop while $S \neq \emptyset$
    1. Pick $s \in S$ such that s has the highest score
    2. $R = R \cup \{s\}$
    3. Let $T = \{i \mid C_{si} > m\}$
    4. $S = S - T$
    5. If R exceeds the word limit, trancate the last sentence added to R to match the word limit
    6. If R matches the word limit, terminate the loop
4. Output sentences in R according to their original order in the article

The output is the generated summary.

4 Experiment Setup

We test our algorithm on 178 papers in Scholarly Paper Recommendation Dataset *(Sugiyama and Kan (2013))*. The abstracts are used as model summaries and the contents are used as input articles. We then use ROUGE *(Lin (2004))* to evaluate the quality of the generated summaries. We use the R-score of ROGUE since the model summaries do not have a word limit.

We setup our algorithm such that the word limit is 200 and $m = 20$.

Moreover, we run the experiments on the following algorithms:

1. Random: randomly pick sentences from the article until word limit is matched.
2. Degree: Set $k_a = 0$, so we do not consider co-reference. Further, we do not run PageRank and instead rank sentences by their degrees in the graph (sum of the row in C).
3. PageRank: Set $k_a = 0$. Again we do not consider co-references.
4. Coref_equal: Set $k_a = 1$ and $k_b = 1$.
5. Coref_twice: Set $k_a = 2$ and $k_b = 1$.

6. Coref_large: Set $k_a = 5$ and $k_b = 1$.
7. Coref_only: Set $k_a = 1$ and $k_b = 0$, so we only consider co-reference but not word overlaps.

5 Results

Algorithm	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-L
Random	0.266158	0.056647	0.015765	0.251459
Degree	0.302039	0.079483	0.023131	0.279868
PageRank	0.308187	0.080359	0.024296	0.285539
Coref_equal	0.312450	0.078048	0.022415	0.287519
Coref_twice	0.321284	0.083117	0.025682	0.294462
Coref_large	0.312604	0.080656	0.025286	0.288393
Coref_only	0.287429	0.072876	0.022825	0.268019

Table 1: Mean values of ROUGE-R Scores

Test	Random < Degree	Degree < PageRank	PageRank < Coref_equal	PageRank < Coref_twice	PageRank < Coref_large	Coref_only < PageRank
p-value	0.000	0.055	0.307	0.032	0.307	0.008

Table 2: 1-tailed paired t-tests on ROUGE-L scores on various algorithms

This shows that the SenRank algorithm (even the stripped down version) can produce summaries of reasonable quality. Moreover, by t-test results, we can see that PageRank seems to perform better than Degree, although the difference is not very significant.

Furthermore, the coefficients k_a and k_b affect the performance of SenRank, as coref_twice performs the best while the other algorithms that uses co-references does not perform significantly better than PageRank and word overlaps alone. Interestingly, co-reference alone is even worse than using word overlaps. While

this can mean that co-reference data alone cannot adequately represent the relation between sentences, it can also be due to the fact that the co-reference resolution is not good enough. Moreover, this also shows that machine learning can be applied to derive an optimal set of variables used in SenRank to achieve better performance.

6 Related works

Two similar PageRank-based document summarization algorithms have been independently proposed previously. They both run PageRank on a graph constructed from sentences, then aggregate the highest ranking sentences to produce a summary.

TextRank *(Mihalcea and Tarau (2004))* was developed as a model for general text processing and it allows one to define the vertices as any text units as the task requires. A TextRank solution for document summarization was proposed by defining sentences as vertices and also using word overlaps as relations, although the authors used a different formula for counting word overlaps and normalized the result by the length of two sentences.

LexRank *(Erkan and Radev (2004))* employs similar methods, but use consine similarity as the measure for relation. Moreover, it proposed an additional method in constructing the graph. Other than using the weighted edges directly, LexRank can also build an unweighted graph by cutting off edges whose weights are below a threshold. These two methods, however, did not result in statistically significant results.

However, in contrast to SenRank, both methods do not consider the case where similar sentences may be extracted. This means that these two methods may not be able to succinctly summarize articles that tend to repeat ideas often. With the DUC dataset that both papers run experiments on, this may not be a problem, but it will likely be so for scientific articles. Moreover, both algorithms in the papers use only the textual information without any further processing. This means that they cannot account for similarity between different tokens that actually refer to the same entity, and thus the graphs built may not be representative of the article.

Generally, PageRank-based methods have the advantage of not requiring supervised learning. Other methods usually applies machine learning to derive a function to map a set of various features such as sentences position, occurrence frequency of words etc. to a score, and thus strongly rely on the availability of training sets. (Das and Martins (2007)) While machine learning can improve the quality of summarization of SenRank, it is not necessary.

7 Conclusion

In this paper, we propose a novel automatic document summarization algorithm SenRank which models an article as a graph of sentences related to each other. We then construct the weight of edges by using co-reference and counting overlap of words. After which we can run PageRank on this weighted graph and generate a summary by aggregating a number of top non-overlaping sentences. The experiment results show that the quality of generated summary is reasonable, and that the co-efficients used in constructing the edges affect the performance of the algorithm.

However, the dataset we test only includes science papers which are usually long and very well structured. It may be worthwhile to investigate the performance of SenRank on different types of articles such as news articles.

To better assess the performance of SenRank, we should apply machine learning techniques to derive a good set of variables k_a, k_b and m, also, other co-reference resolution algorithms can be tested.

References

Das, Dipanjan, and André FT Martins. 2007. "A Survey on Automatic Text Summarization." *Literature Survey for the Language and Statistics II Course at CMU* 4: 192–95.

Erkan, Günes, and Dragomir R Radev. 2004. "LexRank: Graph-Based Lexical Centrality as Salience in Text Summarization." *J. Artif. Intell. Res.(JAIR)* 22 (1): 457–79.

Lee, Heeyoung, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. "Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules." MIT Press.

Lee, Heeyoung, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. "Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task." *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, 28–34.

Lin, Chin-Yew. 2004. "Rouge: A Package for Automatic Evaluation of Summaries." In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 74–81.

Mihalcea, Rada, and Paul Tarau. 2004. "TextRank: Bringing Order into Texts." In *Proceedings of EMNLP*. Vol. 4. 4. Barcelona, Spain.

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. "The PageRank Citation Ranking: bringing Order to the Web." Stanford InfoLab.

Raghunathan, Karthik, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. "A Multi-Pass Sieve for Coreference Resolution." *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 492–501.

Recasens, Marta, Marie-Catherine de Marneffe, and Christopher Potts. 2013. "The Life and Death of Discourse Entities: Identifying Singleton Mentions." In *Proceedings of NAACL-HLT*, 627–33.

Sugiyama, Kazunari, and Min-Yen Kan. 2013. "Exploiting Potential Citation Papers in Scholarly Paper Recommendation." *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*. ACM, 153–62.