Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization

Rada Mihalcea
Department of Computer Science
University of North Texas
rada@cs.unt.edu

Abstract

This paper presents an innovative unsupervised method for automatic sentence extraction using graph-based ranking algorithms. We evaluate the method in the context of a text summarization task, and show that the results obtained compare favorably with previously published results on established benchmarks.

1 Introduction

Graph-based ranking algorithms, such as Kleinberg's HITS algorithm (Kleinberg, 1999) or Google's PageRank (Brin and Page, 1998), have been traditionally and successfully used in citation analysis, social networks, and the analysis of the link-structure of the World Wide Web. In short, a graph-based ranking algorithm is a way of deciding on the importance of a vertex within a graph, by taking into account global information recursively computed from the entire graph, rather than relying only on local vertex-specific information.

A similar line of thinking can be applied to lexical or semantic graphs extracted from natural language documents, resulting in a graph-based ranking model called TextRank (Mihalcea and Tarau, 2004), which can be used for a variety of natural language processing applications where knowledge drawn from an entire text is used in making local ranking/selection decisions. Such text-oriented ranking methods can be applied to tasks ranging from automated extraction of keyphrases, to extractive summarization and word sense disambiguation (Mihalcea et al., 2004).

In this paper, we investigate a range of graph-based ranking algorithms, and evaluate their application to automatic unsupervised sentence extraction in the context of a text summarization task. We show that the results obtained with this new unsupervised method are competitive with previously developed state-of-the-art systems.

2 Graph-Based Ranking Algorithms

Graph-based ranking algorithms are essentially a way of deciding the importance of a vertex within a graph, based on information drawn from the graph structure. In this section, we present three graph-based ranking algorithms – previously found to be successful on a range of ranking problems. We also show how these algorithms can be adapted to undirected or weighted graphs, which are particularly useful in the context of text-based ranking applications.

Let $G = (V, E)$ be a directed graph with the set of vertices V and set of edges E, where E is a subset of $V \times V$. For a given vertex V_i, let $In(V_i)$ be the set of vertices that point to it (predecessors), and let $Out(V_i)$ be the set of vertices that vertex V_i points to (successors).

2.1 HITS

HITS (Hyperlinked Induced Topic Search) (Kleinberg, 1999) is an iterative algorithm that was designed for ranking Web pages according to their degree of "authority". The HITS algorithm makes a distinction between "authorities" (pages with a large number of incoming links) and "hubs" (pages with a large number of outgoing links). For each vertex, HITS produces two sets of scores – an "authority" score, and a "hub" score:

$$HITS_A(V_i) = \sum_{V_j \in In(V_i)} HITS_H(V_j) \tag{1}$$

$$HITS_H(V_i) = \sum_{V_j \in Out(V_i)} HITS_A(V_j) \tag{2}$$

2.2 Positional Power Function

Introduced by (Herings et al., 2001), the positional power function is a ranking algorithm that determines the score of a vertex as a function that combines both the number of its successors, and the score of its successors.

$$POS_P(V_i) = \frac{1}{|V|} \sum_{V_j \in Out(V_i)} (1 + POS_P(V_j)) \tag{3}$$

The counterpart of the positional power function is the positional weakness function, defined as:

$$POS_W(V_i) = \frac{1}{|V|} \sum_{V_j \in In(V_i)} (1 + POS_W(V_j)) \tag{4}$$

2.3 PageRank

PageRank (Brin and Page, 1998) is perhaps one of the most popular ranking algorithms, and was designed as a method for Web link analysis. Unlike other ranking algorithms, PageRank integrates the impact of both incoming and outgoing links into one single model, and therefore it produces only one set of scores:

$$PR(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{PR(V_j)}{|Out(V_j)|} \qquad (5)$$

where d is a parameter that is set between 0 and 1 [1].

For each of these algorithms, starting from arbitrary values assigned to each node in the graph, the computation iterates until convergence below a given threshold is achieved. After running the algorithm, a score is associated with each vertex, which represents the "importance" or "power" of that vertex within the graph. Notice that the final values are not affected by the choice of the initial value, only the number of iterations to convergence may be different.

2.4 Undirected Graphs

Although traditionally applied on directed graphs, recursive graph-based ranking algorithms can be also applied to undirected graphs, in which case the out-degree of a vertex is equal to the in-degree of the vertex. For loosely connected graphs, with the number of edges proportional with the number of vertices, undirected graphs tend to have more gradual convergence curves. As the connectivity of the graph increases (i.e. larger number of edges), convergence is usually achieved after fewer iterations, and the convergence curves for directed and undirected graphs practically overlap.

2.5 Weighted Graphs

In the context of Web surfing or citation analysis, it is unusual for a vertex to include multiple or partial links to another vertex, and hence the original definition for graph-based ranking algorithms is assuming unweighted graphs.

However, in our TextRank model the graphs are build from natural language texts, and may include multiple or partial links between the units (vertices) that are extracted from text. It may be therefore useful to indicate and incorporate into the model the "strength" of the connection between two vertices V_i and V_j as a weight w_{ij} added to the corresponding edge that connects the two vertices.

Consequently, we introduce new formulae for graph-based ranking that take into account edge weights when computing the score associated with a vertex in the graph.

$$HITS_A^W(V_i) = \sum_{V_j \in In(V_i)} w_{ji} HITS_H^W(V_j) \qquad (6)$$

$$HITS_H^W(V_i) = \sum_{V_j \in Out(V_i)} w_{ij} HITS_A^W(V_j) \qquad (7)$$

$$POS_P^W(V_i) = \frac{1}{|V|} \sum_{V_j \in Out(V_i)} (1 + w_{ij} POS_P^W(V_j)) \qquad (8)$$

$$POS_W^W(V_i) = \frac{1}{|V|} \sum_{V_j \in In(V_i)} (1 + w_{ji} POS_W^W(V_j)) \qquad (9)$$

$$PR^W(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} w_{ji} \frac{PR^W(V_j)}{\sum_{V_k \in Out(V_j)} w_{kj}} \qquad (10)$$

While the final vertex scores (and therefore rankings) for weighted graphs differ significantly as compared to their unweighted alternatives, the number of iterations to convergence and the shape of the convergence curves is almost identical for weighted and unweighted graphs.

3 Sentence Extraction

To enable the application of graph-based ranking algorithms to natural language texts, TextRank starts by building a graph that represents the text, and interconnects words or other text entities with meaningful relations. For the task of sentence extraction, the goal is to rank entire sentences, and therefore a vertex is added to the graph for each sentence in the text.

To establish connections (edges) between sentences, we are defining a "similarity" relation, where "similarity" is measured as a function of content overlap. Such a relation between two sentences can be seen as a process of "recommendation": a sentence that addresses certain concepts in a text, gives the reader a "recommendation" to refer to other sentences in the text that address the same concepts, and therefore a link can be drawn between any two such sentences that share common content.

The overlap of two sentences can be determined simply as the number of common tokens between the lexical representations of the two sentences, or it can be run through syntactic filters, which only count 句法 words of a certain syntactic category. Moreover, to avoid promoting long sentences, we are using a normalization factor, and divide the content overlap of two sentences with the length of each sentence. Formally, given two sentences S_i and S_j, with a sentence being represented by the set of N_i words that appear in the sentence: $S_i = W_1^i, W_2^i, ..., W_{N_i}^i$, the similarity of S_i and S_j is defined as:

$$Similarity(S_i, S_j) = \frac{|W_k | W_k \in S_i \& W_k \in S_j|}{log(|S_i|) + log(|S_j|)}$$

The resulting graph is highly connected, with a weight associated with each edge, indicating the

这个相似度是不是可以改改函数.

[1] The factor d is usually set at 0.85 (Brin and Page, 1998), and this is the value we are also using in our implementation.

应该是把纯文本文件里面的每一个句子之间都看做默认相连接.
然后算similarity系数,如果大于0,就表示连接生效.然后计算后的图里面
各个句子的权重就保持这些比例系数.

strength of the connections between various sentence pairs in the text[2]. The text is therefore represented as a weighted graph, and consequently we are using the weighted graph-based ranking formulae introduced in Section 2.5. The graph can be represented as: (a) simple *undirected* graph; (b) directed weighted graph with the orientation of edges set from a sentence to sentences that follow in the text (*directed forward*); or (c) directed weighted graph with the orientation of edges set from a sentence to previous sentences in the text (*directed backward*).

After the ranking algorithm is run on the graph, sentences are sorted in reversed order of their score, and the top ranked sentences are selected for inclusion in the summary.

Figure 1 shows a text sample, and the associated weighted graph constructed for this text. The figure also shows sample weights attached to the edges connected to vertex 9[3], and the final score computed for each vertex, using the PR formula, applied on an undirected graph. The sentences with the highest rank are selected for inclusion in the abstract. For this sample article, sentences with id-s 9, 15, 16, 18 are extracted, resulting in a summary of about 100 words, which according to automatic evaluation measures, is ranked the second among summaries produced by 15 other systems (see Section 4 for evaluation methodology).

4 Evaluation

The TextRank sentence extraction algorithm is evaluated in the context of a single-document summarization task, using 567 news articles provided during the Document Understanding Evaluations 2002 (DUC, 2002). For each article, TextRank generates a 100-words summary — the task undertaken by other systems participating in this single document summarization task.

For evaluation, we are using the ROUGE evaluation toolkit, which is a method based on Ngram statistics, found to be highly correlated with human evaluations (Lin and Hovy, 2003a). Two manually produced reference summaries are provided, and used in the evaluation process[4].

3: BC−HurricaineGilbert, 09−11 339
4: BC−Hurricaine Gilbert, 0348
5: Hurricaine Gilbert heads toward Dominican Coast
6: By Ruddy Gonzalez
7: Associated Press Writer
8: Santo Domingo, Dominican Republic (AP)
9: Hurricaine Gilbert Swept towrd the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains, and high seas.
10: The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.
11: "There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly after midnight Saturday.
12: Cabral said residents of the province of Barahona should closely follow Gilbert's movement.
13: An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo.
14. Tropical storm Gilbert formed in the eastern Carribean and strenghtened into a hurricaine Saturday night.
15: The National Hurricaine Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.
16: The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westard at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm.
17. The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.
18: Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds, and up to 12 feet to Puerto Rico's south coast.
19: There were no reports on casualties.
20: San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night.
21: On Saturday, Hurricane Florence was downgraded to a tropical storm, and its remnants pushed inland from the U.S. Gulf Coast.
22: Residents returned home, happy to find little damage from 90 mph winds and sheets of rain.
23: Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane.
24: The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.

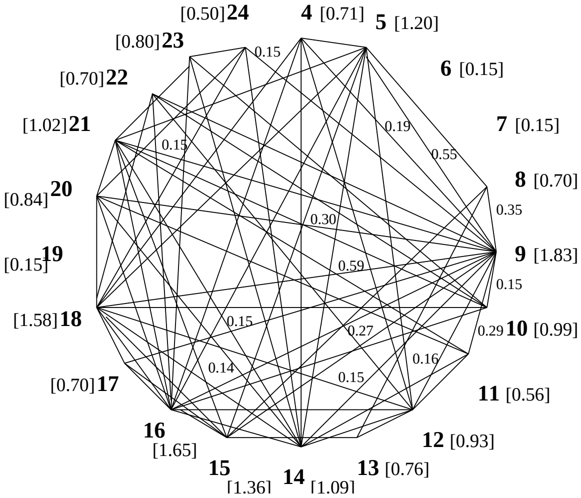

Figure 1: Sample graph build for sentence extraction from a newspaper article.

We evaluate the summaries produced by TextRank using each of the three graph-based ranking algorithms described in Section 2. Table 1 shows the results obtained with each algorithm, when using graphs that are: (a) undirected, (b) directed forward, or (c) directed backward.

For a comparative evaluation, Table 2 shows the results obtained on this data set by the top 5 (out of 15) performing systems participating in the single document summarization task at DUC 2002 (DUC, 2002). It also lists the baseline performance, computed for 100-word summaries generated by taking the first sentences in each article.

Discussion. The TextRank approach to sentence extraction succeeds in identifying the most important sentences in a text based on information exclusively

[2] In single documents, sentences with highly similar content are very rarely if at all encountered, and therefore sentence redundancy does not have a significant impact on the summarization of individual texts. This may not be however the case with multiple document summarization, where a redundancy removal technique – such as a maximum threshold imposed on the sentence similarity – needs to be implemented.

[3] Weights are listed to the right or above the edge they correspond to. Similar weights are computed for each edge in the graph, but are not displayed due to space restrictions.

[4] The evaluation is done using the Ngram(1,1) setting of ROUGE, which was found to have the highest correlation with human judgments, at a confidence level of 95%. Only the first 100 words in each summary are considered.

Algorithm	Graph		
	Undirected	Dir. forward	Dir. backward
$HITS_A^W$	0.4912	0.4584	**0.5023**
$HITS_H^W$	0.4912	**0.5023**	0.4584
POS_P^W	0.4878	0.4538	0.3910
POS_W^W	0.4878	0.3910	0.4538
$PageRank$	0.4904	0.4202	**0.5008**

Table 1: Results for text summarization using TextRank sentence extraction. Graph-based ranking algorithms: HITS, Positional Function, PageRank. Graphs: undirected, directed forward, directed backward.

Top 5 systems (DUC, 2002)					Baseline
S27	S31	S28	S21	S29	
0.5011	0.4914	0.4890	0.4869	0.4681	0.4799

Table 2: Results for single document summarization for top 5 (out of 15) DUC 2002 systems, and baseline.

drawn from the text itself. Unlike other supervised systems, which attempt to learn what makes a good summary by training on collections of summaries built for other articles, TextRank is fully unsupervised, and relies only on the given text to derive an extractive summary.

Among all algorithms, the $HITS_A$ and $PageRank$ algorithms provide the best performance, at par with the best performing system from DUC 2002[5]. This proves that graph-based ranking algorithms, previously found successful in Web link analysis, can be turned into a state-of-the-art tool for sentence extraction when applied to graphs extracted from texts.

Notice that TextRank goes beyond the sentence "connectivity" in a text. For instance, sentence 15 in the example provided in Figure 1 would not be identified as "important" based on the number of connections it has with other vertices in the graph[6], but it is identified as "important" by TextRank (and by humans – according to the reference summaries for this text).

Another important advantage of TextRank is that it gives a ranking over all sentences in a text – which means that it can be easily adapted to extracting very short summaries, or longer more explicative summaries, consisting of more than 100 words.

5 Related Work

Sentence extraction is considered to be an important first step for automatic text summarization. As a consequence, there is a large body of work on algorithms for sentence extraction undertaken as part of the DUC evaluation exercises. Previous approaches include supervised learning (Teufel and Moens, 1997), vectorial similarity computed between an initial abstract and sentences in the given document, or intra-document similarities (Salton et al., 1997). It is also notable the study reported in (Lin and Hovy, 2003b) discussing the usefulness and limitations of automatic sentence extraction for summarization, which emphasizes the need of accurate tools for sentence extraction, as an integral part of automatic summarization systems.

6 Conclusions

Intuitively, TextRank works well because it does not only rely on the local context of a text unit (vertex), but rather it takes into account information recursively drawn from the entire text (graph). Through the graphs it builds on texts, TextRank identifies connections between various entities in a text, and implements the concept of *recommendation*. A text unit recommends other related text units, and the strength of the recommendation is recursively computed based on the importance of the units making the recommendation. In the process of identifying important sentences in a text, a sentence recommends another sentence that addresses similar concepts as being useful for the overall understanding of the text. Sentences that are highly recommended by other sentences are likely to be more informative for the given text, and will be therefore given a higher score.

An important aspect of TextRank is that it does not require deep linguistic knowledge, nor domain or language specific annotated corpora, which makes it highly portable to other domains, genres, or languages.

References

S. Brin and L. Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7).

DUC. 2002. Document understanding conference 2002. http://www-nlpir.nist.gov/projects/duc/.

P.J. Herings, G. van der Laan, and D. Talman. 2001. Measuring the power of nodes in digraphs. Technical report, Tinbergen Institute.

J.M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.

C.Y. Lin and E.H. Hovy. 2003a. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of Human Language Technology Conference (HLT-NAACL 2003)*, Edmonton, Canada, May.

C.Y. Lin and E.H. Hovy. 2003b. The potential and limitations of sentence extraction for summarization. In *Proceedings of the HLT/NAACL Workshop on Automatic Summarization*, Edmonton, Canada, May.

R. Mihalcea and P. Tarau. 2004. TextRank – bringing order into texts.

R. Mihalcea, P. Tarau, and E. Figa. 2004. PageRank on semantic networks, with application to word sense disambiguation. In *Proceedings of the 20st International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland, August.

G. Salton, A. Singhal, M. Mitra, and C. Buckley. 1997. Automatic text structuring and summarization. *Information Processing and Management*, 2(32).

S. Teufel and M. Moens. 1997. Sentence extraction as a classification task. In *ACL/EACL workshop on "Intelligent and scalable Text summarization"*, pages 58–65, Madrid, Spain.

[5] Notice that rows two and four in Table 1 are in fact redundant, since the "hub" ("weakness") variations of the HITS (Positional) algorithms can be derived from their "authority" ("power") counterparts by reversing the edge orientation in the graphs.

[6] Only seven edges are incident with vertex 15, less than e.g. eleven edges incident with vertex 14 – not selected as "important" by TextRank.