TextRank: Bringing Order into Texts

Abstract

We introduce TextRank - a graph-based ranking model for text processing, and two unsupervised methods for keyword and sentence extraction.

1. Introduction

2. The TextRank Model

The scores associated with a vertex is determined based on the votes that are cast for it, and the score of the vertices casting these votes.

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 vertices $V \times V$. The score of a vertex V_i is defined as follows:

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

where $d \in (0, 1)$ (0.85) is a damping factor.

Starting from arbitrary values assigned to each node, the computation iterates until convergence below a given threshold is achieved. Convergence is achieved when the error rate for any vertex falls below a given threshold. This error rate is approximated with the difference between the scores computed at two successive iterations: $S^{k+1}(V_i) - S^k(V_i)$.

2.1 Undirected Graphs

In which case the out-degree of the vertex is equal to the in-degree of the vertex, a graph-based ranking algorithm can be also applied.

2.2 Weighted Graphs

We introduce a formula for graph-based ranking:

$$WS(V_i) = (1 - d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j)$$

2.3 Text as a Graph

The application of graph-based ranking algorithms consists of the following steps:

1. Identify text units and add them as vertices in the graph.
2. Identify relations that connect such text units to draw edges. Edges can be directed or undirected, weighted or unweighted.
3. Iterate the graph-base ranking algorithm until convergence.
4. Sort vertices based on their final score. Use these values for ranking/selection decisions.

3. Keyword Extraction

3.1 TextRank for Keyword Extraction

First, the text is tokenized, and annotated with part of speech tags. We consider only single words as candidates for addition to the graph, with multi-word keywords being eventually reconstructed in the post-processing phase.

Next, an edge is added between those lexical units that co-occur within a window of N words. After the graph is constructed (undirected, unweighted graph), the score associated with each vertex is set to 1, and the ranking algorithm in section 2 is run.

The top T vertices are retained for post-processing. These lexical units are marked in the text, and sequences of adjacent keywords are collapsed into a multi-word keyword.

3.2 Evaluation

I focus on section 4, so I skip over this subsection.

4. Sentence Extraction

4.1 TextRank for Sentence Extraction

A vertex is added to the graph for each sentence in the text. 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 text is therefore represented as a weighted graph, and consequently the ranking algorithm is run on the graph.

4.2 Evaluation

We evaluate TextRank sentence extraction algorithm on summarization task using 567 news articles in DUC 2002. We compare the performance of TextRank with the top five performing systems between fifteen systems, as well as with the baseline proposed by the DUC evaluators.

System	ROUGE score – Ngram(1,1)		
	basic (a)	stemmed (b)	stemmed no-stopwords (c)
S27	0.4814	0.5011	0.4405
S31	0.4715	0.4914	0.4160
TextRank	**0.4708**	**0.4904**	**0.4229**
S28	0.4703	0.4890	0.4346
S21	0.4683	0.4869	0.4222
Baseline	*0.4599*	*0.4779*	*0.4162*
S29	0.4502	0.4681	0.4019

Discussion The TextRank is fully unsupervised and derive an extractive summary. Notice that TextRank goes beyond the sentence "connectivity" in a text. A sentence is not identified as important based on the number of connections it has with other vertices in the graph, but it is identified as important by TextRank.

5. Why TextRank Works

TextRank does not only rely on the local context of a text unit, but rather it takes into account information recursively drawn from the entire text.

In the process of identifying important sentences, a sentence recommends another sentence that addresses similar concepts as being useful for the overall understanding of the text. The sentences that are highly recommended by other sentences in the text are likely to be more information for the given text, and will be therefore given a higher score.

TextRank implements what we refer to as "text surfing", which relates to the concept of text cohesion: from a certain concept C in a text, we are likely to follow links to connected concepts - that is, concepts that have a relation with the current concept C (be that a lexical or semantic relation).

6. Conclusions

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.