Graph Based Keyphrase Extraction

Debapriyo Majumdar
Indian Statistical Institute Kolkata
debapriyo@isical.ac.in

Introduction

A long article

Option 1: Title

The title describes the main goal of the article

Option 2: A summary or abstract

The summary in a *few sentences*

Can you tell me briefly what is in the article?

Option 3: The main topics discussed in the article

tags, keywords, keyphrases

A collection of descriptive phrases that represent the main concepts

Keyphrases — Examples

Query Suggestions in the Absence of Query Logs

Sumit Bhatia
Computer Science and Engineering
Pennsylvania State University
University Park, PA-16802, USA
sumit@cse.psu.edu

Debapriyo Majumdar
IBM Research India
Bengaluru-560045, India
debapriyo@in.ibm.com

Prasenjit Mitra
Information Science and Technology
Pennsylvania State University
University Park, PA-16802, USA
pmitra@ist.psu.edu

ABSTRACT

After an end-user has partially input a query, intelligent search engines can suggest possible completions of the partial query to help end-users quickly express their information needs. All major web-search engines and most proposed methods that suggest queries rely on search engine query logs to determine possible query suggestions. However, for customized search systems in the enterprise domain, intranet search, or personalized search such as email or desktop search or for infrequent queries, query logs are either not available or the user base and the number of past user queries is too small to learn appropriate models. We propose a probabilistic mechanism for generating query suggestions from the corpus without using query logs. We utilize the document corpus to extract a set of candidate phrases. As soon as a user starts typing a query, phrases that are highly correlated with the partial user query are selected as completions of the partial query and are offered as query suggestions. Our proposed approach is tested on a variety of datasets and is compared with state-of-the-art approaches. The experimental results clearly demonstrate the effectiveness of our approach in suggesting queries with higher quality.

Categories and Subject Descriptors

H.3.3 **[Information Storage and Retrieval]**: Information Search and Retrieval – Query formulation

General Terms

Algorithms, experimentation

Keywords

Query suggestion, query formulation, query completion, query log analysis, enterprise search.

1. INTRODUCTION

As users start typing a query in search engines' query boxes, most search engines assist users by providing a list of queries that have been proven to be effective in the past [26]. The user can quickly choose one of the suggested completions (in some cases, alternatives) and thus, does not have to type the whole query herself. Feuer et al. [15] analyzed more than 1.5 million queries from search logs of a commercial search engine and found that query suggestions accounted for roughly 30% of all the queries indicating the important role played by query suggestions in modern information retrieval systems. Furthermore, Kelly et al. [20] observed that the use of offered query suggestions is more for difficult topics, i.e., topics about which users have little knowledge to formulate good queries.

Most existing works on query suggestion utilize query logs to suggest queries [2, 4, 9, 10, 13, 18, 24, 25, 27]. Such query log based techniques are suitable for systems with a large user base. For example, web search engines have millions of users and if a user has some information need, almost always it turns out that some other users have searched for the same information before. The search engine can then utilize these large amounts of past usage data to offer possible query suggestions.

The scenario, however, is quite different for information retrieval systems that are not on the web or have a smaller user base, and thus, lack large amounts of query log data. For example, search engines built for customized applications in the enterprise domain are used by only a few hundreds or thousands of users. An effective query suggestion mechanism is required because the difference between users not finding some critical information and the users finding the information with ease may have a very significant business impact for the organization. The query logs of such systems are relatively much smaller, especially when a system is newly deployed for a small number of users, and even later on, the log seldom becomes as exhaustive as that of a web search engine. Even if a system can accumulate a relatively large query log over time, it would take a lot of time to be mature enough and most often that time is not affordable due to business needs. Furthermore, the data residing in these systems are also so customized that using a query log from another system is not a valid approach. However, the effectiveness of such enterprise search systems has significant business implications and even a small improvement can have a positive impact on the organization's business. Similarly, for personal data search systems, such as desktop search or personal email search, often there is only a single user resulting in very small query logs. Besides, query logs may not always be accessible in some applications due to privacy and legal constraints. Furthermore, even in the case of general-purpose web search engines, end-users sometimes pose queries that are not there in query logs or are not very frequent. *How to offer meaningful query suggestions to users in such scenarios* is thus, an interesting research problem, which we explore in this paper.

SIGIR'11, July 24–28, 2011, Beijing, China.

In research papers

- Actually phrases
- Manually assigned by the authors
- Helps in describing the main topics
- Easier to find by searching

KeyPhrases ≠ Word Cloud

Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types.

www.jasondavies.com/wordcloud/

linear constraints
linear diophantine equations
natural numbers
nonstrict inequations
strict inequations
upper bounds

TextRank algorithm

However, you can generate a much better cloud after extracting keyphrases

KeyPhrases > Word Cloud

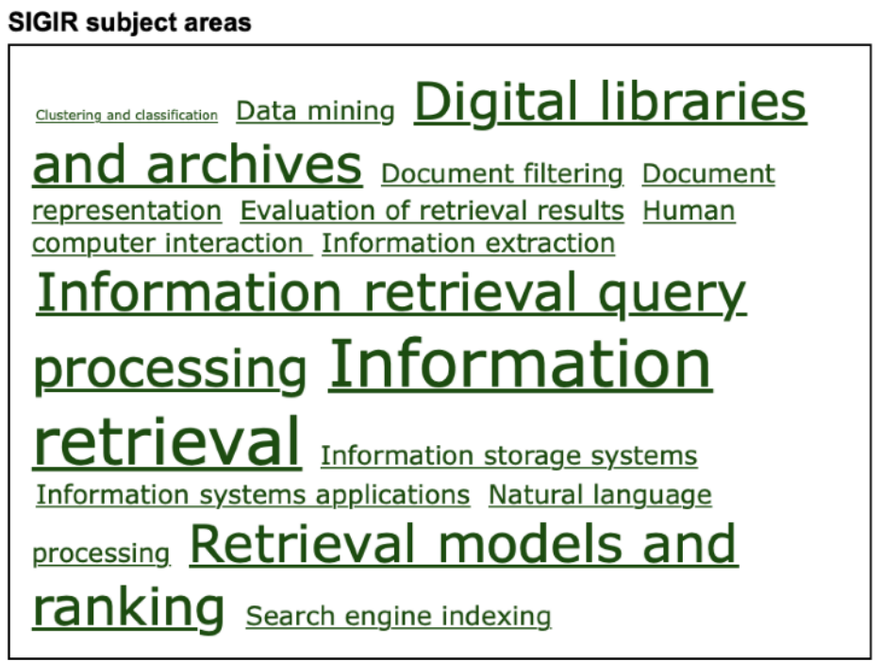
Applications of KeyPhrases

Visualize

1. Make a cloud (static)

2. Plot change of keyphrases (e.g., key issues in a services organization) over time

3. Insights, reporting



Add structure to data

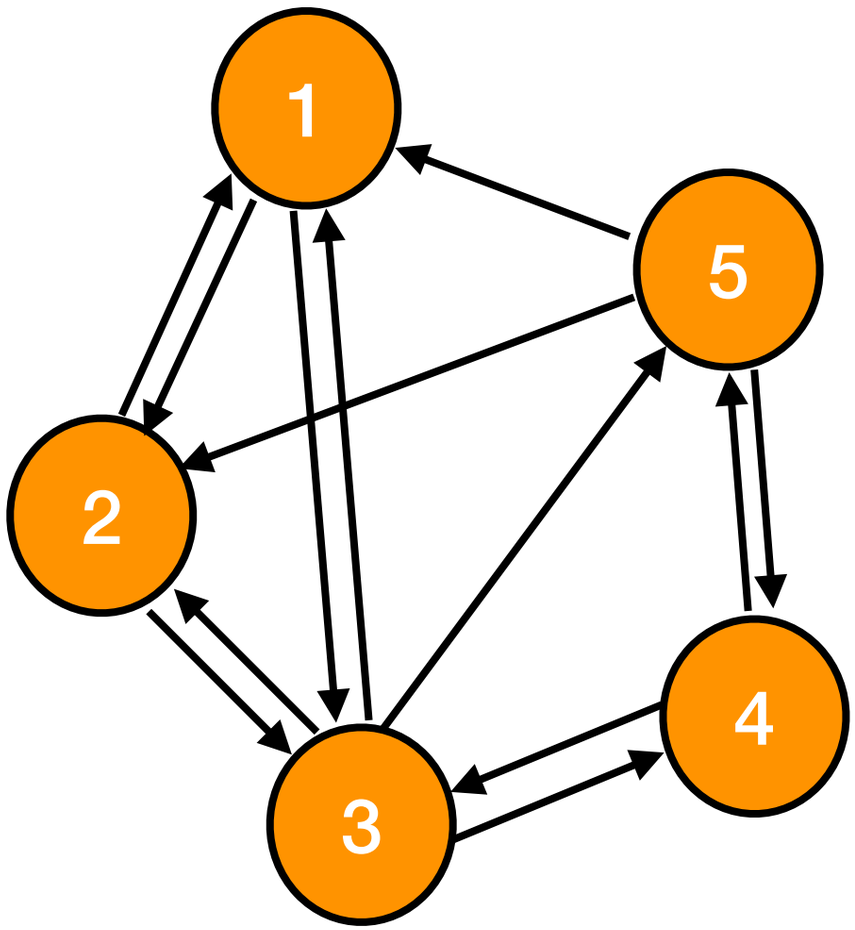
1. Automatic tagging: helps in browsing and filtering data — Faceted search

2. Tags, keywords: search engine optimization — better indexing

Others

Features for classification task
Sentiment Analysis
etc…

Revisiting PageRank



- The web as a graph, pages as nodes
- Links are directed edges
- Random surfer model:
  - A surfer starts at a random position
  - Follows a link randomly at each step
- Markov process: future state solely based on present state
- Can define a transition matrix
  - $M_{ij} = P[\text{surfer will next be at node } i \mid \text{now at } j]$

Random start

PageRank

$$M = \begin{bmatrix} 0 & 1/3 & 1/4 & 0 & 1/3 \\ 1/2 & 0 & 1/4 & 0 & 1/3 \\ 1/2 & 1/3 & 0 & 0 & 0 \\ 0 & 1/3 & 1/4 & 1/2 & 1/3 \\ 0 & 0 & 1/4 & 1/2 & 0 \end{bmatrix} \quad v = \begin{bmatrix} 1/5 \\ 1/5 \\ 1/5 \\ 1/5 \\ 1/5 \end{bmatrix} \longrightarrow Mv = \begin{bmatrix} 0.18 \\ 0.22 \\ 0.26 \\ 0.18 \\ 0.15 \end{bmatrix} \longrightarrow \cdots \longrightarrow M^k v = \begin{bmatrix} 0.19 \\ 0.21 \\ 0.26 \\ 0.19 \\ 0.16 \end{bmatrix}$$

Revisiting PageRank — 2

- Correction by damping factor d (usually $d = 0.85$)
  - For PageRank, a way to fight term spam
  - At each step, follow a link with probability d
  - But jump back to any random node with probability $1 - d$
    - Weight of all nodes the same here

$$v_k = (1 - d) \times \begin{bmatrix} 1/n \\ \vdots \\ 1/n \end{bmatrix} + d \times M v_{k-1}$$

Some nodes can be given higher priority by changing this vector here

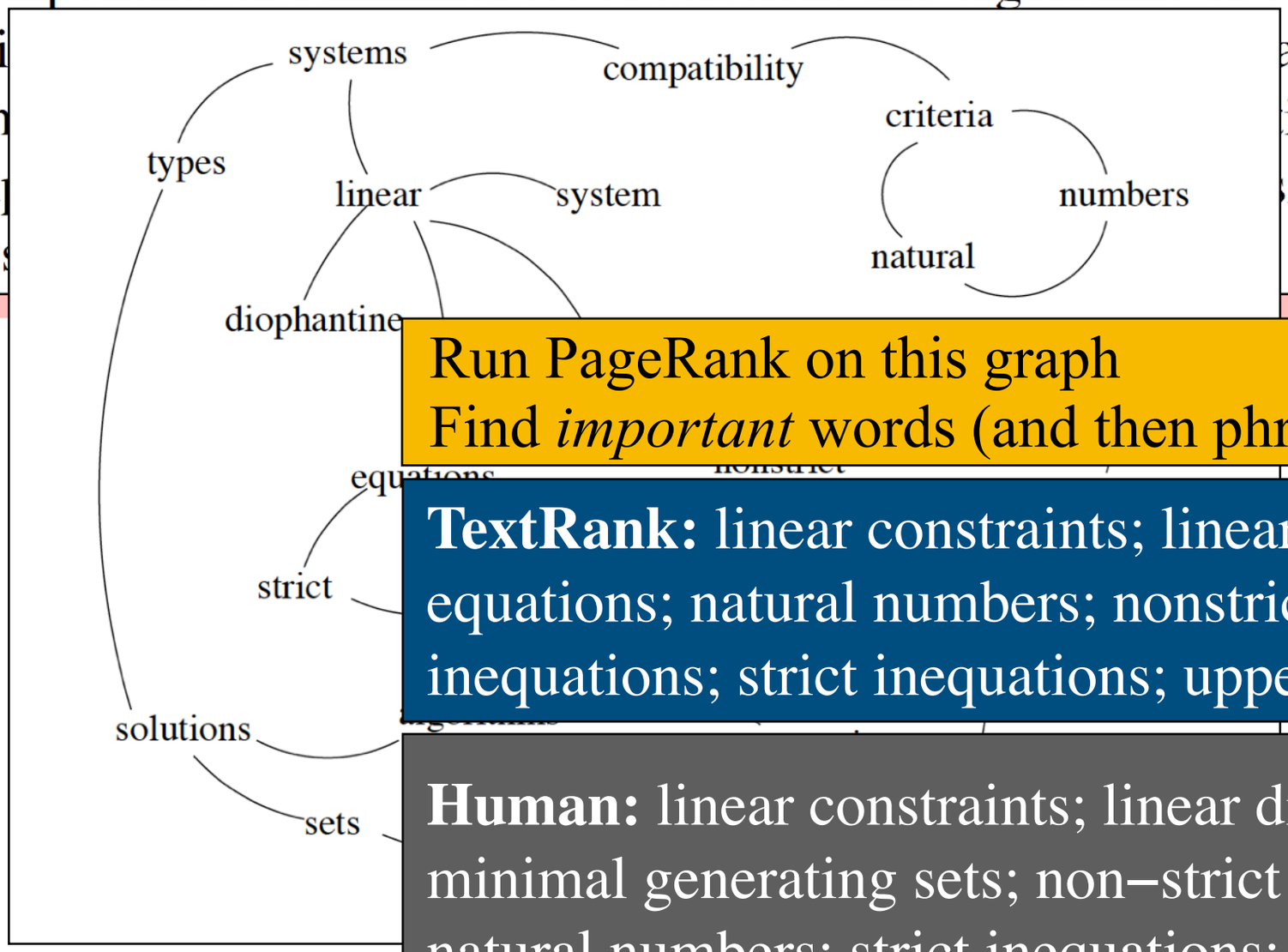
TextRank (Mihalcea and Tarau, EMNLP 2004)

Close to 3000 citations!

Text document

Compatibility of systems of linear constraints over the set of natural numbers.
Cr uations, strict
ine bounds for
components of a minimal set of solutions and algorithms of construction of
mi are given.
Th ing a minimal
su sidered types
sy

Find some relations between the words
Build a graph



Run PageRank on this graph
Find *important* words (and then phrases)

TextRank: linear constraints; linear diophantine equations; natural numbers; nonstrict inequations; strict inequations; upper bounds

Human: linear constraints; linear diophantine equations; minimal generating sets; non–strict inequations; set of natural numbers; strict inequations; upper bounds

TextRank: The Framework

- Identify the text units that best define the task
  - Add them as nodes of the graph
- Define a *relation* between the units
  - Use the relation to draw edges between the nodes
  - Edges can be directed / undirected, weighted / unweighted
- Iterate the graph-based ranking algorithm till convergence
  - Based on some convergence threshold, or maximum number of iterations
- Post process high scoring nodes to produce final ranked list of nodes

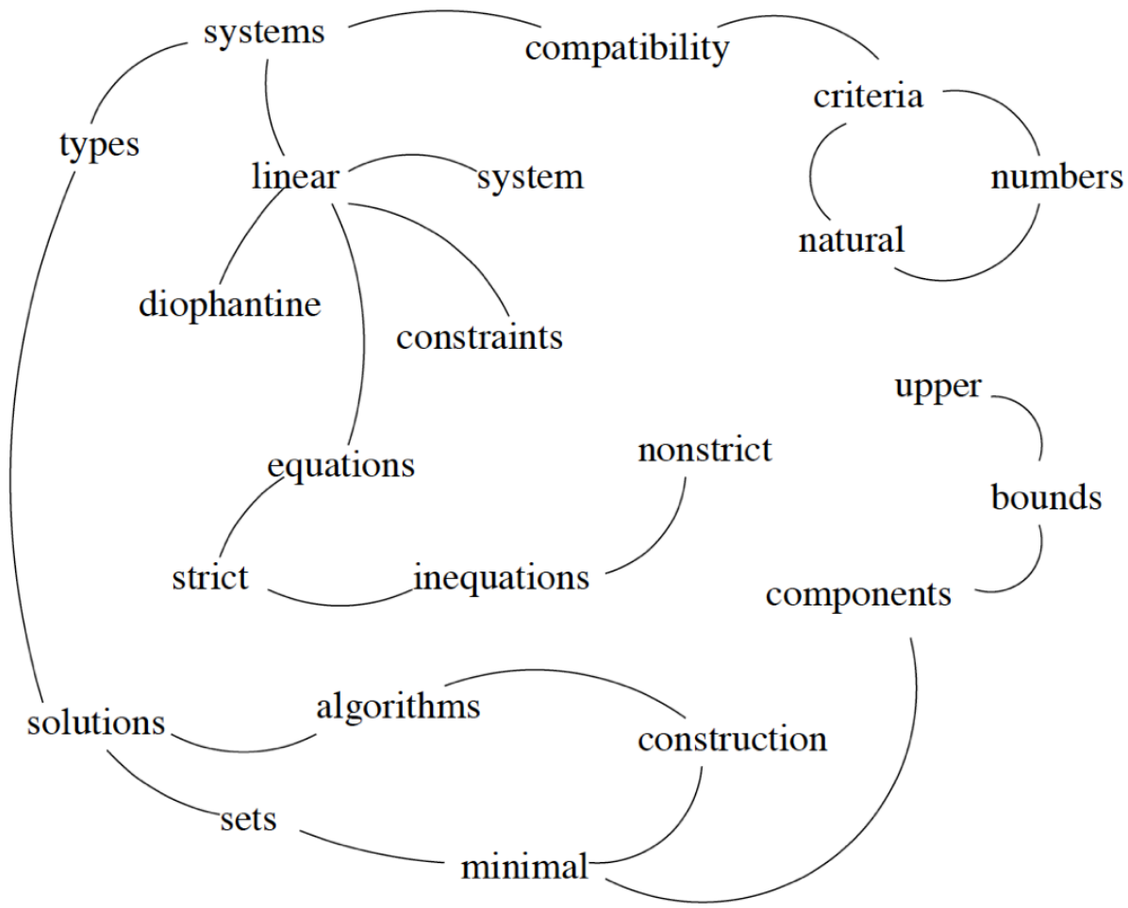
TextRank: selecting the nodes

- Goal: Find top keyphrases — hence phrases
  - But for now, let's start with words (more later)
- All words?
  - Possibile syntactic filters:
    - All words
    - Some combination of nouns, verbs and adjectives
    - Requires part of speech tagging upfront
  - Nouns and adjectives produce the best results (experimental)
    - **But the choice is open, depends on what we want**

TextRank: how are the words related?

Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types.

- Proposed approach: completely unsupervised
- Relationship: word co-occurrence within a pre-defined boundary (say, between 2 and 10 words)
- Intuition: recommendation (links)
  - An important words recommends that the next (or previous) one is also important



TextRank: Weighted / Unweighted?

- PageRank
  - Assumes unweighted graph
  - A page hardly includes multiple or partial links to another page

- TextRank
  - May include multiple or partial link between the units
    - The graphs are build from natural language text
  - Incorporate the "strength" of connectivity
    - Weight of the edge
    - If word w_1 is linked to w_2 three times, and to w_3 once, then weight of (w_1, w_2) would be three times that of (w_1, w_3)
    - Improves results!

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

edge weights

damping factor

TextRank: Extraction of KeyPhrases

The iterative graph algorithm produces scores (TextRank) for each word (node)

- But we wanted to find *phrases*, not single words only
- If consecutive words within a clause are get *high scores*, then join them

Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types.

Constructed **Keyphrases:**
linear constraints; linear diophantine equations; natural numbers; nonstrict inequations; strict inequations; upper bounds

Beyond TextRank (but not too far beyond)

- SingleRank and ExpandRank (Wan and Xiao, AAAI 2008)
- SingleRank (extraction from a single document)
  - Same as TextRank, except in the post-processing
  - Do not just join high-scoring words, rather add their scores to obtain a score for the combined phrase
  - For example: linear (0.3) + Diophantine (0.2) + equations (0.3) $\Longrightarrow$ “linear Diophantine equations” (0.8)
  - Rank the phrases using the combined score
  - An effort to add some intuition to joining words to make phrases
  - Problem: single words (but some of them are good enough to be keyphrases) lose the advantage severely

ExpandRank

- ExpandRank (Wan and Xiao, AAAI 2008)
  - Find *k* neighbor documents using some document similarity measure
  - Expand the word graph to a bigger one (using the neighbor documents)
  - Run graph iteration algorithm (PageRank) and get scores for words
- Post-process similar to SingleRank
- Note: outside phrases do not qualify (due to post-processing)

TopicRank (Bougouin, Boudin and Daille, IJCNLP 2013)

- Initially, extract keyphrases using SingleRank
- Similarity between keyphrases: take the stemmed form of phrases, and if two phrases have at least 25% words common, there is a link between them
- Hierarchical Agglomerative Clustering to group keyphrases
- A graph is formed between clusters, then PageRank