Final Project Proposal, Advanced Natural Language Processing, S'17
Title: Graph-based ranking algorithms for scientific keyphrase extraction
Author: Michael Spector

Background

Keyphrases - short, descriptive sequences of words - provide a compact summary of a document or set of documents, and can be useful for clustering, classifying, and indexing those documents. Keyphrases can also aid readers in efficiently understanding the gist of their contents. Unfortunately, identifying useful keyphrases often involves manual annotation by humans, a process which can be laborious and time-consuming. Automatically extracting these keyphrases from unstructured text without human annotations is a longstanding goal of natural language processing, with many applications to text analysis. This work will focus on one such application: the automatic extraction of scientific keyphrases from abstracts of scientific publications.

Keyphrase extraction typically takes place in two stages: (1) identifying candidate keyphrases, and (2) ranking the candidate keyphrases. Identifying candidates can be a simple process; for example, extracting all tokens of a certain part-of-speech, or all token sequences that match a certain sequence of part-of-speech tags. It is also possible to use external resources or domain knowledge to define heuristics that identify likely candidates; for example, extracting all phrases that appear as an entry or link in Wikipedia.

Ranking candidate keyphrases is the more difficult of the two stages. This is partly because evaluating keyphrases is inherently subjective: although there do exist cases in which one keyphrase seems clearly “better” than another, there are also many cases in which this is not the case. Another factor that contributes to the difficulty of ranking candidate keyphrases is the relative scarcity of gold-standard datasets. Although they do exist, they require expensive human annotation, which limits the amount of training data available (particularly when considering domain-specific keyphrase extraction).

Algorithms for ranking candidate keyphrases can be broadly divided into supervised and unsupervised methods. Using a training set of documents and gold-standard keyphrases for each document, supervised methods typically learn a function that compares two candidate keyphrases and computes their relative suitability as a keyphrase. By evaluating this function on pairs of all candidate keyphrases, a global ranking can be established.

Unsupervised methods for ranking candidate do not make use of training data; instead, they attempt to uncover latent structure in the documents that can then be used to rank candidate keyphrases by relative keyphrase quality. The most popular unsupervised keyphrase ranking algorithms are graph-based: in these methods, a graph is constructed that treats each phrase as a node, and draws edges between nodes based on certain similarity criteria of the phrases. A graph-based similarity algorithm, such as Google's PageRank[1], then computes the most relevant nodes based on the connectedness between nodes. Other graph-based variations on PageRank exist, such as TextRank[2] and DivRank[3]. The measures of similarity between nodes can be established either by document statistics, or by external resources and domain knowledge.

Project Scope

Although it will be necessary (as part of the first stage of the overall pipeline) to establish a reasonable way to generate keyphrase candidates, this project will focus on different methods for ranking candidate keyphrases; specifically, graph-based unsupervised approaches. Because this project will mostly involve experiments in an unsupervised setting, human annotation will not be required. Any supervised learning algorithms that I do end up using will be trained on the SemEval-2010 dataset (referenced below).

The preliminary experiment will consist of the following subcomponents:

1. Establish and implement a reasonable way to generate multi-word keyphrase candidates. This will likely be based on matching sequences of words based a regular expression over part-of-speech tags.
2. Implement a simple baseline approach to determining candidate keyphrase relatedness using co-occurrence frequency statistics
3. Implement a graph-based algorithm (likely TextRank or some variant) and produce keywords

The full project will extend the preliminary experiment in the following ways:

4. Experiment with different ways of determining candidate keyphrase relatedness, in the following broad categories:
    a. A simple baseline approach using co-occurrence frequency statistics
    b. An external resource-based approach, using a measure of relatedness based on Wikipedia links
    c. A word-embedding based approach
5. Implement other graph-based algorithms, and compare performance to the initial algorithm
6. (If time permits) experiment with simple supervised approaches and compare against unsupervised algorithms

Resources

This project will require a large quantity of scientific paper abstracts. Fortunately, as part of my final project for last semester's "Introduction to Natural Language Processing" course, I have established a dataset consisting of titles, authors, abstracts, and categories on 88,618 academic papers from the arXiv preprint server, submitted between 2012 and 2016, and restricted to only papers in the field of computer science.

If time permits, I plan to also compare supervised methods against my unsupervised graph-based ranking algorithms. To do this, I will use the keyphrase extraction dataset from SemEval-2010 task 5[4], which includes gold-standard keyphrases of 284 scientific articles and their abstracts.

Part of my proposed project involves using Wikipedia as an external source of domain knowledge to determine candidate keyphrase relatedness. To gather this information, I will use the publicly available `wikipedia` python package, available on the Python Package Index (PyPI).

Finally, I will use the publicly available Natural Language Toolkit (`nltk`) python package for low-level natural language processing and pre-processing requirements.

References

1. Page, Lawrence, et al. The PageRank citation ranking: Bringing order to the web. Stanford InfoLab, 1999.
2. Mihalcea, Rada, and Paul Tarau. "TextRank: Bringing order into texts." Association for Computational Linguistics, 2004.
3. Mei, Qiaozhu, Jian Guo, and Dragomir Radev. "Divrank: the interplay of prestige and diversity in information networks." Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. Acm, 2010.
4. Kim, Su Nam, et al. "Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles." Proceedings of the 5th International Workshop on Semantic Evaluation. Association for Computational Linguistics, 2010.