Review of YAKE! for Keyword Extraction

Jefferson Zhan
Grainger College of Engineering, University of Illinois at Urbana-Champaign
CS 410: Text Information Systems
Dr. ChengXiang Zhai
zhan35@illinois.edu

Introduction

Keyword extraction is a text analysis technique that extracts the most relevant tokens, words, phrases, or sentences from a given input of a text in an automated process. One common application of keyword extraction is to batch analyze thousands of reviews and/or tweets about a specific product and detect what words or phrases best describe the data. With the rise of machine learning and artificial intelligence, particularly in the realm of natural language processing, different algorithms have risen in popularity. There are a few common statistical approaches, namely term frequency – inverse document frequency (TF-IDF) and Rapid Automatic Keyword Extraction (RAKE). They tend to take advantage of word frequency, n-gram statistics or co-occurrences, and word collocations. Extracting keywords using existing statistical models has grown to be inefficient as information grows exponentially in complexity and size. Language diversity and defining what is relevant is chief amongst the challenges. Yet Another Keyword Extractor or simply Yake! is an attempt to create a multi-lingual keyword extraction algorithm to support texts in various sizes, languages, and domain.

Yake! Technical Review

Amongst the most common algorithms for keyword extraction, supervised approaches seem to dominate. Yake! is an unsupervised alternative to such methods. Due to its unsupervised approach, it can easily be "plugged and played" into various systems, is domain and language independent, and term frequency free. The five main steps of Yake! are as follows: (1) text pre-processing and candidate term identification; (2) feature extraction; (3) computing term score; (4) n-gram generation and computing candidate keyword score; and (5) data deduplication and ranking. There are a few unique, interesting decisions the algorithm makes. Within feature extraction and computing how "good" a word is, 5 features are utilized: casing, position, frequency, relation to context and word different sentence. Due to the heuristic of uppercase words tending to be more relevant, uppercased terms, particular acronyms, causes the term frequency score (TF) to be weighted more. A max is taken between the number of occurrences of the candidate term t starting with the uppercase letter and the number of times the candidate term t is marked as an acronym.

$$Tcase = \frac{\max\left(TF(U(t)), TF(A(t))\right)}{\ln\left(TF(t)\right)}$$

In addition, the heuristic of relevant keywords tending to appear at the beginning of a document causes the term position to be weighted with the following:

$$Tposition = \ln(\ln(3 + median(Sen))$$

The double log here is used smooth the difference between terms that occur with a large median difference. For term frequency normalization, YAKE! attempts to find candidate terms whose frequencies are above the mean with a certain degree of dispersion.

$$TFnorm = \frac{TF(t)}{MeanTF + 1 * \sigma}$$

Word relatedness is calculated by counting how many different terms occur to the left or right of a candidate word noting that words that occur frequency are more likely to be stop words. Lastly, there is a count of the number of different sentences a candidate word occurs in. These 5 metrics are then combined to form a word score by using a 3-gram model. Even with these equations for candidate scoring, the algorithm remains what they call "term frequency free". There are no conditions set with respect to the minimum frequency that a candidate keyword must have. To compute the candidate keyword score, YAKE! relies on a sliding window to generate a sequence of terms from 1-gram to n-grams. Keywords are then produced with each candidate being assigned a score.

YAKE! Results

In general, YAKE! tends to outperform other unsupervised keyword extraction algorithms especially with larger texts and texts with different domains and type of documents. Crucially, YAKE! takes advantage of the relative position of words. This concept of positional information causes YAKE! to outperform most statistical-based methods, graph-based methods, and topic-based methods. For instance, though it captures co-occurrence relationships, TextRank, a graph based method based on PageRank, cannot capture all the heuristics that YAKE! takes advantage of. With that said, there is still room for improvement. Notably, in the case of long texts will multiple topics, YAKE!'s relative position advantage is actually a potential detriment. Key phrases that appear relatively far away from the beginning of the document and previous location-based features would cause the candidate score to be fairly low.

Conclusion

YAKE! is a light weight unsupervised approach to keyword extraction that does not rely on external corpora or any common linguistic tools. Though it does not take advantage of Named Entity Recognition or Part of Speech tagging, YAKE! does take advantage of many heuristics in relative position of candidate words while remaining term frequency independent. A keyword may be considered significant or insignificant with either one or multiple occurrences. This approach makes YAKE! easy to adopt in various languages, differing domains, and contrasting topics. As a recent entry into keyword extraction methods, YAKE! provides complex feature engineering to separate itself from not only other statistical based methods but also graph based and topic based approaches. As one of the few multilingual algorithms, YAKE!, despite its flaws, lends itself to be easily scaled to vast collections