Keyword Extraction

Keyword Extractor Tools

Online tool

Ref_1: https://www.cortical.io/freetools/extract-keywords/

Ref_2: https://linguakit.com/en/keyword-extractor

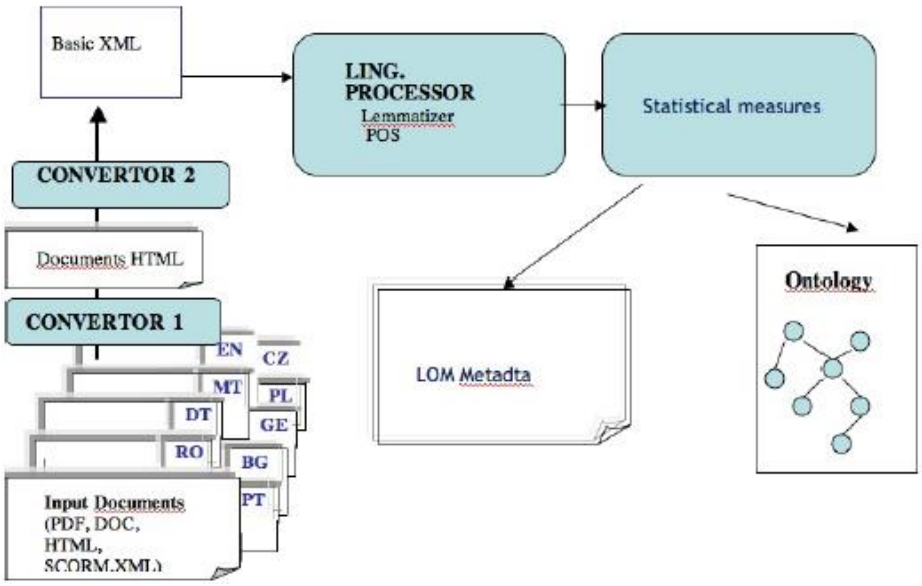
1. Khái niệm

Ref: Keyword extractor of Learning Object

The Keyword Extractor, p.2

The task of a *keyword extractor* is to automatically identify a set of terms in a document that best describes *its content*. Keyword extractors have been employed to identify appropriate entries for building an automatic index for a document collection and have been used to classify text

Keyword Extractor – Trình rút trích từ khoá



Hình 1 : Kiến trúc trình rút trích từ khoá

parts:

• Lexical units: they represent the combination of a lemma and a part of speech tag. They are the basic units on which statistics are calculated and they are returned as possible keywords. Only those lexical units that can occur as possible keywords are retained – mainly nouns, proper nouns and unknown words.

• Word Form Types: they represent the actual form of the lexical unit in the input file in combination with their morphological information.

• Documents: they represent the documents which constitute the corpus including their names and domains. The two domains of our documents are information technologies for the non-expert and eLearning. Potentially interesting sequences

Potentially interesting sequences of words are extracted using the suffix array data structure [15] but a condition is that they must appear at least twice in the document. Afterwards, filtering occurs on the basis of language specific information and sequences longer than a certain threshold are discarded. In general, sequences comprising up to 3 words are retained.

The list of candidate keywords is ranked by their saliency and to determine it an approach based on frequency has been adopted.

As already mentioned, keywords are those terms that best identify the text and represent what the text is about (i.e. the topics of a text). They tend to occur more often in that text than could be expected if all words were distributed randomly over a corpus.

A well-established way to measure the distribution of terms over a collection of documents is TFIDF, cf.

equation 1.

$$\text{TFIDF với } IDF = \log_2 \frac{N}{df}$$

2. Các phương pháp trích xuất từ khoá

Ref: Keyword extraction and Mmethods

1) Simple Statistics Approaches
2) Linguistics Approaches
3) Machine Learning Approaches

Simple Statistics Approaches

Comprises simple methods which do not require the training data. In addition, methods are language and domain-independent. The statistics of the words from document can be used to identify keywords: n-gram statistics, word frequency, TF-IDF, word co-occurrences, PAT Tree (Patricia Tree; a suffix tree or position tree), etc. The disadvantage is that in some professional texts, such as health and medical, the most important keyword may appear only once in the article. The use of statistically empowered models may inadvertently filter out these words [19].

Linguistics Approaches

Use the linguistics feature of the words mainly, sentences and document. Lexical, syntactic, semantic and discourse analysis are some of the most common but complex analysis.

Machine Learning Approaches

Considers supervised or unsupervised learning from the examples, but related work on keyword extraction prefers supervised approach. Supervised machine learning approaches induce a model which is trained on a set of keywords. They require a manual annotation in the learning dataset which is extremely tedious and inconsistent (sometimes requests predefined taxonomy). Unfortunately, authors usually assign keywords to their documents only when they are compelled to do it. Thus induced model is applied for keyword extraction from a new document. This approach includes Naïve Bayes, SVM, C4.5, Bagging, etc. Thus methods require training data, and are often dependent on the domain. System needs to re-learn and establish the model every time when domain was changed [20, 21]. Model induction can be very demanding and time consuming on massive datasets.

3. Tìm hiểu thuật toán tf – idf

Ref: Giới thiệu trích xuất thông tin – tf-idf

Term frequency tf – tần số từ

Term frequency tf

- The term frequency $tf_{t,d}$ of term *t* in document *d* is defined as the number of times that *t* occurs in *d*.
- We want to use tf when computing query-document match scores. But how?
- Raw term frequency is not what we want:
  - A document with 10 occurrences of the term is more relevant than a document with 1 occurrence of the term.
  - But not 10 times more relevant.
- Relevance does not increase proportionally with term frequency.

NB: frequency = count in IR

idf (inverse document frequency)

idf weight

- df_t is the document frequency of *t*: the number of documents that contain *t*
  - df_t is an inverse measure of the informativeness of *t*
  - $df_t \leq N$
- We define the idf (inverse document frequency) of *t* by

$$idf_t = \log_{10}(N/df_t)$$

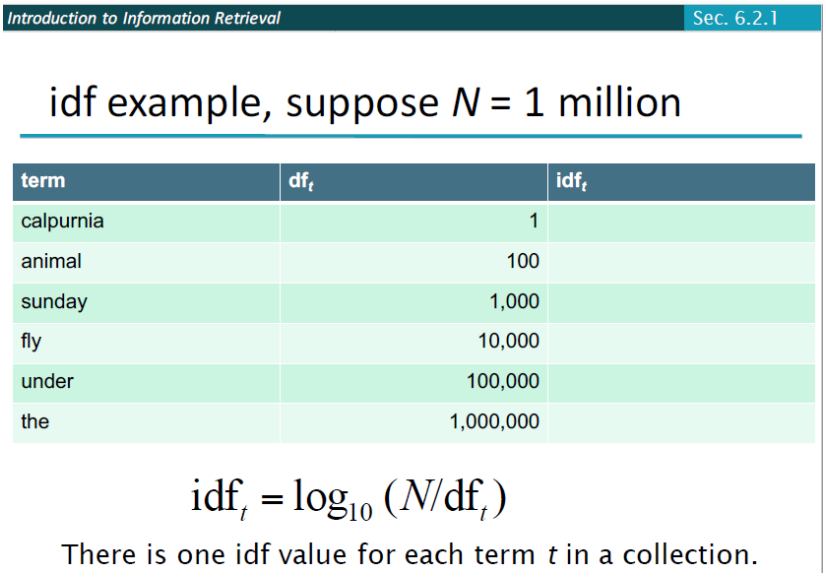
  - We use log (N/df_t) instead of N/df_t to "dampen" the effect of idf.

Will turn out the base of the log is immaterial.

.turn out: prove to be the case

.immaterial : unimportant, irrelevant

idf Example



Term Extraction Approaches

Ref: Term Extraction Approaches

Term Extraction Algorithm

- TF-IDF
- RIDF
- Rake Algorithm

Phase 1 : Pre-Processing – tiền xử lý

Tiền xử lý tài liệu (document)

stopList pre-processing

Extremely common words which would appear to be of little value in helping select documents matching a user need are excluded from the vocabulary entirely. These words form the Stop List.

• Use Jate's built in "StopList" for filtering.

Ref: Stop word list 1 , prouns (các đại từ), động từ to 'be' và chia thì của nó (is, are, was, has, …), ..

Ví dụ các stopword trong tiếng Anh: a, an, the, is, be, been, can, cannot, do, does, done, has, have, 'I', me, my,man, he, her, his, him, it, its, itself, one, our, their, they, them, these, this, us, we, you, your, yours , v.v…

Ref : Stopword tiếng Việt: Danh sách stop word tiếng Việt

Cái, các, cần, có, của, đã, đang, để, đều, do, đó, được, gì, không, là, lúc, này, nếu, nhiều, phải, qua, rất, sẽ, sự, thì, và, vì, việc, vừa, v.v…

Lemmatization

Group together words that are present in the document as different inflected forms to a single word so they can be analyzed as a single item.

Example : "*run, runs, ran and running are forms of the same lexeme, with run as the lemma.*"

Phase 2 : Candidate Term Extraction

Candidate Term Extraction

• **Approaches to Candidate Term Extraction :**

1. Simply extracting single words as candidate terms. If you task extracts single words as terms.

(Naïve Approach)

2. A generic N-gram extractor that extracts '*n grams*'.

Final Approach : Stanford's OpenNLP NPE (Noun Phrase Extractor) that extracts noun phrases as candidate terms.

Ref: http://corenlp.run/ , Stanford CoreNLP 3.6.1 (update 2018/04/05)

Phase 3 : Index Building

Building Document Index

• Using Jate toolkit to build a corpus index (Pre-Requisite for Term Extraction).

• Memory Based / Disk Resident file / Exporting to HSQL (HyperSQL).