Web Services and Web Data

XJCO3011

Session 10. Parsing and Tokenization

Document Parsing

- Document parsing involves the recognition of the content and structure of text documents.
- Extracting words from the sequence of characters in a document is called *tokenizing* or *lexical analysis*
- In addition to natural language words, there can be many other types of content in a document, such as metadata, images, graphics, code, and tables. Among them, metadata is information about a document that is not part of the text content
- Metadata is information about a document that is not part of the text, and includes:
  1. Document attributes such as date, author, and most importantly, the tags.
  2. The tags are used by *markup languages* to identify document components.
  3. The most popular *markup languages* are HTML (Hypertext Markup Language) and XML (Extensible Markup Language).
- After tokenization, a parser uses the tags and other metadata recognized in the document to interpret the document's structure based on the syntax of the markup language (syntactic analysis) and to produce a representation of the document that includes both the structure and content.
- For example, an HTML parser can interpret the structure of a web page, and creates a Document Object Model (DOM) representation of the page that is used by a web browser.

Tokenizing

- Tokenizing is the process of forming words from the sequence of characters in a document.
- In many early systems, a “word” was defined as any sequence of alphanumeric characters of length 3 or more, terminated by a space or other special character. All uppercase letters were also converted to lowercase, for example, the text:

  Bigcorp’s 2007 bi-annual report showed profits rose 10%.

  would produce the following tokens:

  bigcorp 2007 annual report showed profits rose.
- However, this simple tokenizing process is not adequate for most search applications because too much information is discarded

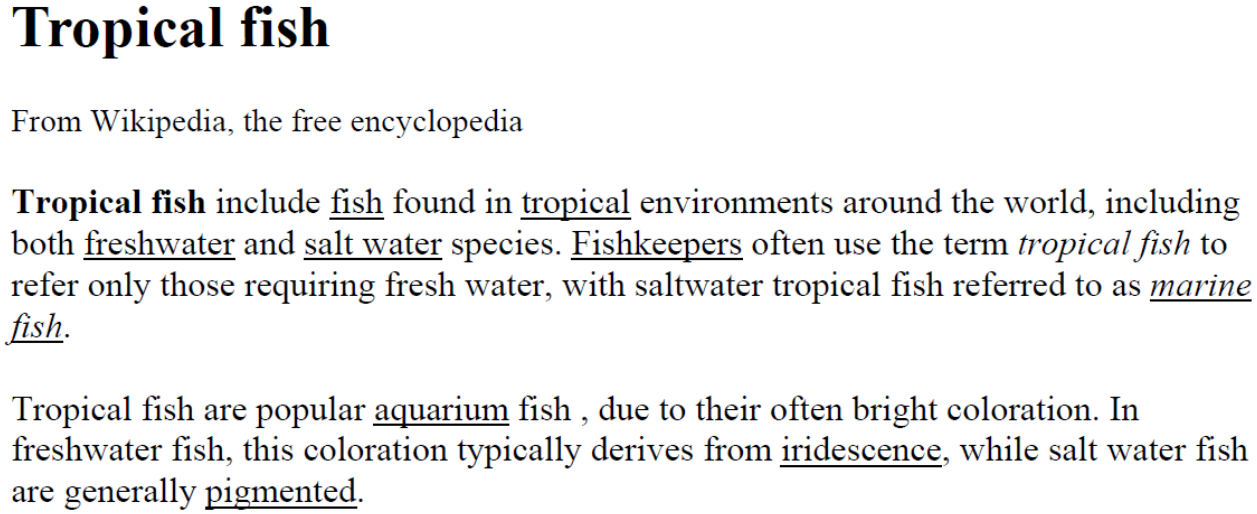
Tokenization is more complicated than it seems

- Small words (one or two characters) can be important in some queries, usually in combinations with other words. For example, xp, pm, ben e king, el paso, master p, j lo, world war II.
- Both hyphenated and non-hyphenated forms of many words are common. In some cases the hyphen is not needed. For example, e-bay, wal-mart, active-x, cd-rom, t-shirts. At other times, hyphens should be considered either as part of the word or a word separator. For example, winston-salem, mazda rx-7, e-cards, pre-diabetes, t-mobile, spanish-speaking.
- Special characters (e.g. ! or &) are an important part of the tags, URLs, code, and other important parts of documents that must be correctly tokenized.
- Capitalized words can have different meaning from lowercase words. For example, “Bush” and “Apple”.
- Apostrophes can be a part of a word, a part of a possessive, or just a mistake. For example, rosie o’donnell (a famous American actress, comedian, talk show host, and social activist name), can’t, don’t, 80’s, 1890’s, men’s straw hats, master’s degree, England's ten largest cities.
- Numbers can be important, including decimals. For example, Nokia 3250, top 10 courses, united 93, QuickTime 6.5 pro, 92.3 the beat, 288358 (yes, this was a real query; it’s a patent number).
- Periods can occur in numbers, abbreviations (e.g., “I.B.M.”, “Ph.D.”), URLs, ends of sentences, and other situations.

Document Structure and Markup

- In database applications, the fields or attributes of database records are a critical part of searching. Queries are specified in terms of the required values of these fields. In the case of web search, queries usually do not refer to document structure or fields. Some parts of the structure of web pages, indicated by HTML markup, are very significant features used by the ranking algorithm. The document parser must recognize this structure and make it available for indexing.

- The main heading for the page, e.g. “tropical fish”, indicates that this phrase is particularly important.
- If the same phrase is also in **bold** and *italics* in the body of the text, then this is further evidence of its importance.
- Other words and phrases are used as the anchor text for links and are likely to be good terms to represent the content of the page.
  <a> href=”http://www.somewhere.com”>the somewhere page.



Tropical fish

From Wikipedia, the free encyclopedia

Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species. Fishkeepers often use the term *tropical fish* to refer only those requiring fresh water, with saltwater tropical fish referred to as *marine fish*.

Tropical fish are popular aquarium fish , due to their often bright coloration. In freshwater fish, this coloration typically derives from iridescence, while salt water fish are generally pigmented.

Document Structure and Markup

National Flag:

Area: 83,858 square kilometres

Population: 8,205,000

Iso: AT

Country: Austria

Capital: Vienna

Continent: EU

Tld: .at

Currency Code: EUR

Currency Name: Euro

Phone: 43

Postal Code Format: ####

Postal Code Regex: ^(\d{4})$

Languages: de-AT,hr,hu,sl

Neighbours: CH DE HU SK CZ IT SI LI

Edit

Rendered Page

```
pe="multipart/form-data" method="post"><table><tr id="places_national_flag__row"><td class="w2p_fl"><label class="readonly"
lag" id="places_national_flag__label">National Flag: </label></td><td class="w2p_fw"><img src="/places/static/images/flags/at.png" /></td><td
tr><tr id="places_area__row"><td class="w2p_fl"><label class="readonly" for="places_area" id="places_area__label">Area: </label></td><td
square kilometres</td><td class="w2p_fc"></td></tr><tr id="places_population__row"><td class="w2p_fl"><label class="readonly" for="places_population"
_label">Population: </label></td><td class="w2p_fw">8,205,000</td><td class="w2p_fc"></td></tr><tr id="places_iso__row"><td class="w2p_fl"><label
places_iso" id="places_iso__label">Iso: </label></td><td class="w2p_fw">AT</td><td class="w2p_fc"></td></tr><tr id="places_country__row"><td
class="readonly" for="places_country" id="places_country__label">Country: </label></td><td class="w2p_fw">Austria</td><td class="w2p_fc"></td></tr>
__row"><td class="w2p_fl"><label class="readonly" for="places_capital" id="places_capital__label">Capital: </label></td><td
/td><td class="w2p_fc"></td></tr><tr id="places_continent__row"><td class="w2p_fl"><label class="readonly" for="places_continent"
label">Continent: </label></td><td class="w2p_fw"><a href="/places/default/continent/EU">EU</a></td><td class="w2p_fc"></td></tr><tr
td class="w2p_fl"><label class="readonly" for="places_tld" id="places_tld__label">Tld: </label></td><td class="w2p_fw">.at</td><td class="w2p_fc">
es_currency_code__row"><td class="w2p_fl"><label class="readonly" for="places_currency_code" id="places_currency_code__label">Currency Code: </label>
">EUR</td><td class="w2p_fc"></td></tr><tr id="places_currency_name__row"><td class="w2p_fl"><label class="readonly" for="places_currency_name"
id="places_currency_name__label">Currency Name: </label></td><td class="w2p_fw">Euro</td><td class="w2p_fc"></td></tr><tr id="places_phone__row"><td class="w2p_fl"><label
class="readonly" for="places_phone" id="places_phone__label">Phone: </label></td><td class="w2p_fw">43</td><td class="w2p_fc"></td></tr><tr
id="places_postal_code_format__row"><td class="w2p_fl"><label class="readonly" for="places_postal_code_format" id="places_postal_code_format__label">Postal Code Format:
</label></td><td class="w2p_fw">####</td><td class="w2p_fc"></td></tr><tr id="places_postal_code_regex__row"><td class="w2p_fl"><label class="readonly"
for="places_postal_code_regex" id="places_postal_code_regex__label">Postal Code Regex: </label></td><td class="w2p_fw">^(\d{4})$</td><td class="w2p_fc"></td></tr><tr
id="places_languages__row"><td class="w2p_fl"><label class="readonly" for="places_languages" id="places_languages__label">Languages: </label></td><td class="w2p_fw">de-
AT,hr,hu,sl</td><td class="w2p_fc"></td></tr><tr id="places_neighbours__row"><td class="w2p_fl"><label class="readonly" for="places_neighbours"
id="places_neighbours__label">Neighbours: </label></td><td class="w2p_fw"><div><a href="/places/default/iso/CH">CH </a><a href="/places/default/iso/DE">DE </a><a
href="/places/default/iso/HU">HU </a><a href="/places/default/iso/SK">SK </a><a href="/places/default/iso/CZ">CZ </a><a href="/places/default/iso/IT">IT </a><a
href="/places/default/iso/SI">SI </a><a href="/places/default/iso/LI">LI </a></div></td><td class="w2p_fc"></td></tr></table><div style="display:none;"><input name="id"
type="hidden" value="1602483" /></div></form>

<a href="/places/default/edit/Austria-15">Edit</a>
```

Part of the Page's HTML Source

Two-pass Tokenization

- The tokenizing process can be divided into two passes.

- In the first pass we focus entirely on identifying markup or tags in the document. This could be done using a tokenizer and parser designed for the specific markup language used (e.g., HTML). One such HTML parser in Python is called Beautiful Soup which is a Python library for pulling data out of HTML and XML files.

- In the second pass we focus on the appropriate parts of the document structure (e.g. headings or body text or tables.. etc). Parts that are not useful for searching, such as those containing HTML code, are ignored in this pass.

Stopwords

- Human language is filled with function words, i.e. words that have little meaning in isolation from other words.
  - e.g. “the,” “a,” “an,” “that,” and “those. These words are part of how we describe nouns in text, and express concepts like location or quantity.
  - Prepositions, such as “over,” “under,” “above,” and “below,” represent relative position between two nouns.
- Two properties of these function words cause us to want to treat them in a special way in text processing.
  - These function words are extremely common in English (see table); however, they rarely indicate anything about document relevance on their own.
  - If we are considering individual words in the retrieval process and not phrases, these function words will help us very little.
- They are also called stopwords because text processing usually stops when one is seen, and they are thrown out. Throwing out these words decreases index size, increases retrieval efficiency, and generally improves retrieval effectiveness. But on the other hand, removing too many of these will negatively impact retrieval effectiveness. For instance, the query ”to be or not to be” consists entirely of words that are usually considered stopwords.

Word	*Freq.*	*r*	P_r(%)	$r.P_r$	*Word*	*Freq*	*r*	P_r(%)	$r.P_r$
the	2,420,778	1	6.49	0.065	has	136,007	26	0.37	0.095
of	1,045,733	2	2.80	0.056	are	130,322	27	0.35	0.094
to	968,882	3	2.60	0.078	not	127,493	28	0.34	0.096
a	892,429	4	2.39	0.096	who	116,364	29	0.31	0.090
and	865,644	5	2.32	0.120	they	111,024	30	0.30	0.089
in	847,825	6	2.27	0.140	its	111,021	31	0.30	0.092
said	504,593	7	1.35	0.095	had	103,943	32	0.28	0.089
for	363,865	8	0.98	0.078	will	102,949	33	0.28	0.091
that	347,072	9	0.93	0.084	would	99,503	34	0.27	0.091
was	293,027	10	0.79	0.079	about	92,983	35	0.25	0.087
on	291,947	11	0.78	0.086	i	92,005	36	0.25	0.089
he	250,919	12	0.67	0.081	been	88,786	37	0.24	0.088
is	245,843	13	0.65	0.086	this	87,286	38	0.23	0.089
with	223,846	14	0.60	0.084	their	84,638	39	0.23	0.089
at	210,064	15	0.56	0.085	new	83,449	40	0.22	0.090
by	209,586	16	0.56	0.090	or	81,796	41	0.22	0.090
it	195,621	17	0.52	0.089	which	80,385	42	0.22	0.091
from	189,451	18	0.51	0.091	we	80,245	43	0.22	0.093
as	181,714	19	0.49	0.093	more	76,388	44	0.21	0.090
be	157,300	20	0.42	0.084	after	75,165	45	0.20	0.091
were	153,913	21	0.41	0.087	us	72,045	46	0.19	0.089
an	152,576	22	0.41	0.090	percent	71,956	47	0.19	0.091
have	149,749	23	0.40	0.092	up	71,082	48	0.19	0.092
his	142,285	24	0.38	0.092	one	70,266	49	0.19	0.092
but	140,880	25	0.38	0.094	people	68,988	50	0.19	0.093

Stemming

- Stemming, also called conflation, is a component of text processing that captures the relationships between different variations of a word.

- More precisely, stemming reduces the different forms of a word that occur because of inflection (e.g. plurals, tenses) or derivation (e.g. making a verb into a noun by adding the suffix -ation) to a common stem.

- For example to search for Mark Spitz's Olympic swimming career, you might type "Mark Spitz swimming" into a search engine, but a relevant page might contain the word swam. It is the job of the stemmer to reduce "swimming" and "swam" to the same stem (swim) and thereby allow the search engine to determine that there is a match between these two words.

- In general, using a stemmer for search applications with English text produces a small but noticeable improvement in the quality of results. However, in highly inflected languages, such as Arabic or Russian, stemming is a crucial part of effective search.

Stemmer Types

There are two basic types of stemmers: algorithmic and dictionary-based

- An algorithmic stemmer uses a small program to decide whether two words are related, usually based on knowledge of word suffixes for a particular language
- By contrast, a dictionary-based stemmer has no logic of its own, but instead relies on pre-created dictionaries of related terms to store term relationships (e.g. swimming and swim)
- In dictionary-based stemmers, the related words do not even need to look similar; a dictionary stemmer can recognize that “is,” “be,” and “was” are all forms of the same verb.

Porter stemmer

- One of the most popular algorithmic stemmers is the Porter stemmer (dating back to the 1970s)
- The stemmer consists of a number of steps, each containing a set of rules for removing suffixes.
- At each step, the rule for *the longest applicable suffix* is executed.
- As an example, here are the part a of step 1 (of 5 steps) of the Porter stemmer

Step 1a:

- Replace *sses* by *ss* (e.g., stresses→stress).
- Delete *s* if the preceding word part contains a vowel not immediately before the *s* (e.g., gaps→gap but gas→gas).
- Replace *ied* or *ies* by *i* if preceded by more than one letter, otherwise by *ie* (e.g., ties→tie, cries→cri).
- If suffix is *us* or *ss* do nothing (e.g., stress→stress).

- The original version of the Porter stemmer made a number of errors
- A more recent form of the stemmer (called Porter2) fixes some of these problems and provides a mechanism to specify exceptions.
- The stemmer is also available for many other languages, such as Russian and Turkish …

Highly Inflectional Languages

- Some languages are highly inflectional which means that the root word can have many variants, e.g. Arabic and Spanish. For example, the following table shows some Arabic words derived from the same root.

- Clearly, a stemming algorithm that reduced Arabic words to their roots would not help in search (there are less than 2,000 roots in Arabic), but a broad range of prefixes and suffixes must be considered.

- In a highly inflectional language, proper stemming can make a large difference to the accuracy of the ranking. An Arabic search engine with high-quality stemming can be more than 50% more effective, on average, at finding relevant documents than a system without stemming.

- By contrast, improvements for an English search engine vary from less than 5% on average for large collections to about 10% for small, domain-specific collections.

kitab	*a book*
kitabi	*my book*
alkitab	*the book*
kitabuki	*your book* (f)
kitabuka	*your book (m)*
kitabuhu	*his book*
kataba	*to write*
maktaba	*library, bookstore*
maktab	*office*

Table 4.8. Examples of words with the Arabic root **ktb**

Phrases and N-grams

- Many of the two- and three-word queries submitted to search engines are phrases, and finding documents that contain those phrases will be part of any effective ranking algorithm.
- Phrases are more precise than single words as topic descriptions (e.g., “tropical fish” versus “fish”) and usually less ambiguous
- The impact of phrases on retrieval can be complex; for example given a query such as “fishing supplies”, should the retrieved documents contain exactly that phrase, or should they get credit for containing the words “fish“, “fishing”, and “supplies” in the same paragraph, or even the same document? The details of how phrases affect ranking will depend on the specific retrieval model that is incorporated into the search engine
- There are a number of possible definitions of a phrase, Since a phrase has a grammatical definition, it seems reasonable to identify phrases using the syntactic structure of sentences. The definition of a phrase that is used most frequently in information retrieval is that a phrase is equivalent to a simple noun phrase. This is often restricted even further to include just sequences of nouns, or adjectives followed by nouns.
- Phrases defined by these criteria can be identified using a part-of-speech (POS) tagger.

Part-of-speech (POS) taggers

- A POS tagger marks the words in a text with labels corresponding to the part-of-speech of the word in that context.
- Taggers are based on statistical or rule-based approaches and are trained using large corpora that have been manually labelled.
- Typical tags that are used to label the words include NN (singular noun), NNS (plural noun), VB (verb), VBD (verb, past tense), VBN (verb, past participle), IN (preposition, e.g. in, out), JJ (adjective), CC (conjunction, e.g., “and”, “or”), PRP (pronoun, e.g. she, it), and MD (modal auxiliary, e.g., “can”, “will”).

Original text:
Document will describe marketing strategies carried out by U.S. companies for their agricultural chemicals, report predictions for market share of such chemicals, or report market statistics for agrochemicals, pesticide, herbicide, fungicide, insecticide, fertilizer, predicted sales, market share, stimulate demand, price cut, volume of sales.

Brill tagger:
Document/NN will/MD describe/VB marketing/NN strategies/NNS carried/VBD out/IN by/IN U.S./NNP companies/NNS for/IN their/PRP agricultural/JJ chemicals/NNS ,/, report/NN predictions/NNS for/IN market/NN share/NN of/IN such/JJ chemicals/NNS ,/, or/CC report/NN market/NN statistics/NNS for/IN agrochemicals/NNS ,/, pesticide/NN ,/, herbicide/NN ,/, fungicide/NN ,/, insecticide/NN ,/, fertilizer/NN ,/, predicted/VBN sales/NNS ,/, market/NN share/NN ,/, stimulate/VB demand/NN ,/, price/NN cut/NN ,/, volume/NN of/IN sales/NNS ./.

That's it Folks

Further Reading

Chapter 4: Search Engines Information Retrieval in Practice by W. Bruce Croft, Donald Metzler, and Trevor Strohman