

Introduction to Corpus Resources, Annotation and Access: *Tokenisation*

Sabine Schulte im Walde
Universität Stuttgart

Foundational Course
Departament de Traducció i Filologia
Universitat Pompeu Fabra
April 16-20, 2007

Tokenisation / Text Segmentation

- Tokenisation divides the **raw input character sequence** of a text into **sentences** and the sentences into **tokens**.
- What is a token?
 - » words: *time / as / 40. / House / runs ...*
 - » punctuation: *! “ ,) ...*
- Simple tokeniser: *Split the character sequence at white-space positions and cut off punctuation, to obtain the sequence of tokens.*
- Problem: ambiguities, mainly caused by periods
- Errors made at this stage are very likely to cause more errors at later stages (morphology, syntax, etc.).

Tokenisation Problems

- Major problem categories:
 - » disambiguation of sentence boundaries
 - » normalisation of capitalised words
 - » identification of abbreviations
- Language-dependent task:
 - » Each language has different patterns.
 - » The language families **alphabetic vs. ideographic** languages differ strongly. Ideographic languages provide less information (on punctuation, spaces, etc.)
- Common problem in ambiguous cases:
disambiguation of periods

Excursus: Regular Expressions

- Origin: automata theory and formal languages
- A regular expression is a string that describes a set of strings, without having to list all elements of that set.
- Regular expressions are used (among others) to search and manipulate bodies of text based on patterns.
- Some programming languages include the concept of regular expression matching, e.g. Perl, Python, Tcl.
- Basic concepts (examples):
 - » **set:** `[.,!?` `[0-9]` `[A-Za-z]`
 - » **alternation:** `grey | gray` `gr(e|a)y`
 - » **quantification:** `colou?r` `[0-9]+[.]` `([0-9])*` `[.]` `[0-9]+`

Tokenisation: *Numbers*

- Numbers are language-specific, e.g.
 - » English: 123,456.78
 - » French: 123 456,78
 - » German: 123.456,78
- Numbers are the least ambiguous of the structural types.
- Regular expressions can help to recognise numbers:
 - » English: $([0-9]+[,])^* [0-9] ([.][0-9]+)?$
 - » French: $([0-9]+[])^* [0-9] ([,][0-9]+)?$
 - » German: $([0-9]+[.])^* [0-9] ([,][0-9]+)?$

(Note: These strings overgenerate, but are still sufficient for recognition.)

Tokenisation: *Abbreviations*

- Three classes of abbreviations:
 - » a single capital followed by a period: *A. / B. / C.*
 - » a sequence of letter-period-letter-period's: *m.p.h.*
 - » a capital letter followed by a sequence of consonants followed by a period: *Mr. / Dr. / Assn.*
- Abbreviations do not form a closed set, so one cannot list all possible abbreviations.
- Abbreviations can coincide with regular words, e.g.
no - abbreviation of *number* vs. negation

Tokenisation: *Capitalisation*

- In English mixed-case texts capitalised words usually denote proper names, but there are special positions in the text where capitalisation is expected:
 - » first word in a sentence
 - » words in titles
 - » after a colon or open quote ...
- Capitalised words can be ambiguous, such as *continental* vs. *Continental*.
- In languages with capitalisation (such as German), the degree of ambiguity increases, because capitalised words a priori represent more than only proper names.

Tokenisation: *Sentence Boundaries*

- A period, an exclamation mark, or a question mark usually signals a sentence boundary.
- Other functions of periods:
 - » decimal point
 - » part of an abbreviation
 - » end-of-sentence indicator (full-stop) and at the same time part of an abbreviation
- Examples:
 - Anna went home late . Her father was angry .*
 - Anna came back from the U . S . A . last month .*
 - Anna came back from the U . S . A . She enjoyed it .*
 - Anna came back from the U . S . A . Continental ...*

Tokenisation: *Capitalisation - Sentences*

- The disambiguation of capitalised words and sentence boundaries presents a **chicken-and-egg problem**.
- If we know that a capitalised word that follows a period is a common word, we can assign such period a sentence terminal.
- On the other hand, if we know that a period is not sentence terminal, then we can conclude that the following capitalised word is a proper name.

Tokenisation: *Hyphenation*

- Given a line ending with some string *x* and a hyphen, followed by a line starting with some string *y*, the tokeniser needs to disambiguate between three possible output strings:
 - » string *xy* which was split because of a newline:
conclu-sion, de-pends
 - » string *x-y* (a hyphenated word): *part-of-speech*
 - » string *x- y* (a truncated word): *on- and offline*
- Words which were split in the raw text to fit the length of a line to the width of the column need to be rejoined.
→ *dehyphenation*

Tokenisation: *Multiword Expressions*

- Assumption: Tokens do not contain whitespace.
- Problem: Multiword expressions contain whitespace. Do they represent one or several tokens?
- Examples:
 - » *Feb. 1, 2004*
 - » *Daimler Chrysler AG*
 - » *because of*
- For some applications it is advantageous to treat multiword expressions as a single token.

Tokenisation: *Clitics*

- Clitics combine several tokens without separating whitespace.
- Examples:
 - » English: *isn't we'll*
 - » French: *Permettez-vous?*
 - » German: *Stimmt's?*
 - » Spanish: *garantizarles*
 - » Italian: *applicarlo*

Disambiguation Methods

Heuristics and information sources:

- Dictionary information
- Abbreviation lists (manual/automatic)
- Sentence positions ...

Heuristics-based approaches:

- Define heuristics about correspondences between a token and a set of classes.
- Define heuristics as rules and order the rules according to their reliability.

Classification approaches (supervised/unsupervised):

- Decision trees, neural networks, maximum entropy, etc.

Tokenisation: References

- Gregory Grefenstette and Pasi Tapanainen (1994): "What is a word, what is a sentence? Problems of tokenization." In *Proceedings of the 3rd Conference on Computational Lexicography and Text Research*, pp. 79-87. Budapest, Hungary.
- Andrei Mikheev (2002): "Periods, Capitalized Words, etc." *Computational Linguistics*, 28(3):289-318.
- Andrei Mikheev (2003): "Text segmentation". In: Ruslan Mitkov, editor: "The Oxford Handbook of Computational Linguistics", pp. 376-394. Oxford University Press.
- Helmut Schmid (2007?): "Tokenizing". In: Anke Lüdeling and Merja Kytö, editors: "Corpus Linguistics. An International Handbook." Mouton de Gruyter, Berlin.

Tokenisation: Tools

- LT TTT, versions 1.0 and 3.0 beta:

<http://www.ltg.ed.ac.uk/software/ttt/>

Edinburgh Language Technology Group: text tokenization and toolset

Steps: 1. Plain text to sgml text (if necessary), 2. Identify titles and paragraphs, 3. Identify words, 4. Disambiguate fullstops, 5. Recognise named entities (numbers, quantities, money, percent, date and time), 6. Conversion to output format (if necessary)

- MXTERMINATOR:

<http://www.id.cbs.dk/~dh/corpus/tools/MXTERMINATOR.html>

Jeffrey C. Reynar and Adwait Ratnaparkhi (1997): “A maximum entropy approach to identifying sentence boundaries”. In *Proceedings of the 5th Conference on Applied Natural Language Processing*. Washington, D.C.

Type/Token Frequency Distributions

Types and Tokens

- **Token**: total number of word instances in a corpus
→ corpus size

Peter₁ 's₂ father₃ is₄ a₅ cook₆ .₇

Peter₈ 's₉ mother₁₀ is₁₁ also₁₂ a₁₃ cook₁₄ .₁₅

- **Types**: number of distinct words in a corpus
→ vocabulary size

Peter₁ 's₂ father₃ is₄ a₅ cook₆ .₇

Peter₇ 's₇ mother₈ is₈ also₉ a₉ cook₉ .₉

Types and Tokens

- Frequency information is **distinctive** to corpus-based methodologies.
- What is counted?
 - » all the instances (**tokens**)
 - » all the distinct words (**types**) in the corpus
- Count the tokens and the corresponding types:
 - » number of tokens → **corpus size N**
 - » number of types → **vocabulary size V**

Token-Type Mapping

- Determination of tokens
 - » text segmentation, 'tokenisation'
- Mapping of tokens to types
 - » normalise upper and lower case?
 - » lemmatise inflected word forms?
- What is not distinguished?
 - » different word senses

Basics for Lexical Statistics

frequency list

type	f	type	f
across	1	see	2
also	1	supporter	1
bridge	2	the	3
can	1	Wayne	1
he	1		

rank / frequency profile

r	f	r	f
1	3	6	1
2	2	7	1
3	2	8	1
4	1	9	1
5	1		

frequency spectrum

f	V(f)
1	6
2	2
3	1

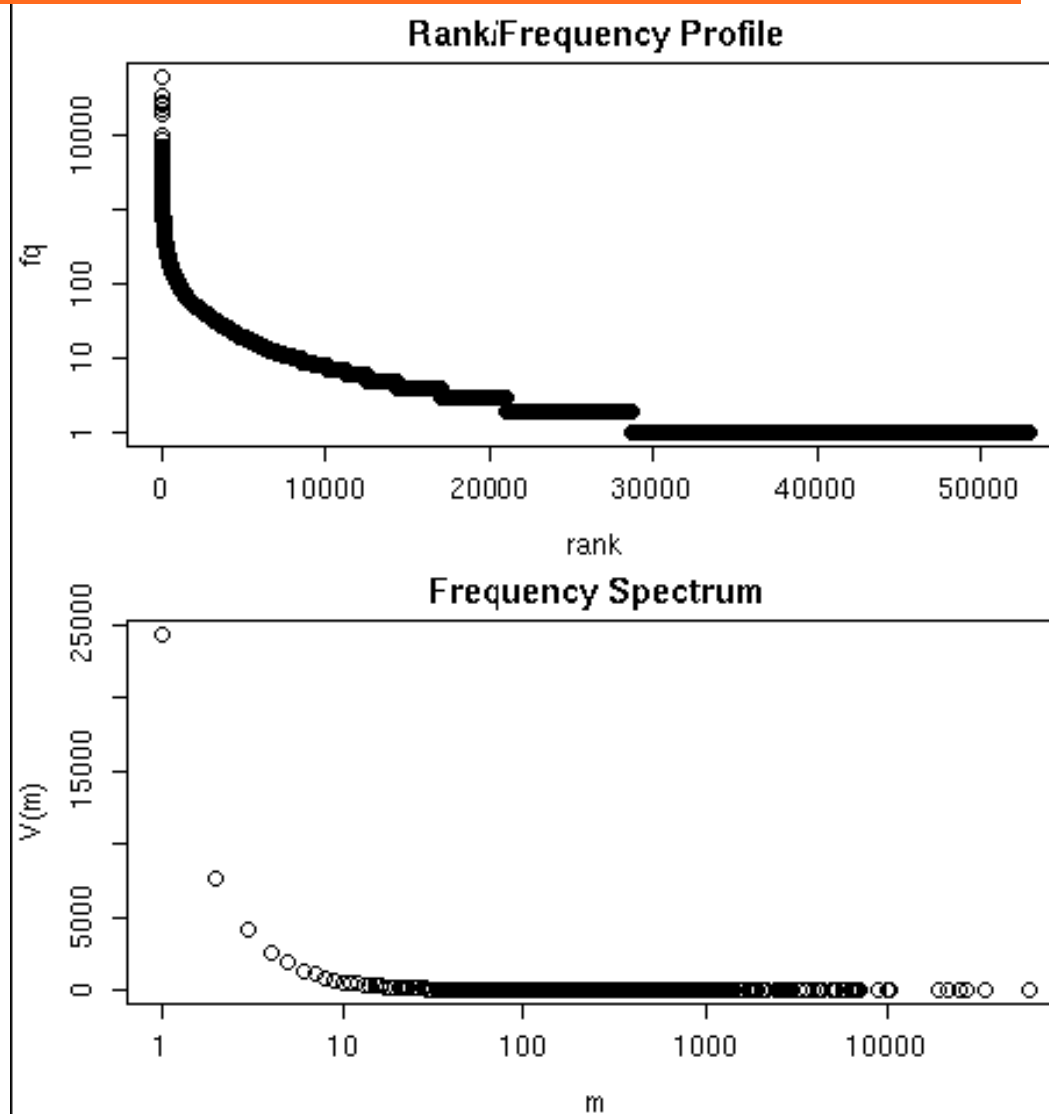
Frequencies of Brown Corpus

<i>top frequencies</i>			<i>bottom frequencies</i>		
rank	fq	word	rank range	fq	randomly selected examples
1	62642	the	7967-8522	10	recordings undergone privileges
2	35971	of	8523-9236	9	Leonard indulge creativity
3	27831	and	9237-10042	8	unnatural Lolotte authenticity
4	25608	to	10043-11185	7	diffraction Augusta postpone
5	21883	a	11186-12510	6	uniformly throttle agglutinin
6	19474	in	12511-14369	5	Bud Councilman immoral
7	10292	that	14370-16938	4	verification gleamed groin
8	10026	is	16939-21076	3	Princes nonspecifically Arger
9	9887	was	21077-28701	2	blitz pertinence arson
10	8811	for	28702-53076	1	Salaries Evensen parentheses

Table 4: Top and bottom of the Brown frequency list

(Baroni, prefinal: 5)

Frequencies of Brown Corpus



(Baroni, prefinal: 7)

Zipf's law

Frequency is a **non-linearly decreasing function of rank**.

- It decreases more sharply among high ranks than among low ranks.
- 'Large number of rare events' (**LNRE**) distribution
- First studied by George Kingsley Zipf (1949, 1965)
- Zipf's law predicts the frequency of a word given its rank:

$$f(w) = \frac{C}{r(w)^a}$$

$f(w)$ = frequency of word w

$r(w)$ = rank of word w

C = frequency of most frequent word

a = a constant

Zipf's law

$$f(w) = \frac{C}{r(w)^a}$$

Given $C = 60,000$ and $a = 1$:

$r(w)$	$f(w)$	$r(w)$	$f(w)$
2	$60,000 / 2 =$ 30,000	100	$60,000/100=$ 600.00
3	$60,000 / 3 =$ 20,000	101	$60,000/101=$ 594.06
...	...	102	$60.000/102=$ 588,23

→ about **80,000** words have $f(w)$ between 1.5 and 0.5

Frequency Distributions: References

- George K. Zipf (1949): "Human Behaviour and the Principle of Least-Effort." Addison-Wesley, Cambridge, MA.
- Christopher D. Manning and Hinrich Schütze (1999): "Foundations of Statistical Natural Language Learning", section 1.4. MIT Press.
- Harald Baayen (2001): "Word frequency distributions". Kluwer.
- Marco Baroni (2007?): "Distributions in Text". In: Anke Lüdeling and Merja Kytö, editors: "Corpus Linguistics. An International Handbook." Mouton de Gruyter, Berlin.