Preliminary Steps for Building Linguistic Resources Tokenization PART-2

Resources for Natural Language Processing
LM Language Technologies and Digital Humanities
2024-25

Cristina Bosco

Token and tokenisation

The **token** seems to correspond to the word as
a **character string between two empty spaces**.

The **tokenization** algorithms are based on the principle
to **split the text according to empty spaces**.

But this principle doesn't work in all:

- cases
- writing style
- textual genres
- languages

Token/lexeme/morpheme

In traditional linguistics, **word** is not the only notion used to indicate the basic component of the linguistic analysis, are used also:

- **LEXEME** (also called LEMMA): the form of the word used to cite it in a dictionary, the basic form
  Example:
    the lexeme for the word *CAT* is 'CAT'

- **MORPHEME**: the elements used to form a word
  Example:
    the word *SLEEPING* is the composition of
    the (lexical) morpheme ‘SLEEP-' and
    the (grammatical) morpheme 'ING

Token/lexeme/morpheme

In traditional linguistics:

- **LEXEME** is the smallest component of lexicon

- **MORPHEME** is the smallest unit of morphology

In NLP lexicon and morphology are encompassed in the notion of **TOKEN**.

Token / lexeme / morpheme

The lexemes and morphemes are not tokens.

Apparently they seem to be the same, but they are not the same in principle.

Example:
CAT is a token and a lexeme
but CATS is a token but not a lexeme

Example:
SLEEP is a token and a morpheme
but SLEEPS is a token but not a morpheme

Tokenization

In all cases, tokenization requires the definition of clear and operative principles.

For each task, you need to know exactly what output you want to produce for each input: For each string in the input, you need to know exactly how many and which tokens it corresponds to.

So you need to know the type of input very precisely, have a clear definition of the tokens and know how this definition corresponds to what we find in the text.

Challenges in Tokenization

The tokenization can be challenging:

- For specific tokens and words
- Because of the style of writing
- Because of the language
- Because of the textual genre

Tokenization

Tokenization is the first step of the analysis and enables the application of the subsequent analyses, which are sure that the units of morphological information must be well distinguished.

In this way, we apply a first abstraction that also allows us to treat different languages in the same way.

Once the text is reduced to tokens, I can treat it in the same way, considering that in the following analysis we have to deal with consistent units.

NLP works in the direction of developing multilingual systems, assuming that in the initial stages of analysis it is possible to reduce the input to something uniform for all languages.

Challenges in Tokenization

The tokenization of a text means that the character sequences are divided into minimal units of analysis, called "**tokens**", in order to consider them as minimal elements that cannot be further divided.

A token within a sentence can correspond to:
- a word (*cat*), a number (*3,234*), a punctuation mark (;), a date (*12/31/2021*), ...
This means that a token can be simple, but also complex, including punctuation markers or symbols.

In languages where the boundaries between words are not marked by spaces, tokenization is called ***word segmentation***, a task based on other principles.

Tokenization

Are tokens the same as words? NO

Words are the elements of the sentence delimited by spaces, tokens are NOT strings delimited by spaces:

Eng: *Kitten* = 1 token

It: *Mandaglielo (Send him it)* = 3 tokens, but only one word
manda (send) + glie (him) + lo (it)

It: nei (in they) = 2 tokens, but only one word
In (in) + i (they)

Tokenization

In different language tokenisation can involve different issues, e.g. in an agglutinative language.

In English it is easy, but in Finnish (a language which is strongly agglutinative):

- kirja = book
- kirja*ni* = my book
- kirja*ssa* = in the book

Tokenization in different languages

One of the differences between Chinese and English is that Chinese is written in characters. The Chinese characters store ideas while English phrases store pronunciation.
In Chinese, there is no whitespace in between phrases.
So simply splitting based on whitespace like in English may not work as well as in English.
For this type of languages tokenisation is called word segmentation.

Tokenization and style of writing

The problems with tokenization may also depend on the graphical conventions.

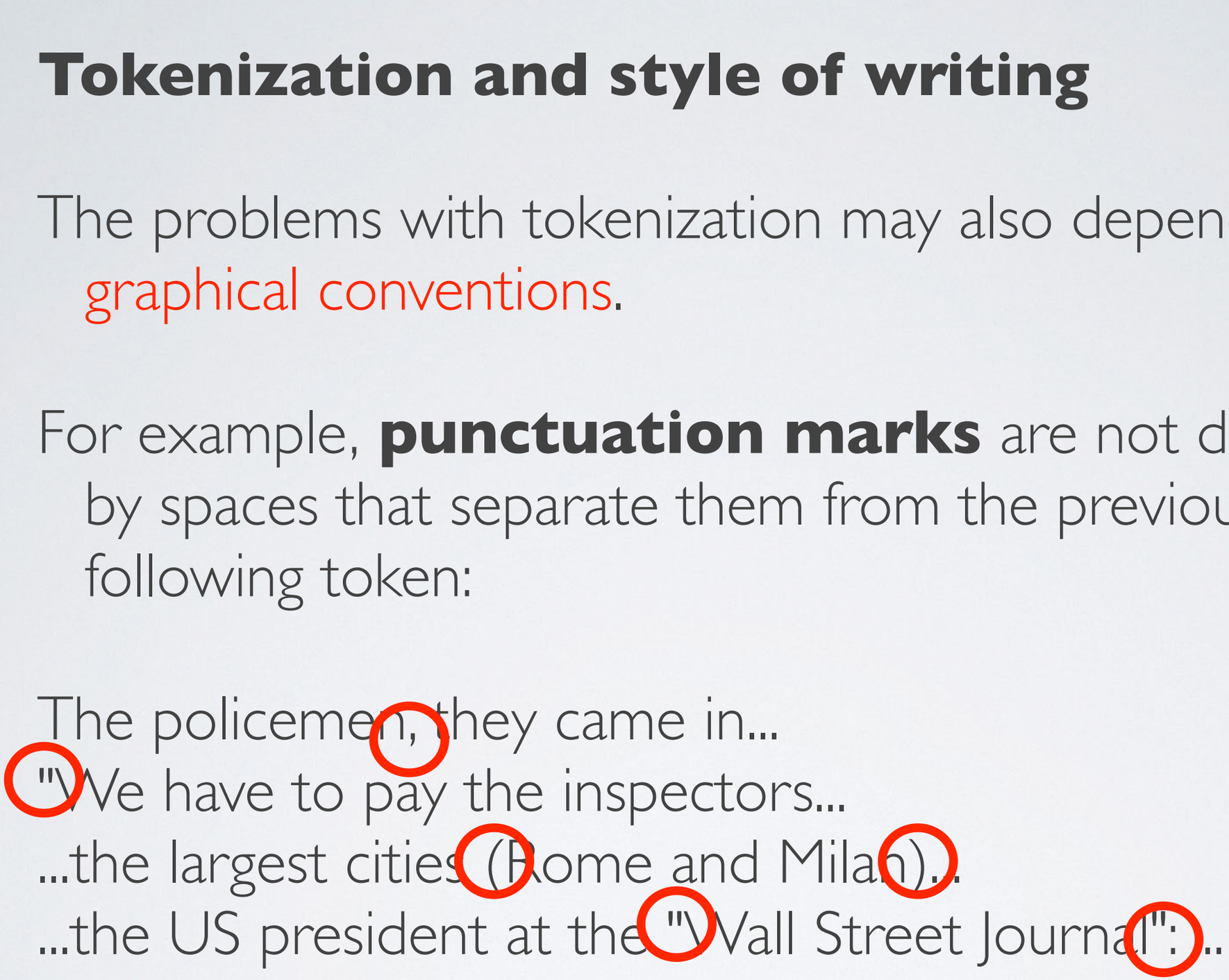
For example, **punctuation marks** are not delimited by spaces that separate them from the previous or following token:

The policemen, they came in...
"We have to pay the inspectors...
...the largest cities (Rome and Milan)...
...the US president at the "Wall Street Journal": ...

Tokenization and style of writing

The problems with tokenization may also depend on the graphical conventions.

For example, **punctuation marks** are not delimited by spaces that separate them from the previous or following token:



Tokenization and style of writing

The problems with tokenization may also depend on the graphical conventions.

The **apostrophe** that appears when the article is elided is not preceded or followed by a space:

It: l'uomo (the man)
Eng: Mary's bar

Tokenization and style of writing

The problems with tokenization may also depend on the graphical conventions.

The **apostrophe** that appears when the article is elided is not preceded or followed by a space:

It: l'uomo (the man)
Eng: Mary's bar

Tokenization and punctuation

The character . (dot) can also have different uses:

- at the end of a sentence
- in abbreviations: Mr. Verdi
- in addresses: C.so Moncalieri 33
- in acronyms: U.S.A
- in numbers as a thousands or decimal separator: 1.375
- in dates: 27.05.2033
- in web or email addresses: www.di.unito.it

Depending on the case, it may or may not be regarded as a token separate from the preceding word.
These decisions also affect the segmentation of text into sentences that is considered in analysis that follow tokenisation

Tokenization and multi-word expressions

Sometimes several strings separated by spaces can form a single token, these are multi-word expressions:

- compound proper names: New York, Borgo San Dalmazzo

- polyrematic expressions (multi-word expressions): above, ad hoc, compared to ...

- dates and times: February 7, 2021 at 13:30

Tokenization and text genres: social media

The application of tokenization to certain text genres can be particularly problematic

#governmentUSA

(#governmentUSA)

:-)

S T O P

C-;

http://t.co/vTE7YOWP

HOW MANY TOKENS ?

Eng: **I like it!!!!!!!!!!!!!!!!** **I like it!**

It: **tvb** **ti voglio bene** (I love you)

Fr: **manif** **manifestation**

Tokenization

Tokenization forces us to ask ourselves how the words we find in the sentences are made:

Are the words simple or compound?

Can words contain spaces?

Can a word be tokenized in a way only?

There are no *right answers* to these questions, but choices that are more or less standard and shared by those working on the task or using its result.

Sticking to a **standard** means creating a reusable result from others.

Tokenization

Tokenization forces us to define criteria and rules that allow us to choose a solution in all possible cases.

The rules may also depend on the purpose for which the tokenization is applied.
They are also based on directories and glossaries that list the different ways of representing a token:

- database, data-base and data base

Tokenization

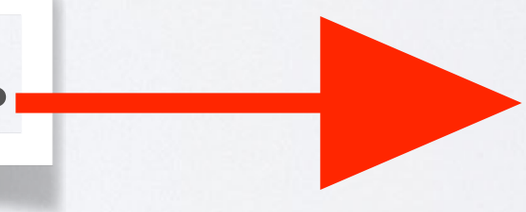
Tokenization is followed by **lemmatization** and **part-of-speech tagging**, that deal with the analysis of each individual token in the context of the morphological profile.

What does a tokenizer do?

- takes in input a sentence
- outputs individual tokens, usually one token per line, thus simplifying subsequent processing.

OUTPUT OF TOKENIZATION

The dog is running in the garden.



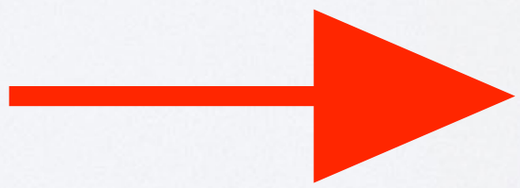
The
dog
is
running
in
the
garden
.

TOKENIZATION

Il cane del mio amico

il
cane
???
mio
amico

Dimmelo!



???

OUTPUT OF TOKENIZATION

The output of a tokenizer in XML:

- *After sleeping the man woke up*
- <token n="1">**After**</token> <token n="2">**sleeping**</token> <token n="3">**the**</token> <token n="4">**man**</token> <token n="5">**woke**</token> <token n="6">**up**</token>