Basic Tokenisation

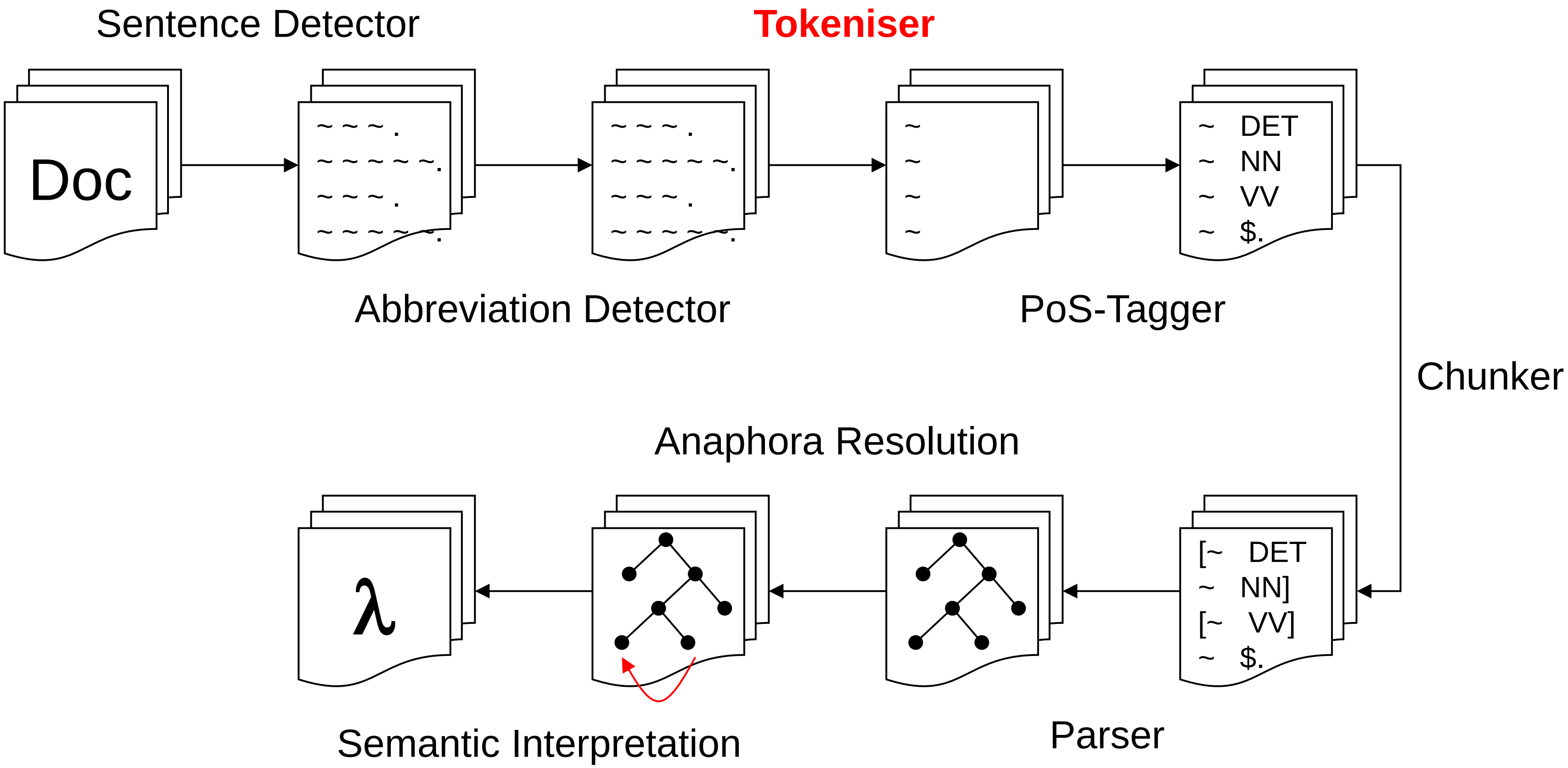
- Radical, but Consistent -

Michael Poprat[1], Harald Kirsch[2]

[1]Language & Information Engineering Lab, Jena University, Germany

[2]European Bioinformatics Institute, Hinxton, UK

Tokenisation in an NLP Pipeline



What Is a Token?

?

Definition of Basic Tokenisation Rules

To be applied sequentially:

Rule 1:
An entity that is surrounded by any kind of white space, is a token.

Rule 2:
Any non-alphanumeric character is a position to split an entity into tokens. The non-alphanumeric character is a token itself.

Rule 3:
Any alpha character followed directly by a numeric character is a position to split an entity into tokens.

Example Processing

Multifactorial contributions to an acute DNA damage response by BRCA1/BARD1-containing complexes.

Rule 1

[Multifactorial] [contributions] [to] [an] [acute] [DNA] [damage] [response] [by] [BRCA1/BARD1-containing] [complexes.]

Rule 2

[Multifactorial] [contributions] [to] [an] [acute] [DNA] [damage] [response] [by] [BRCA1] [/] [BARD1] [-] [containing] [complexes] [.]

Rule 3

[Multifactorial] [contributions] [to] [an] [acute] [DNA] [damage] [response] [by] [BRCA] [1] [/] [BARD] [1] [-] [containing] [complexes] [.]

Why Rule 1 (Is not Enough)?

Rule 1:
An entity that is surrounded by any kind of white space, is a token.

the ‘regular mixed practitioner‘

→ ‘ is a token

from the 5‘ end of the 1-stand

→ ‘ is part of a token (?)

Why Rule 2?

Rule 2:
Any non-alphanumeric character is a position to split an entity into tokens. The non-alphanumeric character is a token itself.

NF-kappa-B vs. NF kappa-B vs. NF-kappa B vs. NF kappa B
→ consistent tokenisation with Rule 2

Why Rule 3?

Rule 3:
Any alpha-character followed directly by a numeric-character is a position to split an entity into tokens.

- BRCA 1, BRCA-1, BRCA1
- BRCA 2, BRCA-2, BRCA2
- ...

is-a BRCA

→ non-standardised spellings can be uniformed in tokens
→ alpha-numeric combinations often point to variations

Why Machine Learning is Not Applicable?

- pre-request: manually annotated corpus

- definition of a token is purpose and domain dependent

  - [IL6-responsive] [gene] → part-of-speech (IL6-responsive/ADJ)

  - [IL6] [-] [responsive] [gene] → named entity recognition (IL6/protein)

  - [IL6] [-] [responsive] [gene] → semantic interpretation, special character [-] (“*a gene that responds to IL6*“)

- no existing tokenised corpus (for the biomedical domain)

- existing annotated corpora are inconsistent (e.g., GENIA)

Known Resources (GENIA)

De facto standard in Bio-NLP, but inconsistent tokenisation:

- **PoS-Annotation and Treebank**

toward/IN humoral/JJ or/CC cell-mediated/JJ immunity/NN*

without/IN TCR-mediated/JJ stimulation/NN*

containing/VBG different/JJ IL-6-responsive/JJ gene/NN elements/NNS[+]

on/IN the/DT induction/NN of/IN endogenous/JJ IL-6-responsive/JJ genes/NNS[+]

* from 93150054

[+] from 96278844

Known Resources (GENIA)

- **NE-Annotation**

toward <cons sem=“other_name“><cons sem=“other_name“>humoral</cons> or <cons lex=“...“>cell-mediated</cons> <cons sem=“other_name“> immunity</cons></cons>*

without <cons sem=“other_name“><cons sem=“protein_family_or_group“> TCR</cons>-mediated stimulation </cons>*

containing different <cons sem=“DNA_family_or_group“>IL-6-responsive gene elements</cons>[+]

on the induction of endogenous <con sem=“DNA_family_or_group“> <con sem=“protein_molecule“>IL-6 </cons>-responsive genes </cons>[+]

[*] from 93150054
[+] from 96278844

But There Is Rule 4:

Don‘t touch annotated entities!

→ highly utilizable

→ highly customisable

by defining modules

<u>Examples</u>:

- nomenclatures (dates, time, URL, chemical formulas?)
    → regulated entities
- named entities, terminologies, acronyms
    → not regulated entities

→ Modules can be applied before or after tokenisation

→ But the modules is not the part of the tokenisation task!

Summarisation & Conclusion

- What is a token?
  - → An entity you don‘t have to look inside for interpretation?
- here: often too fine-grained, but consistent
- but: domain- and purpose-adaptable by applying modules
- future work:
  - programming and providing a Java jar-package
  - defining some example modules
  - testing the effects in an NLP pipeline
  - providing corpora in a tokenised format
- white paper (under development)