Basic Text Processing

Tokenizing

Sentence splitting

Word Normalisation

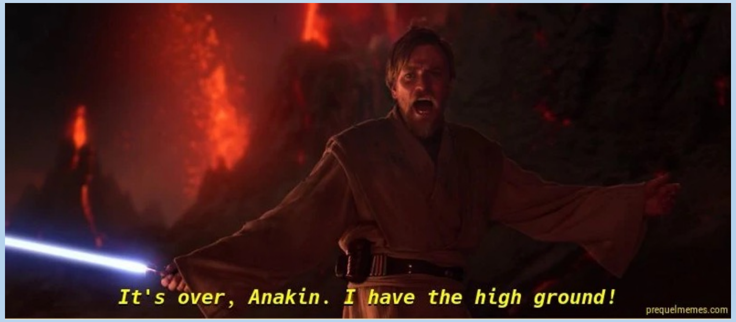
Task description

- Input is plain text (and sometimes the tokens)

It`s over, Anakin. I have the high ground!

- Task is to determine „sentences“

$\boxed{\text{It`s over, Anakin.}}$ $\boxed{\text{I have the high ground!}}$



Sentence splitting

- Challenges:
  - Most common issue is the dot “.”, since it can denote:
    - Abbreviation (Mr.)
    - Enumeration (2.)
    - Punctuation in a number (12.4)
    - End of sentence
    - …
  - Meaning of semi-colon “;” is unspecific, sometimes it breaks sentences, sometimes it does not

Rule Based Sentence Splitting

- The following slides present the PUNKT-Algorithm (available in NLTK)

Unsupervised Multilingual Sentence Boundary Detection

Tibor Kiss*
Ruhr-Universität Bochum

Jan Strunk**
Ruhr-Universität Bochum

ACL 2006, https://www.aclweb.org/anthology/J06-4003.pdf

Formalizing the task

- For now we will restrict ourselves to determining the meaning of the dot "."
- For every occurrence of the dot, we determine:
  - Part of the token (Abbreviation <A>)
  - End of sentence (<S>)
- Strongly connected with tokenization

➔Classical binary decision problem

➔ Or a multi-class classification if there are several categories (abbreviation, part of a number, enumeration,...).

PUNKT-Algorithm

- Idea: We can extract domain specific abbreviations automatically, given a large corpus of text (without labels)
- This results in a procedure which is split into 2 phases:

1. Determine a label for every **word type**
   ➔ We hope to get it right for the majority of a type

2. Correct mistakes introduced in the first phase

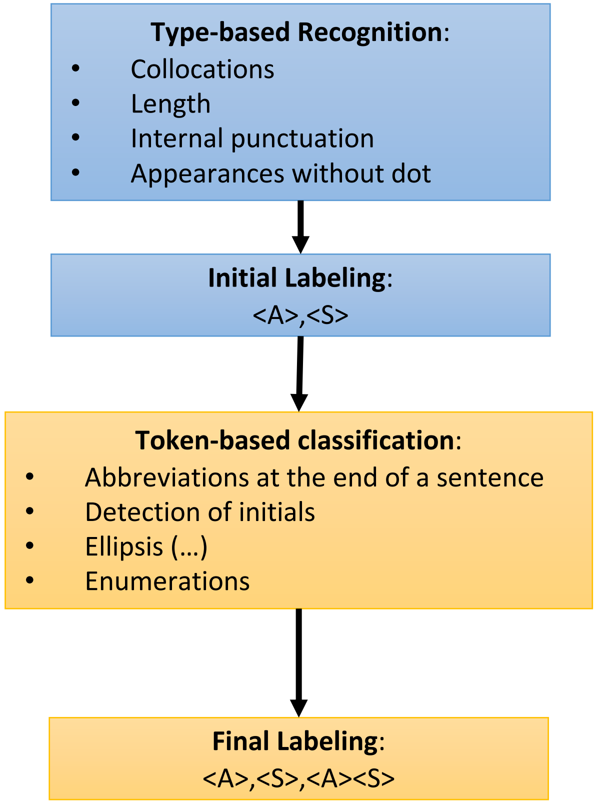
A word type is the form of the token, e.g. „Dr.“, „Mr.“

Every specific appearance of a token

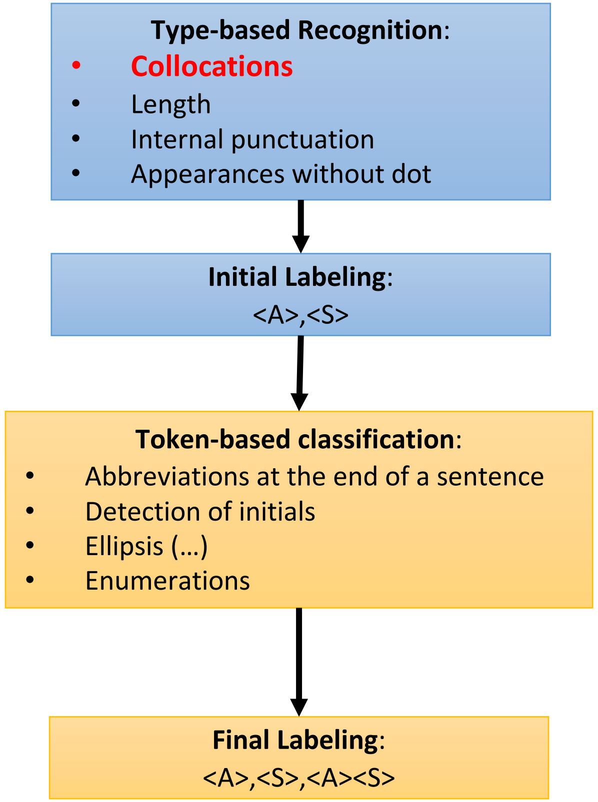
PUNKT-Algorithm

- Basic idea of the algorithm:

<A>: Abbreviation
<S>: End of sentence



PUNKT-Algorithm



Idea:

- Abbreviations should appear more frequently (statistically speaking) with a dot „.“, than without

➔ We can create a statistical test for independence

- We therefore create two hypotheses:

<u>Null-hypothesis:</u> There is no statistical dependency between a word w and a dot •

$$P(\bullet\,|w) = p = P(\bullet\,|\neg w)$$

<u>Alternative-hypothesis:</u> They are not independent

$$P(\bullet\,|w) = p_1 \neq p_2 = P(\bullet\,|\neg w)$$

Likelihood-Ratio Test

- The test we are deriving now is according to Dunning (1993)
- We are deriving the test for arbitrary words w_1 and w_2 (and then apply it to our abbreviation with the dot •)
- But first, we are trying to model our text

We can now count some statistics:

- c_{12} common appearance of the bigram $w_1 w_2$
- c_1 appearances of the word w_1
- c_2 appearances of the word w_2
- N, total amount of words in our text

Likelihood-Ratio Test

We can now count some statistics:

- c_{12} common appearance of the bigram $w_1 w_2$
- c_1 appearances of the word w_1
- c_2 appearances of the word w_2
- N, total amount of words in our text

We observe:

$$p(W_2|W_1) = \frac{p(W_1,\ W_2)}{p(W_1)} = \frac{\frac{c_{12}}{N}}{\frac{c_1}{N}} = \frac{c_{12}}{c_1}$$

$$p(W_2|\neg W_1) = \frac{p(\neg W_1,\ W_2)}{p(\neg W_1)} = \frac{c_2 - c_{12}}{N - c_1}$$

We are now trying to get a total probability of the text, given these observations!

Likelihood-Ratio Test

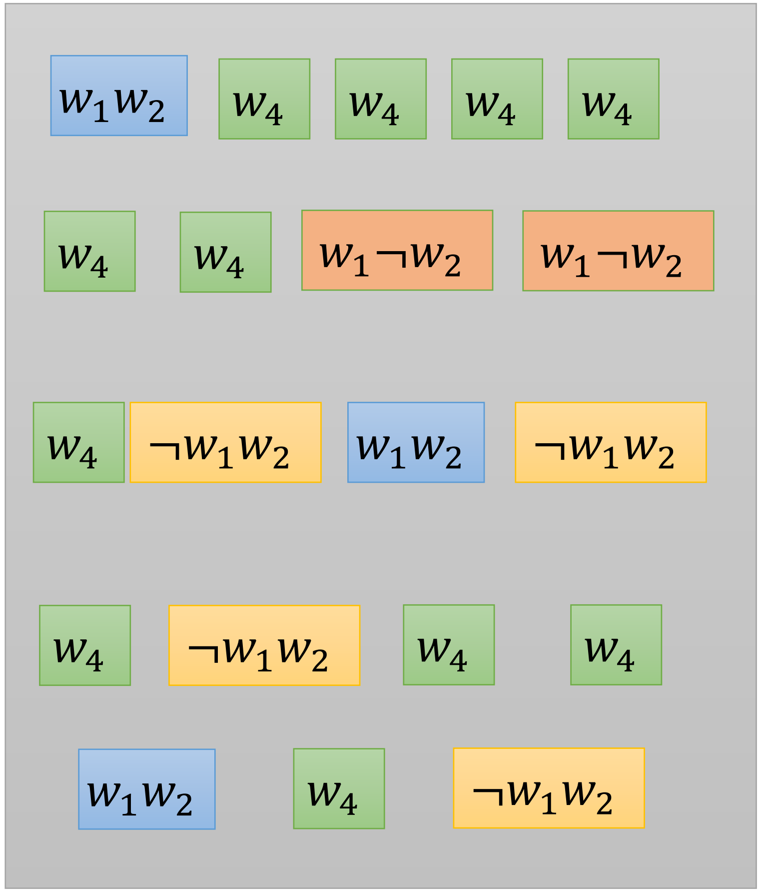
- Modelling the text
  - This test assumes, that our text consists of:

Some appearances of w_1, w_2

Some appearances of $\neg w_1 w_2$

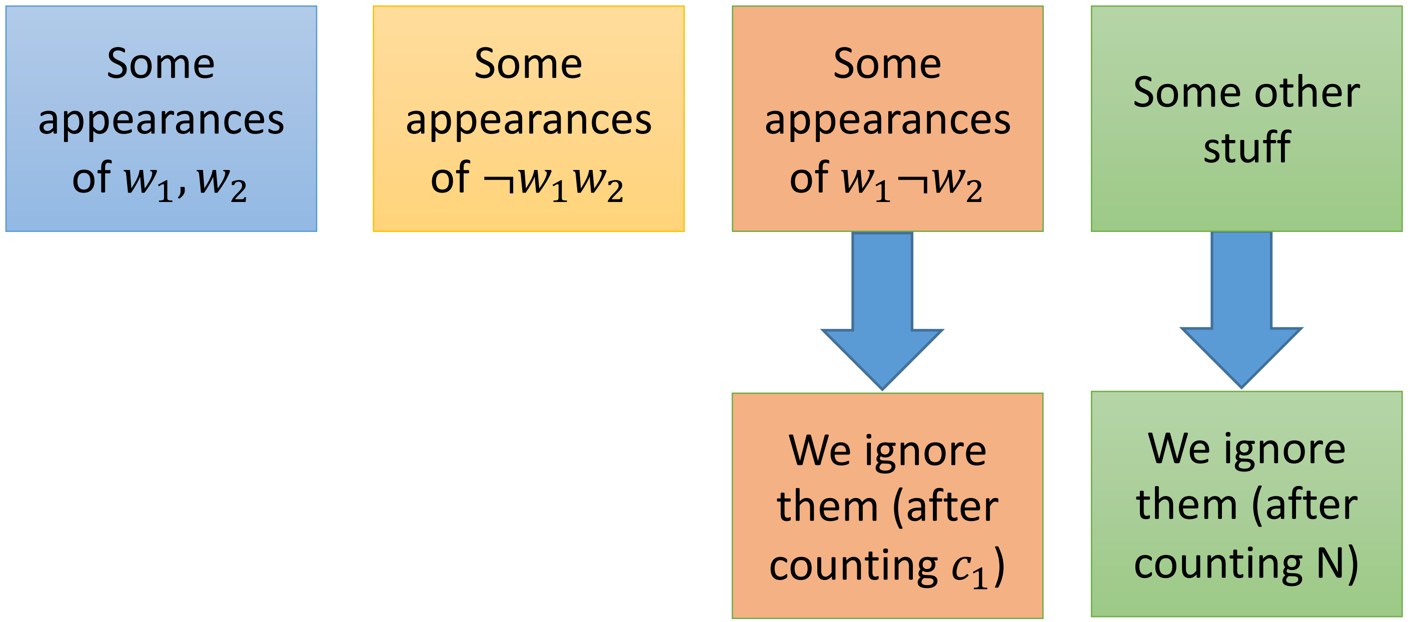
Some appearances of $w_1 \neg w_2$

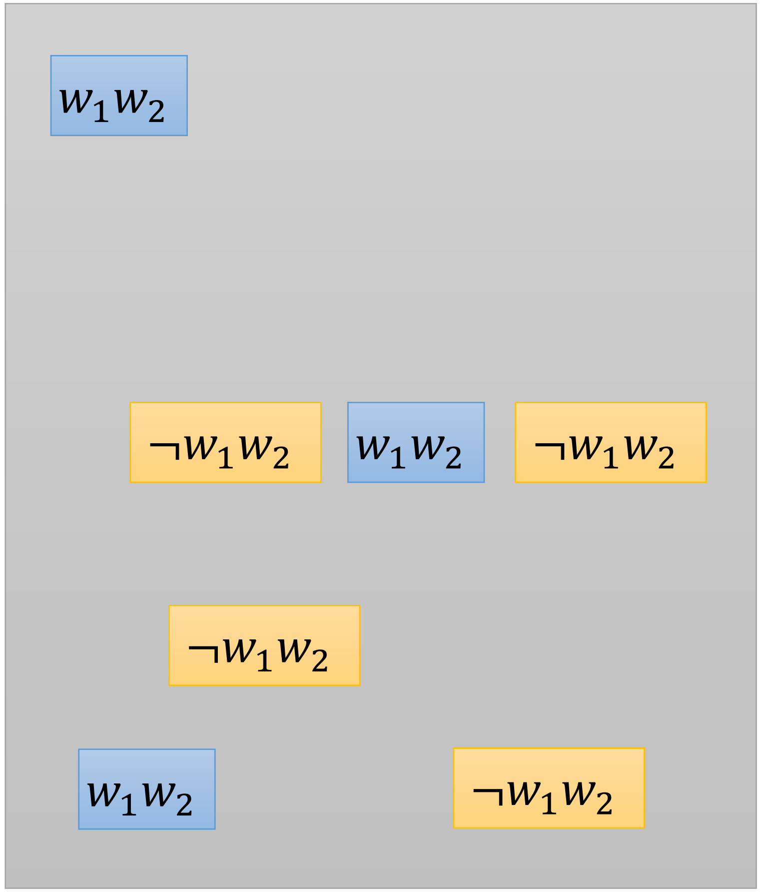
Some other stuff



Likelihood-Ratio Test

- Modelling the text
  - This test assumes, that our text consists of:





- counted c_1
- counted N

Likelihood-Ratio Test

- Modelling the text
  - This test assumes, that our text consists of:

$w_1 w_2$ $w_1 w_2$ $w_1 w_2$

$\neg w_1 w_2$ $\neg w_1 w_2$ $\neg w_1 w_2$

$\neg w_1 w_2$

- counted c_1
- counted N

Some appearances of w_1, w_2

Some appearances of $\neg w_1 w_2$

Some appearances of $w_1 \neg w_2$

Some other stuff

We ignore the order in which they appear (we are only interested in the counts)

We ignore them (after counting c_1)

We ignore them (after counting N)

Likelihood-Ratio Test

- Let us recite our observations:

$w_1 w_2$ $w_1 w_2$ $w_1 w_2$

$\neg w_1 w_2$ $\neg w_1 w_2$ $\neg w_1 w_2$

$\neg w_1 w_2$

- counted c_1
- counted N

We observe:

$$p(W_2|W_1) = \frac{p(W_1,\ W_2)}{p(W_1)} = \frac{\frac{c_{12}}{N}}{\frac{c_1}{N}} = \frac{c_{12}}{c_1}$$

Of c_1 possibilities to observe $w_1 w_2$, we observe it c_{12} times

$$p(W_2|\neg W_1) = \frac{p(\neg W_1,\ W_2)}{p(\neg W_1)} = \frac{c_2 - c_{12}}{N - c_1}$$

➔ We can now model the entire text using the product of 2 Bernoulli distributions, each with their respective observations

Likelihood-Ratio Test

We model the observations: $(W_2|W_1)$ with the following probability:

$$B(c_1, c_{12}, p_1) = \binom{c_1}{c_{12}} p_1^{c_{12}} \cdot (1 - p_1)^{c_1 - c_{12}}$$

$w_1 w_2$ $w_1 w_2$ $w_1 w_2$

- counted c_1

And the observations: $(W_2|\neg W_1)$ with the following probability:

$\neg w_1 w_2$ $\neg w_1 w_2$ $\neg w_1 w_2$

$$B(N - c_1, c_2 - c_{12}, p_2) = \binom{N - c_1}{c_2 - c_{12}} p_2^{c_2 - c_{12}} \cdot (1 - p_2)^{N - c_1 - c_2 + c_{12}}$$

$\neg w_1 w_2$

- counted c_1
- counted N

We can now model the entire observations as the product of both terms:

$$p(\text{observations}) = \binom{c_1}{c_{12}} p_1^{c_{12}} \cdot (1 - p_1)^{c_1 - c_{12}} \cdot \binom{N - c_1}{c_2 - c_{12}} p_2^{c_2 - c_{12}} \cdot (1 - p_2)^{N - c_1 - c_2 + c_{12}}$$

Likelihood-Ratio Test

- Almost there:

Null-hypothesis: There is no statistical dependency between a word w and a dot •

$$p_1 = P(\bullet\,|w) = p = P(\bullet\,|\neg w) = p_2$$

Alternative-hypothesis: They are not independent

$$P(\bullet\,|w) = p_1 \neq p_2 = P(\bullet\,|\neg w)$$

$$H_0 = \binom{c_1}{c_{12}} p^{c_{12}} \cdot (1-p)^{c_1-c_{12}} \cdot \binom{N-c_1}{c_2-c_{12}} p^{c_2-c_{12}} \cdot (1-p)^{N-c_1-c_2+c_{12}}$$

$$H_A = \binom{c_1}{c_{12}} p_1^{c_{12}} \cdot (1-p_1)^{c_1-c_{12}} \cdot \binom{N-c_1}{c_2-c_{12}} p_2^{c_2-c_{12}} \cdot (1-p_2)^{N-c_1-c_2+c_{12}}$$

Likelihood-Ratio Test

- The test now for real:
- We use both our hypotheses and calculate the **Likelihood-Ratio**

$$\lambda = \frac{H_0}{H_A} = \frac{\max_{p} B(c_{12}, c_1, p) \cdot B(c_2 - c_{12}, N - c_1, p)}{\max_{p_1, p_2} B(c_{12}, c_1, p_1) \cdot B(c_2 - c_{12}, N - c_1, p_2)}$$

We compare the most likely manifestations of both hypotheses

Likelihood-Ratio Test

- Maximization of the Hypothesis for p, p_1 and p_2 yields:

$$p = \frac{c_2}{N} \;;\; p_1 = \frac{c_{12}}{c_1} \;;\; p_2 = \frac{c_2 - c_{12}}{N - c_1}$$

- You can recover this by forming the derivations and setting them to zero!

➔ Exercise

Likelihood-Ratio Test

- A small simplification

$$H_0 = \binom{c_1}{c_{12}} p^{c_{12}} \cdot (1-p)^{c_1-c_{12}} \cdot \binom{N-c_1}{c_2-c_{12}} p^{c_2-c_{12}} \cdot (1-p)^{N-c_1-c_2+c_{12}}$$

$$H_A = \binom{c_1}{c_{12}} p_1^{c_{12}} \cdot (1-p_1)^{c_1-c_{12}} \cdot \binom{N-c_1}{c_2-c_{12}} p_2^{c_2-c_{12}} \cdot (1-p_2)^{N-c_1-c_2+c_{12}}$$

- These terms are equal and cancel each other in the fraction!

Likelihood-Ratio Test

- The final form (after applying the logarithm)

$$\log \lambda = log(L(c_{12}, c_1, p)) + \log(L(c_2 - c_{12}, N - c_1, p)) \; - \\ log(L(c_{12}, c_1, p_1\,)) - \log(L(c_2 - c_{12}, N - c_1, p_2))$$

Using: $L(k, n, p) = p^k \cdot (1 - p)^{n-k}$

Note: One can show, that the value $-2 \log \lambda$ follows a Chi-Square distribution with 1 degree of freedom (this means, we can calculate that value and read from a precalculated table if we have statistical significance)

Likelihood-Ratio Test

- Likelihood-Ratio Dunning (1993)
- Examples from his paper (k means count, and ~ means “not”)

Bigrams Ranked by Log-Likelihood Test

$-2\log\lambda$	$k(AB)$	$k(A \sim B)$	$k(\sim AB)$	$k(\sim A \sim B)$	A	B
270.72	110	2442	111	29114	the	swiss
263.90	29	13	123	31612	can	be
256.84	31	23	139	31584	previous	year
167.23	10	0	3	31764	mineral	water
157.21	76	104	2476	29121	at	the
157.03	16	16	51	31694	real	terms
146.80	9	0	5	31763	natural	gas
115.02	16	0	865	30896	owing	to
104.53	10	9	41	31717	health	insurance
100.96	8	2	27	31740	stiff	competition

If $-2\log\lambda \geq 3.841$, there is approx. 95% certainty (sig. value 0.05) that B appears together with A more frequently than without.

d	0.05	0.01	0.001
1	3.841	6.635	10.828
2	5.991	9.210	13.816
3	7.815	11.345	16.266

Chi-Square table for different significance values and d degrees of freedom

We drifted away…

Type-based Recognition:
- Collocations
- Length
- Internal punctuation
- Appearances without dot

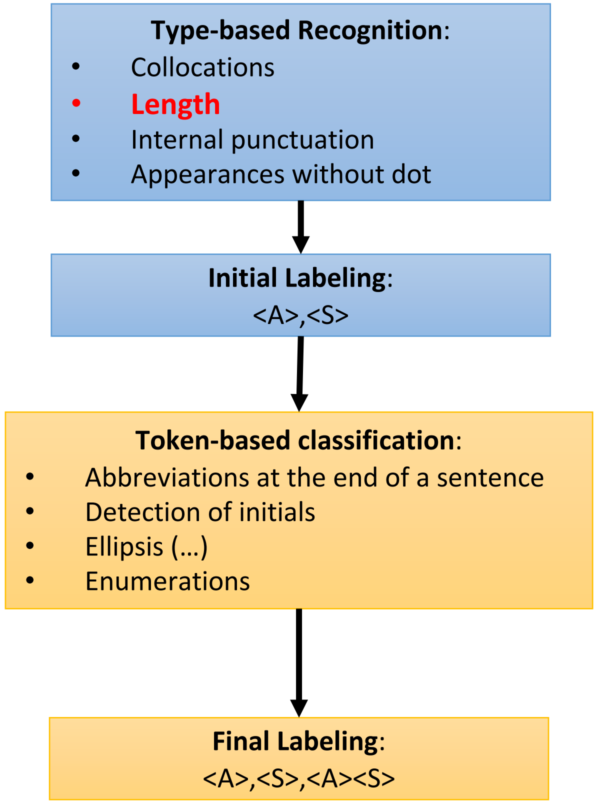
Initial Labeling:
<A>,<S>

Token-based classification:
- Abbreviations at the end of a sentence
- Detection of initials
- Ellipsis (…)
- Enumerations

Final Labeling:
<A>,<S>,<A><S>

… but now we know how to calculate $log\lambda(w,\bullet)$

PUNKT-Algorithm



- Abbreviations tend to be short
- They introduced a factor for the length of the token w:

$$F_{length} = \frac{1}{e^{length(w)}}$$

- With length(w) being the amount of characters excluding inner punctuation
- Example:

length(u.s.a)=3

PUNKT-Algorithm

Type-based Recognition:
- Collocations
- Length
- **Internal punctuation**
- **Appearances without dot**

↓

Initial Labeling:
<A>,<S>

↓

Token-based classification:
- Abbreviations at the end of a sentence
- Detection of initials
- Ellipsis (…)
- Enumerations

↓

Final Labeling:
<A>,<S>,<A><S>

- Internal punctuation is a strong hint for an abbreviation (e.g.)

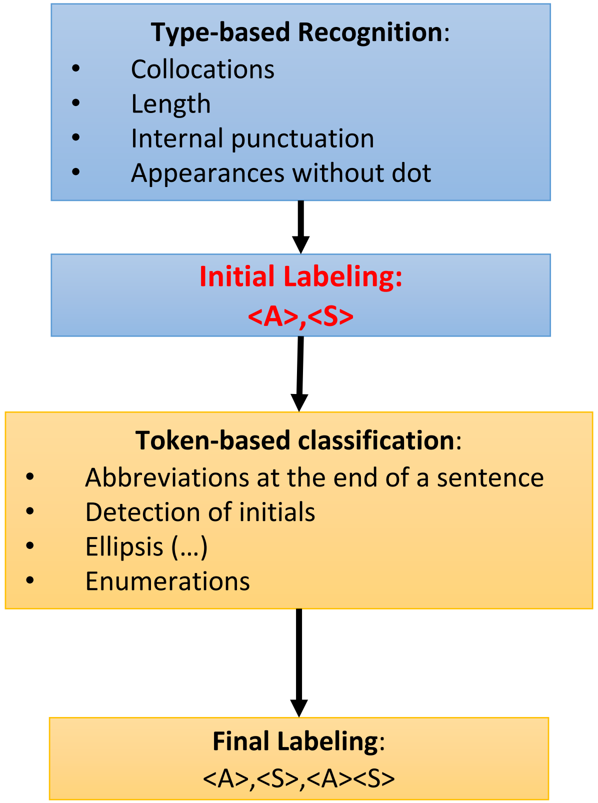
- Another term:

$$F_{periods} = number\ internal\ periods + 1$$

- Abbreviations should (almost never) appear without a dot

$$F_{penalty} = \frac{1}{length(w)^{C(w,\neg\bullet)}}$$

PUNKT-Algorithm – first decision



- The authors argued, that:

$If:$

$$\text{Scaled}\log\lambda = \log\lambda(w,\bullet)\cdot F_{length}(w)\cdot F_{periods}(w)\cdot F_{penalty}(w) \geq 0.3$$

➔ w is an abbreviation (else it is not)

PUNKT-Algorithm – first decision

Type-based Recognition:
- Collocations
- Length
- Internal punctuation
- Appearances without dot

↓

Initial Labeling:
<A>,<S>

↓

Token-based classification:
- Abbreviations at the end of a sentence
- Detection of initials
- Ellipsis (…)
- Enumerations

↓

Final Labeling:
<A>,<S>,<A><S>

- Examples for „Scaled Likelihood“

Final sorting	Scaled log λ
n.h	7.60
a.g	6.08
m.j	4.56
u.n	4.56
u.s.a	4.19
ga	3.04
vt	3.04
ore	0.32
reps	0.31
mo	0.30
1990s	0.26
ounces	0.06
alex	0.03
depositor	0.00

PUNKT-Algorithm – first decision

Type-based Recognition:

- Collocations
- Length
- Internal punctuation
- Appearances without dot

Initial Labeling:
<A>,<S>

Token-based classification:

- **Abbreviations at the end of a sentence**
- Detection of initials
- Ellipsis (…)
- Enumerations

Final Labeling:
<A>,<S>,<A><S>

- Now we have the chance to correct some individual instances
- For all abbreviations (after stage 1), check these heuristics:

  1. Orthographic heuristic (does a token appear both upper- and lowercase)
  2. Frequent Starter heuristic (does a token start a sentence significantly often)
  3. Collocational bond heuristic (does the token before the dot form a significant connection to the token after the dot)

- If any one heuristic matches, then we modify the previous decision

"Punkt"- Unsupervised Multilingual Sentence boundary Detection

- Evaluation on 11 languages :

Results of classification—newspaper corpora (mixed case).

Corpus	Error (<S>) (%)	Prec. (<S>) (%)	Recall (<S>) (%)	F (<S>) (%)	Error (<A>) (%)	Prec. (<A>) (%)	Recall (<A>) (%)	F (<A>) (%)
B. Port.	1.11	99.14	99.72	99.43	0.99	96.88	70.89	81.87
Dutch	0.97	99.25	99.72	99.48	0.66	99.31	90.24	94.55
English	1.65	99.13	98.64	98.89	0.71	99.86	97.52	98.68
Estonian	2.12	98.58	99.07	98.83	1.75	98.22	83.51	90.27
French	1.54	99.31	99.08	99.19	0.72	95.19	79.20	86.46
German	0.35	99.69	99.93	99.81	0.26	99.91	97.34	98.61
Italian	1.13	99.32	99.49	99.41	0.74	96.60	83.48	89.56
Norw.	0.81	99.45	99.68	99.56	0.72	98.16	90.81	94.34
Spanish	1.06	99.66	99.23	99.45	0.35	98.70	93.33	95.94
Swedish	1.76	98.82	99.36	99.09	1.48	94.10	66.32	77.80
Turkish	1.31	99.40	99.24	99.32	0.43	95.35	89.13	92.13
Mean	1.26	99.25	99.38	99.31	0.80	97.48	85.62	90.93
SD	0.49	0.33	0.38	0.29	0.46	1.99	10.19	6.69

Sentence Splitting using Machine Learning

- Instance: For every ambiguous character (“.”), we decide:
  - End of sentence
  - Abbreviation
  - Enumeration
  - …

➔ Multilabel classification

- Requires a labelled corpus
- Can make use of standard classifiers

Sentence Splitting using Machine Learning

- Features (Riley 1989):
  - Probability of a token at the end of a sentence
  - Probability of a token at the start of a sentence
  - Length of the token (in characters)
  - Length of the next token
  - Orthographic features (Uppercase, Lowercase, Full Capitalized, Number)
  - Orthographic features of the next token
  - Character after the dot (or null)
  - Equivalence class of an abbreviation according to a (e.g. month, unit, title, address, name)

➔ Error rate of 0.2% on the Brown Corpus!