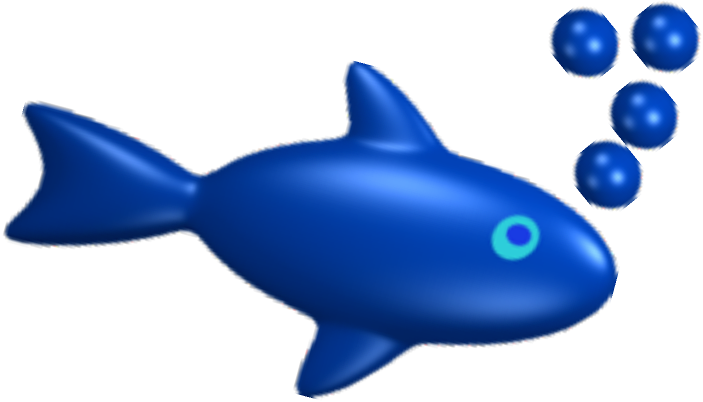


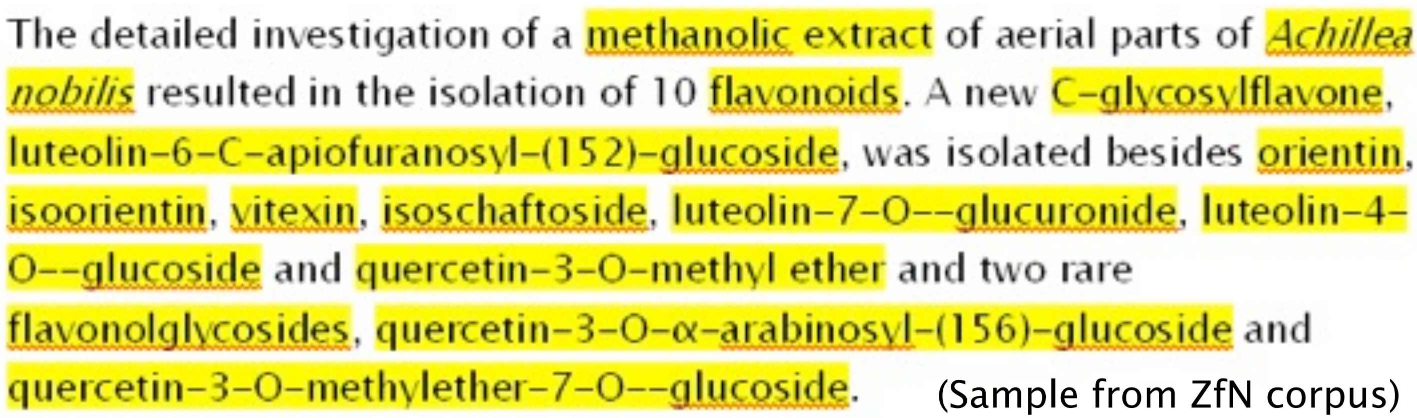
Bootstrapping Noun Groups & Technical Terms

Using Closed-Class Elements Only

Kathrin Eichler and Günter Neumann
LT-Lab, DFKI, Berlin & Saarbrücken

Goal

- Extract and categorize technical terms (TTs), e.g., from scientific texts



The detailed investigation of a methanolic extract of aerial parts of *Achillea nobilis* resulted in the isolation of 10 flavonoids. A new C–glycosylflavone, luteolin–6–C–apiofuranosyl–(152)–glucoside, was isolated besides orientin, isoorientin, vitexin, isoschaftoside, luteolin–7–O––glucuronide, luteolin–4–O––glucoside and quercetin–3–O–methyl ether and two rare flavonolglycosides, quercetin–3–O–α–arabinosyl–(156)–glucoside and quercetin–3–O–methylether–7–O––glucoside.

(Sample from ZfN corpus)

- Extraction method should be…
  - Multilingual
  - Domain–adaptive
  - Nearly unsupervised

Related Tasks

	NE / GN Recognition	Keyword Extraction	TT Extraction
WHAT	PER, LOC, ORG / domain-specific expressions	Small set of important concepts	All technical terms
HOW	(weakly) supervised; lexico-syntactic patterns[1]	Term frequencies; Wikipedia[2]	NG chunking + web-statistics
CATEGORIZATION	yes	no	yes

[1] Etzioni et al., 2005

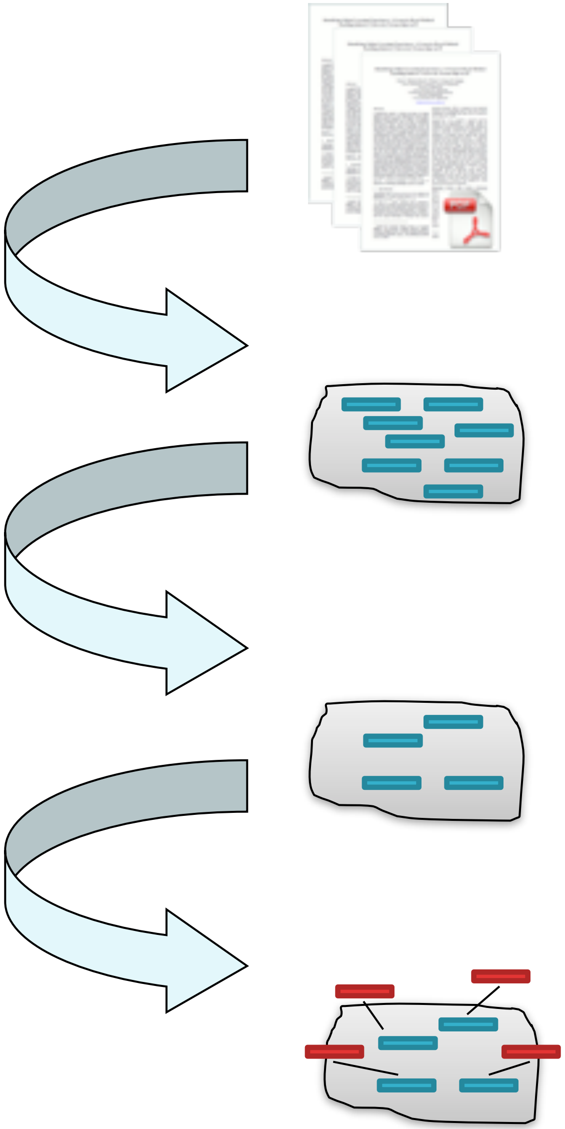
[2] Mihalcea and Csomai, 2007

Our Approach

1. Candidate extraction

5. TT Filtering

8. TT Categorization

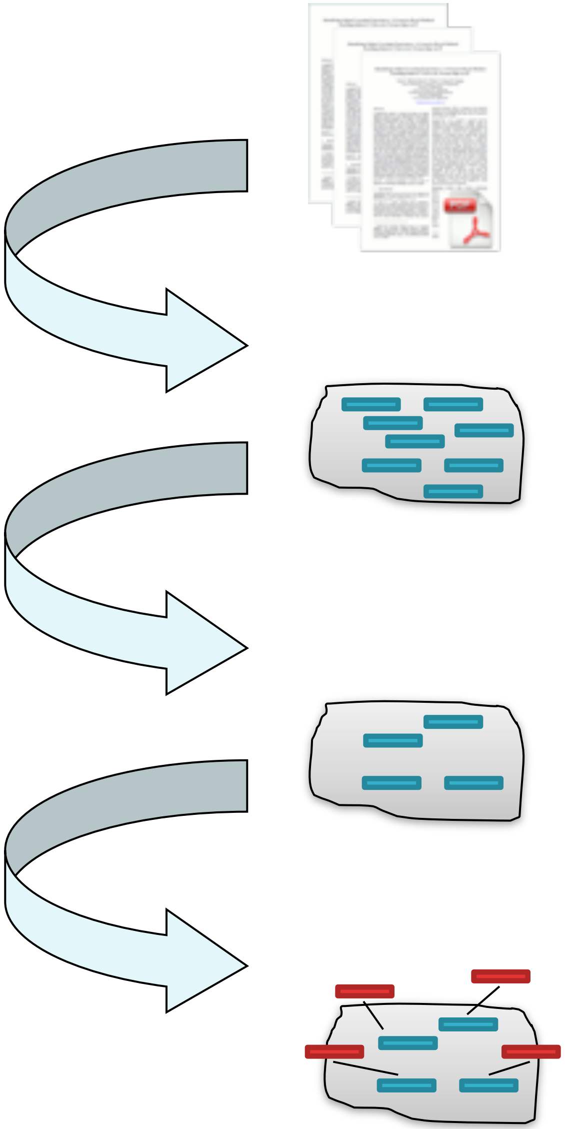


Our Approach

1. Candidate extraction

5. TT Filtering

8. TT Categorization



ccChunk – version 1

Evans and Pourcel (2009): „Lexical concepts associated with the *grammatical subsystem* (…) provide a scaffolding (…) across which the rich content associated with lexical concepts of the *lexical subsystem* can be draped.“

- Determine left/right boundaries of nominal groups using simple context patterns based on
  - Closed-class element lists (i.e. grammatical subsystem)
  - Supervised word class models for V and ADV

	Mary	enjoys	compositions	by	Mozart	.
BOS	Mary	*V*	compositions	*PREP*	Mozart	*PUNCT*

(Dis-)Advantages of the old ccChunk

- Advantages
  - No POS-tagger, no chunk rules, only finite list of closed-class elements
  - Domain-independent: Closed-class elements are the same in all domains
  - Can be adapted to other languages with little effort
  - Scales well to large amounts of textual data
- Main disadvantage: V / ADV models
  - Trained on annotated data → domain-dependent
  - Classify words without using context information
    - cannot deal with word class ambiguities, e.g. V/N ambiguity in „structures", „types", „books", „flies",…

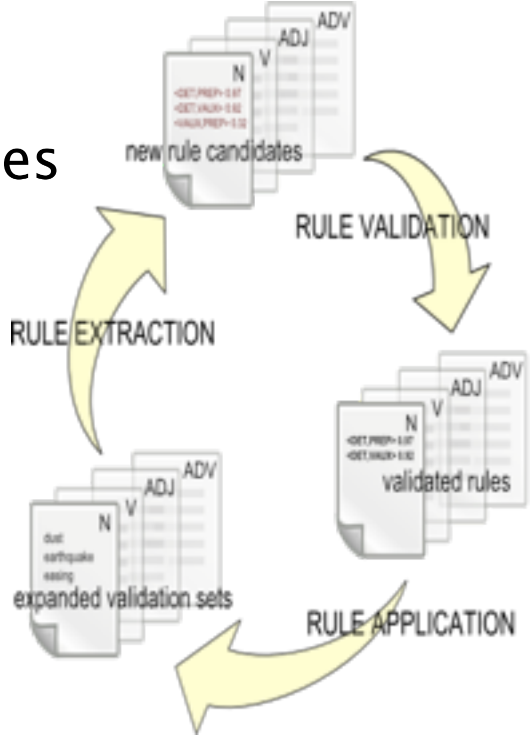
ccChunk version 2 – self domain-adaptive

- Replace V/ADV models by set of context patterns bootstrapped from an unannotated input text using open-class elements
- General idea:
  - Use simple context seed rules to extract validation sets for each open-class type (N, V, ADJ, ADV)
  - Different seeds stand in competition and are later used for disambiguation
  - Apply bootstrapping to iteratively expand set of extraction rules and validation sets

Bootstrapping algorithm

As basis for extracting competing patterns usable for NEGATIVE examples

- INITIALIZATION
  - Use one seed context rule for each OCW class (N, V, ADJ, ADV) to extract initial validation sets
- BOOTSTRAPPING LOOP
  - Step 1: Extract and validate rule candidates based on validation sets
  - Step 2: Expand validation sets based on validated rules
- POSTPROCESSING
  - Disambiguate ambiguous tokens using validated rules



Seed context rules for OCW classes

- Nouns:
  - <DET X PREP>, where X is a single non-CCW token

the computation of

- Verbs:
  - <TO X DET>, where X is a single non-CCW token
    - „to give the“

to give the

- Adjectives:
  - <BE GRAD_ADV X>, where
    - BE is some form of the auxiliary be
    - GRAD_ADV is some grading adverb (e.g. very)
  - „is very proud“

is very proud

- Adverbs:
  - Each seed ADJ-ly that appears in the text

proudly

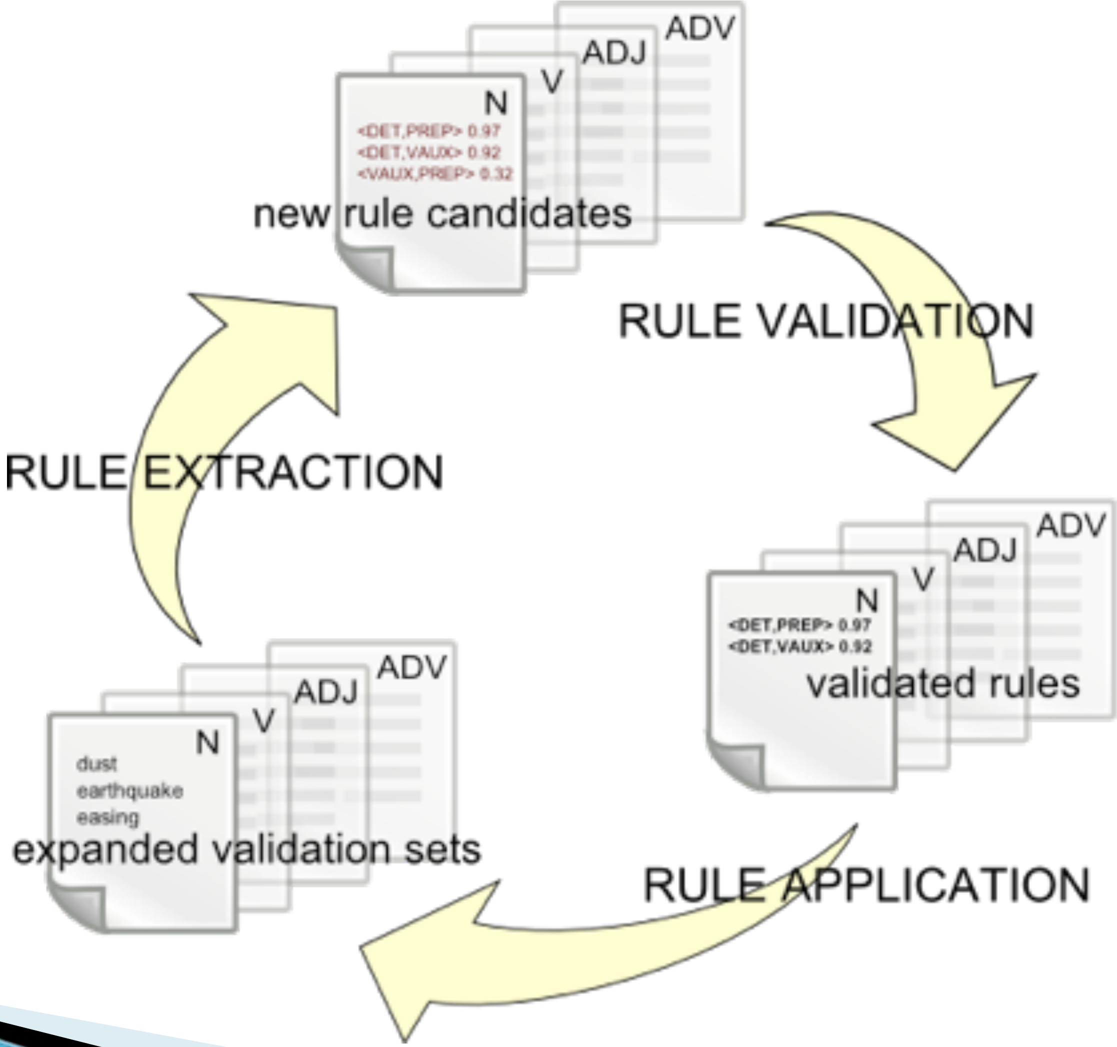
Initial validation sets

- Input text: Wall Street Journal training corpus used for CONLL 2000 shared task on chunking
  - 8,936 sentences
  - 46,874 NP chunks
- Extracted based on seed context rules:

Nouns	1222
Verbs	535
Adjectives	31
Adverbs	9

- Example:
  - Seed context rule <DET X PREP> for nouns extracts „airport“ from „Getting to and from *the* airport *in* coming weeks may be the problem“.

Bootstrapping loop



Rule candidate extraction

- For each entry X in validation set of OCW type O, match all
  <LC X RC> in the text, where
  - LC: left context, i.e. some tagged token
  - RC: right context, i.e. some tagged token
- Add <LC, RC> to set of rule candidates for O
- Example:
  - For entry „airport“ from the validation set for nouns, we can extract the noun rule candidate <DET, VAUX> from
    While the *airport* was closed , flights were diverted …

Rule validation

- Calculate accuracy of rule r for OCW type O:

$$acc(r) = \frac{pos_r + 1}{pos_r + neg_r + 1}$$

- Where
  - pos_r: # of occurrences matching $<LC_r\ O\ RC_r>$
  - neg_r: # of occurrences matching $<LC_r\ \neg O\ RC_r>$

- If $acc(rule_n)$ > threshold (currently set to 0.5)

  → Add $rule_n$ to set of validated rules
- Example: acc(<DET, VAUX>) = 0.92

Validation set expansion

- Apply all validated rules to text to extract additional entries for validation sets
- Example:
  - Applying the validated rule <DET, VAUX> extracts noun „units“ from

    These units *were* handling calls both from people in the San Francisco area and from computers themselves.
  - „units“ is added it to the validation set for nouns and used to validate rules in the next iteration

Preliminary results

- Chunk-/ Token*-based evaluation

	Precision	Recall	F1
Baseline	0.51 / 0.67	0.66 / **0.96**	0.58 / 0.79
Initial seed tagging	0.55 / 0.70	0.69 / 0.95	0.61 / 0.80
Final tagging	**0.60** / **0.75**	**0.69** / 0.91	**0.64** / **0.82**

- Baseline: all non-CCWs tagged as noun
- Bootstrapping process (slightly) improves results, in particular precision → problem: false positives

*Token-based means: count on level of BIO elements as in CoNLL evaluation.

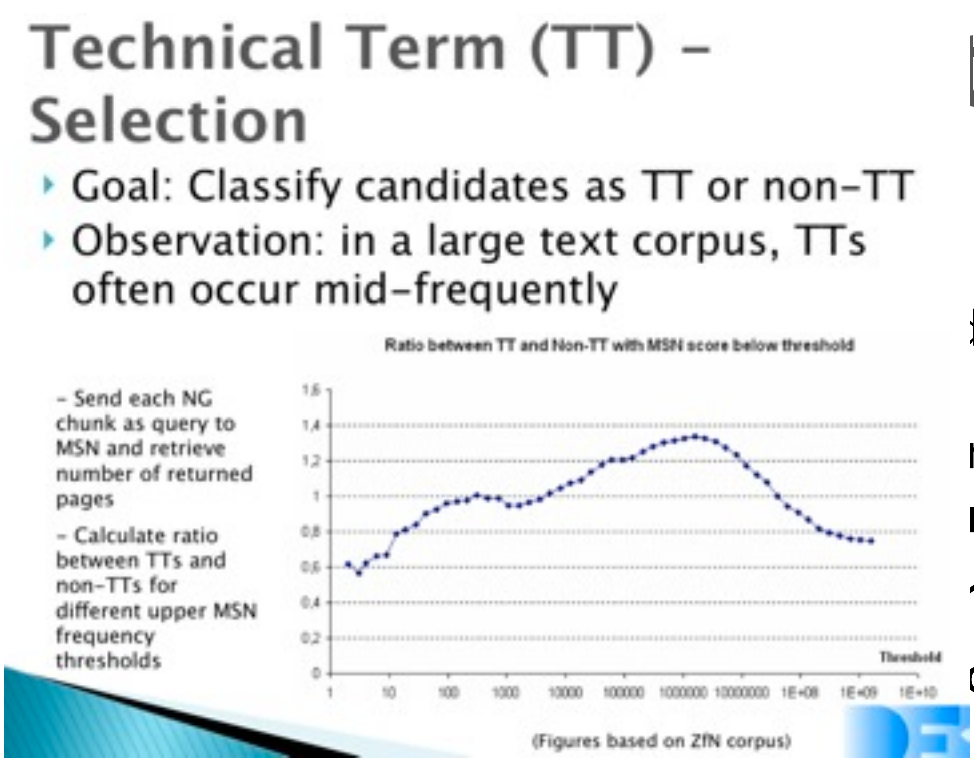
Summary: ccChunk version 2

- Advantages
  - Domain-adaptive: No lexical information used as input
  - Advantage over version 1: V / ADV models replaced by automatically generated lists
- Disadvantages
  - Rules need to be relearned for each new input text, even though they are non-lexical, i.e. domain-independent
  - Like version 1, words are tagged without using context information

    → cannot deal with word class ambiguities, e.g. V/N ambiguity in structures, types, books, flies,…

Additional & Future work

- Improve ccChunk
  - Use more sophisticated rule validation method, e.g. EM-based confidence estimation
  - Testing on more different domains
  - Evaluate how the size of the input text affects the results
- Ranking of technical terms
  - Exploring search engine frequencies (MSN) & SVM^{rank} algorithm (DFKI system KeyWE, SemEval 2010)
- Named Entity Extraction
  - Learning of specialized context patterns for extracting protein names in biomedical names (Project Dilia)
    - NG: enriched polymerase chain reaction amplification
    - Protein name: polymerase
    - left context: enriched
    - right contex: chain reaction amplification



Technical Term (TT) – Selection

- Goal: Classify candidates as TT or non-TT
- Observation: in a large text corpus, TTs often occur mid-frequently

- Send each NG chunk as query to MSN and retrieve number of returned pages
- Calculate ratio between TTs and non-TTs for different upper MSN frequency thresholds

(Figures based on ZfN corpus)

ture work

dation method, e.g. EM-based

ns

ut text affects the results

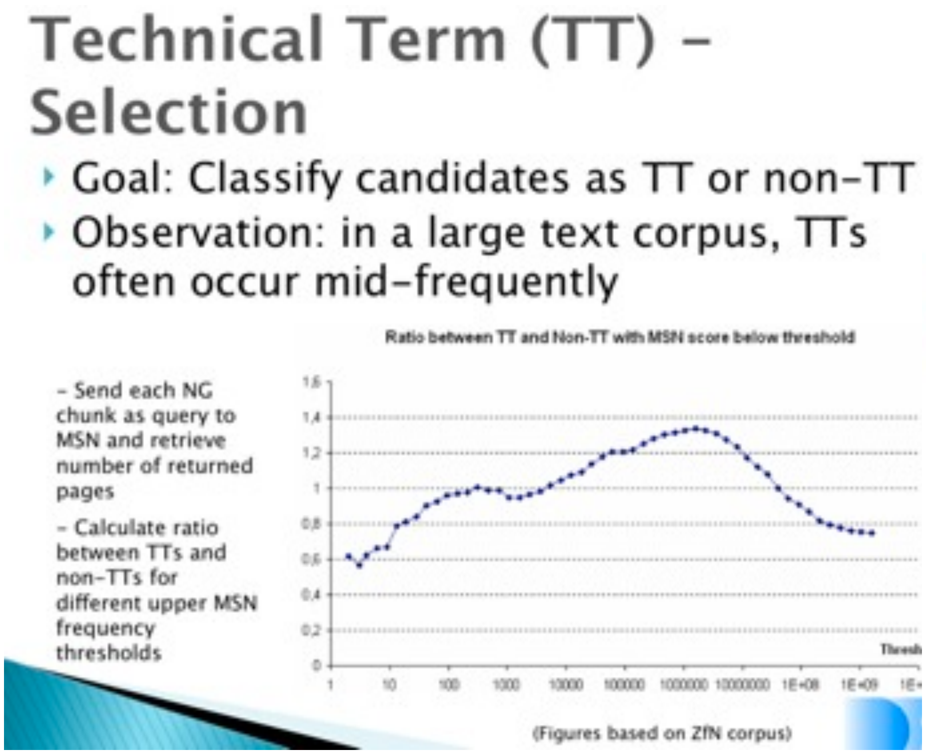
rms

cies (MSN) & SVM^{rank} algorithm

10)

- Named Entity Extraction
  - Learning of specialized context patterns for extracting protein names in biomedical names (Project Dilia)
    - NG: enriched polymerase chain reaction amplification
    - Protein name: polymerase
    - left context: enriched
    - right contex: chain reaction amplification

ture work

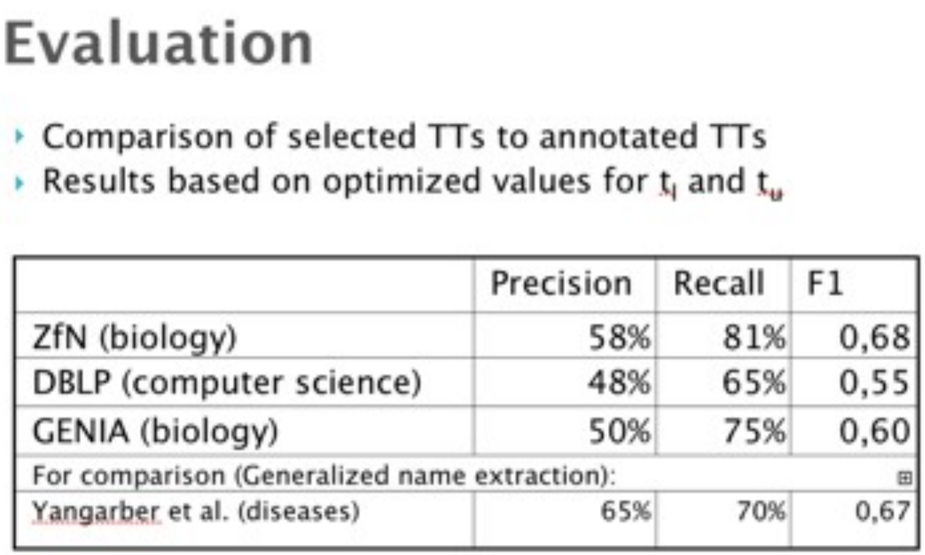


Technical Term (TT) – Selection

- Goal: Classify candidates as TT or non-TT
- Observation: in a large text corpus, TTs often occur mid-frequently

- Send each NG chunk as query to MSN and retrieve number of returned pages
- Calculate ratio between TTs and non-TTs for different upper MSN frequency thresholds

(Figures based on ZfN corpus)



Evaluation

- Comparison of selected TTs to annotated TTs
- Results based on optimized values for t_l and t_u

	Precision	Recall	F1
ZfN (biology)	58%	81%	0,68
DBLP (computer science)	48%	65%	0,55
GENIA (biology)	50%	75%	0,60
For comparison (Generalized name extraction):			
Yangarber et al. (diseases)	65%	70%	0,67

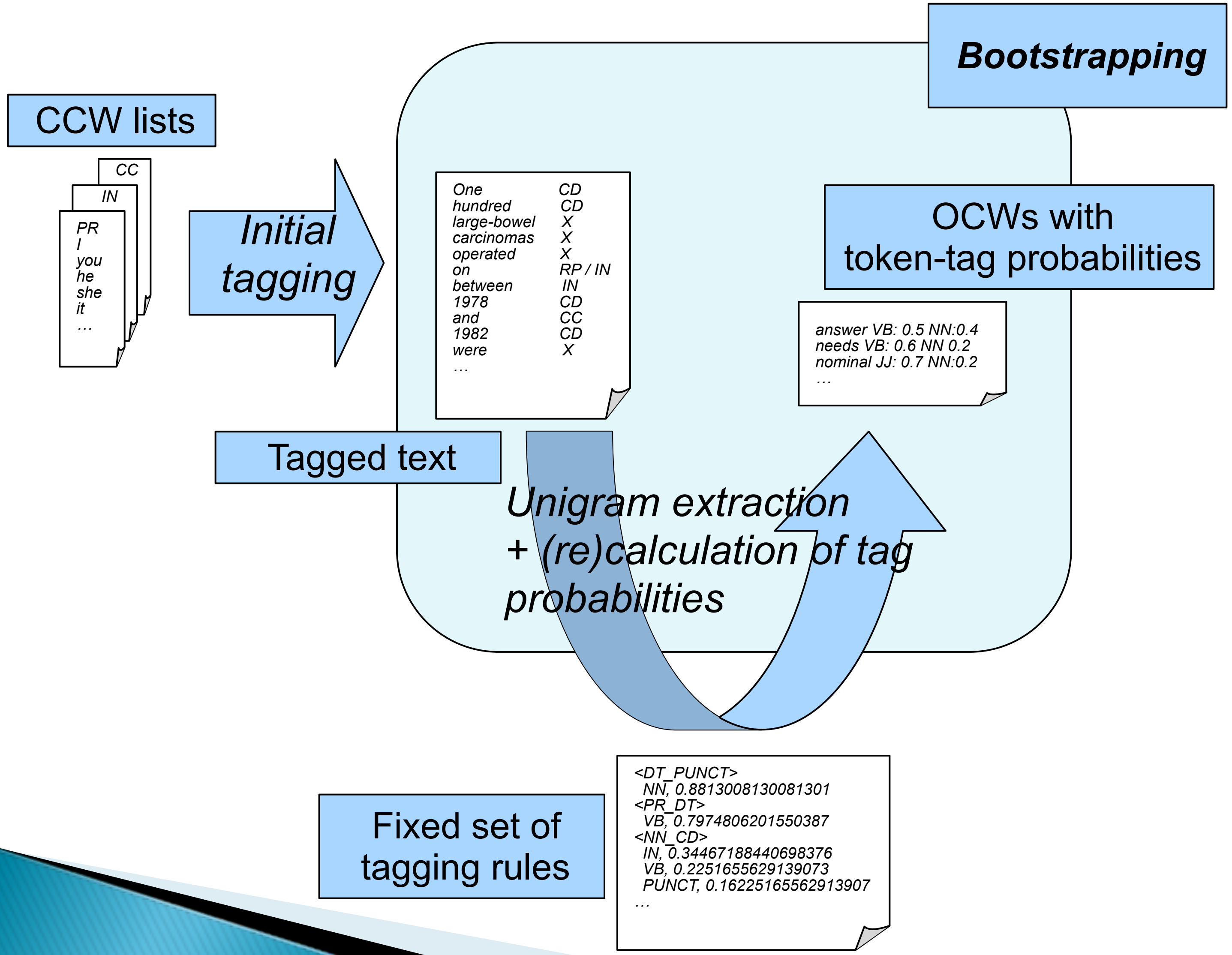
- Named Entity Extraction
  - Learning of specialized context patterns for extracting protein names in biomedical names (Project Dilia)
    - NG: enriched polymerase chain reaction amplification
    - Protein name: polymerase
    - left context: enriched
    - right contex: chain reaction amplification

ccChunk – version 3

- Input
  1. Lists of closed class words
  2. Non-lexical tagging rules (with probabilities) extracted from some annotated text
     - DET _ PUNCT → NN (0.91) … the program .
     - TO _ DET → VB (0.98) … to avoid the …

- Domain-adaptive, like version 2: Open class words are tagged using a bootstrapping approach

Bootstrapping



CCW lists

CC

IN

PR
I
you
he
she
it
…

Initial tagging

One	*CD*
hundred	*CD*
large-bowel	*X*
carcinomas	*X*
operated	*X*
on	*RP / IN*
between	*IN*
1978	*CD*
and	*CC*
1982	*CD*
were	*X*
…	

Tagged text

Bootstrapping

OCWs with token-tag probabilities

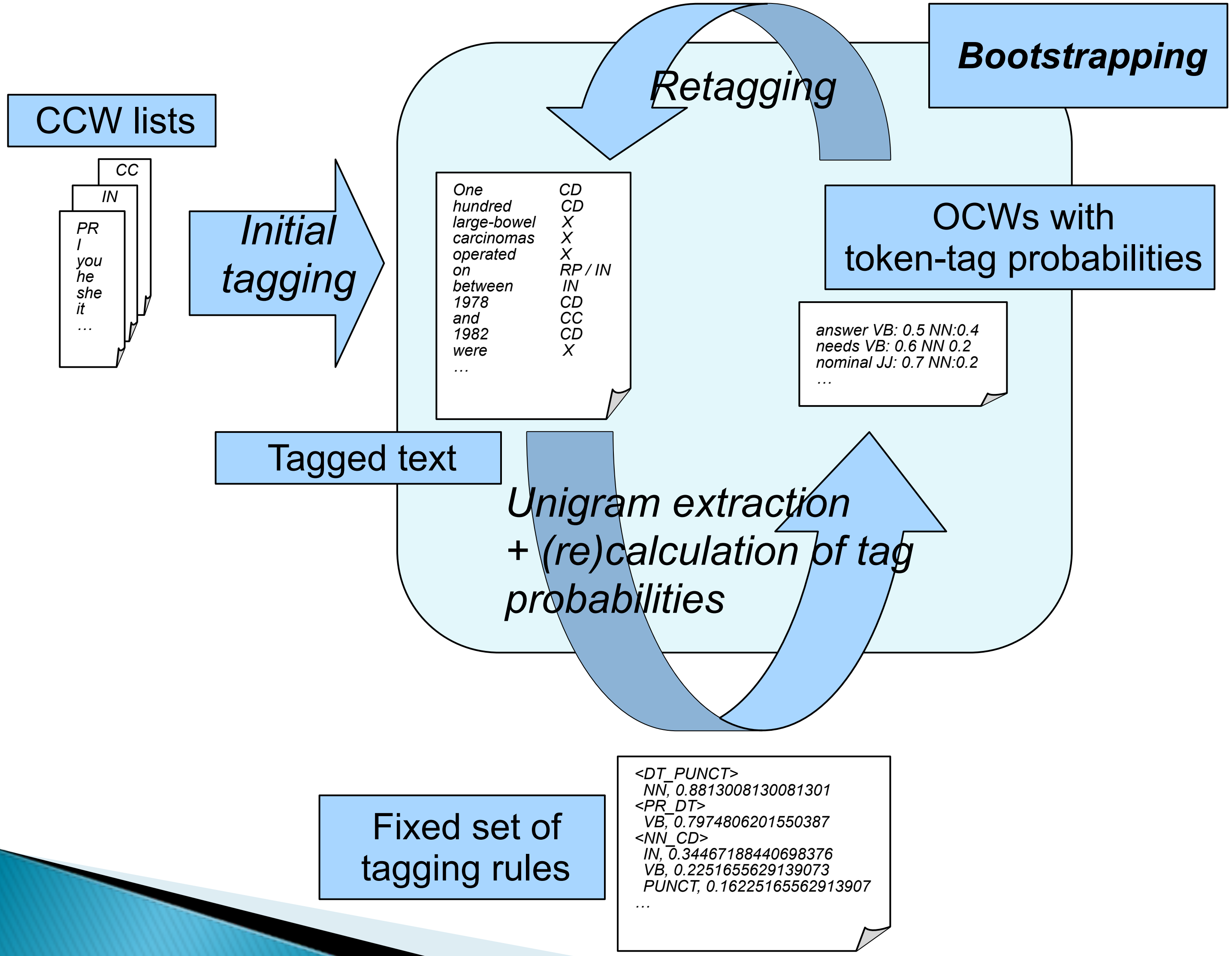
answer VB: 0.5 NN:0.4
needs VB: 0.6 NN 0.2
nominal JJ: 0.7 NN:0.2
…

Unigram extraction + (re)calculation of tag probabilities

Fixed set of tagging rules

```
<DT_PUNCT>
 NN, 0.8813008130081301
<PR_DT>
 VB, 0.7974806201550387
<NN_CD>
 IN, 0.344671884406983 76
 VB, 0.2251655629139073
 PUNCT, 0.16225165562913907
…
```

Bootstrapping



CCW lists

CC

IN

PR
I
you
he
she
it
…

Initial tagging

Tagged text

One	*CD*
hundred	*CD*
large-bowel	*X*
carcinomas	*X*
operated	*X*
on	*RP / IN*
between	*IN*
1978	*CD*
and	*CC*
1982	*CD*
were	*X*
…	

Retagging

Bootstrapping

OCWs with token-tag probabilities

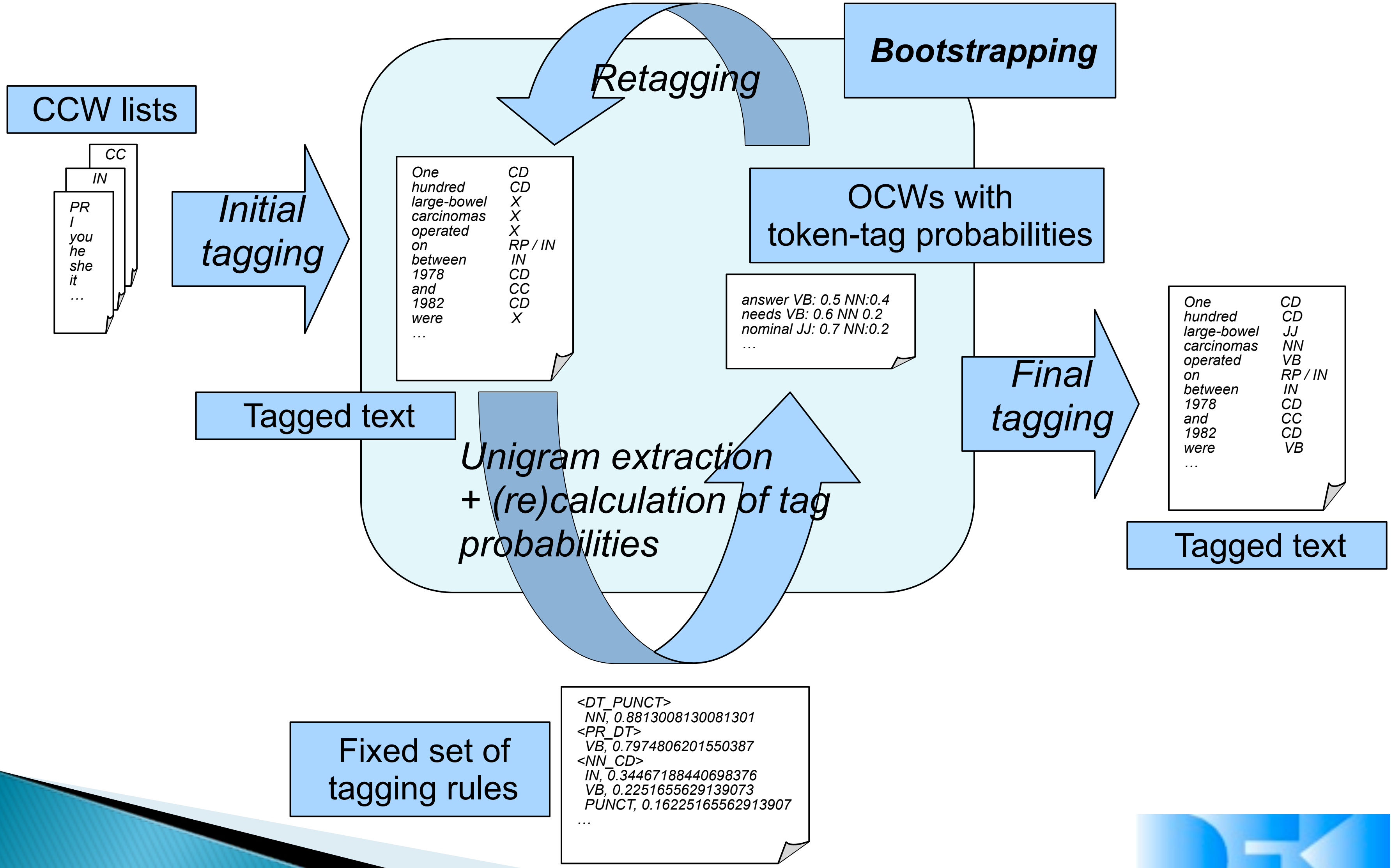
answer VB: 0.5 NN:0.4
needs VB: 0.6 NN 0.2
nominal JJ: 0.7 NN:0.2
…

Unigram extraction + (re)calculation of tag probabilities

Fixed set of tagging rules

<DT_PUNCT>
NN, 0.8813008130081301
<PR_DT>
VB, 0.7974806201550387
<NN_CD>
IN, 0.34467188440698376
VB, 0.2251655629139073
PUNCT, 0.16225165562913907
…

Bootstrapping



Bootstrapping

CCW lists

CC

IN

PR
I
you
he
she
it
…

Initial tagging

One	CD
hundred	CD
large-bowel	X
carcinomas	X
operated	X
on	RP / IN
between	IN
1978	CD
and	CC
1982	CD
were	X
…	

Tagged text

Retagging

OCWs with token-tag probabilities

answer VB: 0.5 NN:0.4
needs VB: 0.6 NN 0.2
nominal JJ: 0.7 NN:0.2
…

Unigram extraction
+ (re)calculation of tag probabilities

Fixed set of tagging rules

```
<DT_PUNCT>
 NN, 0.8813008130081301
<PR_DT>
 VB, 0.7974806201550387
<NN_CD>
 IN, 0.34467188440698376
 VB, 0.2251655629139073
 PUNCT, 0.16225165562913907
…
```

Final tagging

One	CD
hundred	CD
large-bowel	JJ
carcinomas	NN
operated	VB
on	RP / IN
between	IN
1978	CD
and	CC
1982	CD
were	VB
…	

Tagged text

Bootstrapping – Example

Bootstrapping – Example

- Initial tagging: Tagging of all Closed Class (CC) tokens
- Unigram extraction:
  - … c-Jun N-terminal kinase, which phosphorylates and …

…	c-Jun	N-terminal	kinase,	which	phosphorylates	and	…
	?	?	?	WD	?	CC	

  - … kinase that phosphorylates the transactivation domain …

…	kinase	that	phosphorylates	the	transactivation	domain	…
	?	DT/WD	?	DT	?	?	

- Rule-tag probabilities (tagging rules)
  - <WD_CC> → VB: 0.54, NN: 0.31
  - <DT_DT> → IN: 0.38, NN: 0.29, VB: 0.26
  - <WD_DT> → VB: 0.97
- Token-tag probabilities after the first iteration:
  - phosphorylates VB:0.72, NN:0.23
- Unigram extraction
  - … Fos kinase phosphorylates c-Fos at a site near …

Bootstrapping – Example

- Initial tagging: Tagging of all Closed Class (CC) tokens
- Unigram extraction:
  - … c-Jun N-terminal kinase, which phosphorylates and …

…	c-Jun	N-terminal	kinase,	which	phosphorylates	and	…
	?	?	?	WD	?	CC	

  - … kinase that phosphorylates the transactivation domain …

…	kinase	that	phosphorylates	the	transactivation	domain	…
	?	DT/WD	?	DT	?	?	

- Rule-tag probabilities (tagging rules)
  - <WD_CC> → VB: 0.54, NN: 0.31
  - <DT_DT> → IN: 0.38, NN: 0.29, VB: 0.26
  - <WD_DT> → VB: 0.97
- Token-tag probabilities after the first iteration:
  - phosphorylates VB:0.72, NN:0.23
- Unigram extraction
  - … Fos kinase phosphorylates c-Fos at a site near …

…	Fos	kinase	phosphorylates	c-Fos	at	a	site	near	…
			VB	?	IN				

Final tagging

- Tokens are tagged based on
  - token-tag probabilities
  - rule-tag probabilities
- The best tag $t_{max}(x)$ for token x is calculated as follows:

$$\mathbf{t_{max}(x) = arg\ max_{t \in T}(\alpha * P_{token}(t,x) + \beta * P_{rule}(t,x))}$$

- Where:
  - T: set of possible OCW tags
  - $P_{token}(t,x)$: token-tag probability of tag t for token x
  - $P_{rule}(t,x)$: rule-tag probability of tag t in the context of x
  - α, β: weights specifying the contribution of token-tag / rule-tag probabilities to the final score ($\alpha + \beta = 1$)

Evaluation

- Based on the final tagging, adjective noun sequences (JJ*NN+) are extracted as NGs
- (Chunk-based) evaluation on English PennTB dataset used in CONLL 2007 (WSJ, sections 2-11)

	P	R	F1
Baseline*	0.51	0.66	0.58
version 2*	0.60	0.69	0.64
version 3	**0.69**	**0.72**	**0.70**

*) Baseline (all non-CCWs tagged as noun) and version 2 results are based on WSJ, sections 15-18

DiLiA
Digital Library Assistent

Experiments to identify and extract protein names from text
(including preliminary evaluation results)

Goal

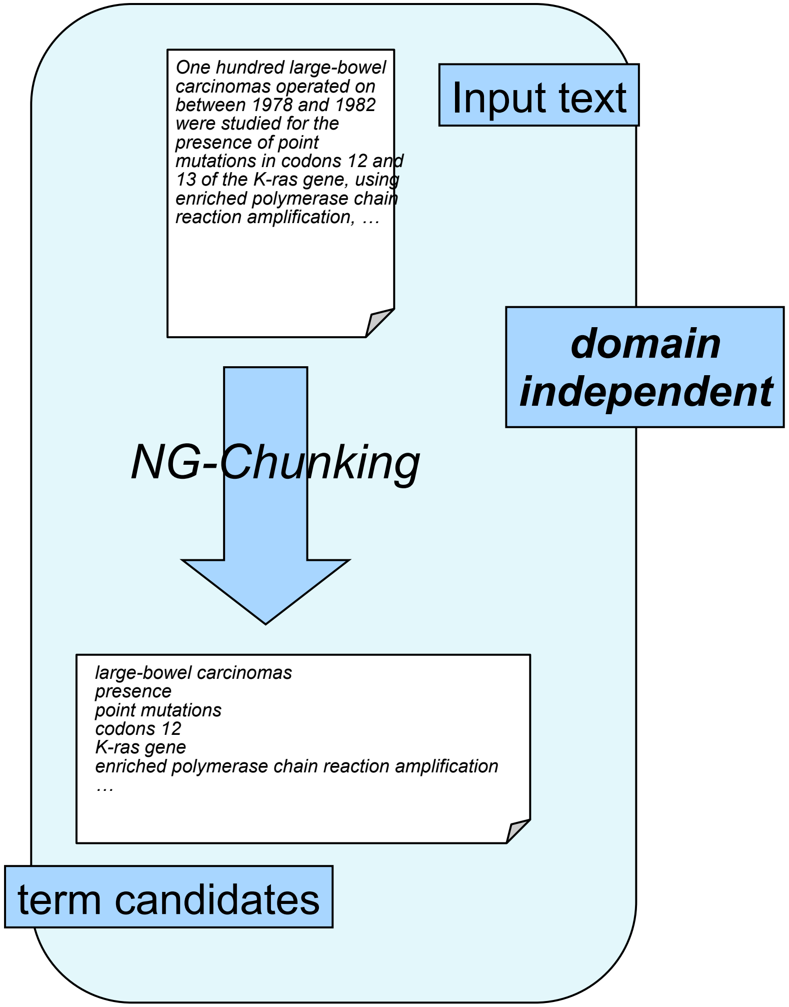
Goal: Identification of protein names in biomedical texts

We studied 52 neuroblastic tumors to test whether the cell death-related proteases, interleukin-1 beta converting enzyme (ICE), CPP32, and Ich-1, were involved in the regression of the tumors.

- Protein names:
  - proteases
  - interleukin-1 beta converting enzyme
  - ICE
  - CPP32
  - Ich-1

→ Identification by using context information:

No mutations resulting in truncation of the APC protein were found.



Input text

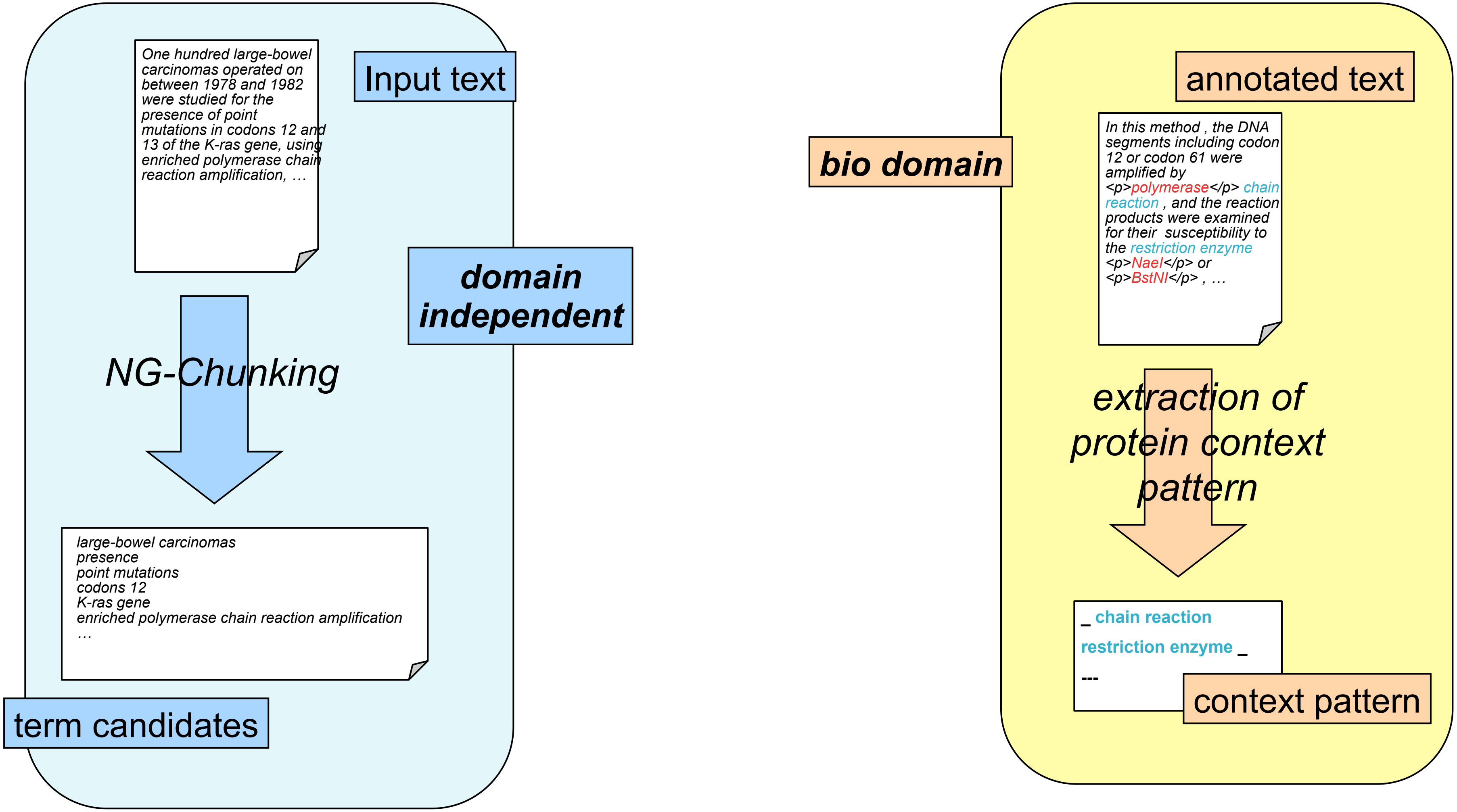
One hundred large-bowel carcinomas operated on between 1978 and 1982 were studied for the presence of point mutations in codons 12 and 13 of the K-ras gene, using enriched polymerase chain reaction amplification, …

domain independent

NG-Chunking

large-bowel carcinomas
presence
point mutations
codons 12
K-ras gene
enriched polymerase chain reaction amplification
…

term candidates



Input text

One hundred large-bowel carcinomas operated on between 1978 and 1982 were studied for the presence of point mutations in codons 12 and 13 of the K-ras gene, using enriched polymerase chain reaction amplification, …

domain independent

NG-Chunking

large-bowel carcinomas
presence
point mutations
codons 12
K-ras gene
enriched polymerase chain reaction amplification
…

term candidates

bio domain

annotated text

In this method , the DNA segments including codon 12 or codon 61 were amplified by <p>polymerase</p> chain reaction , and the reaction products were examined for their susceptibility to the restriction enzyme <p>NaeI</p> or <p>BstNI</p> , …

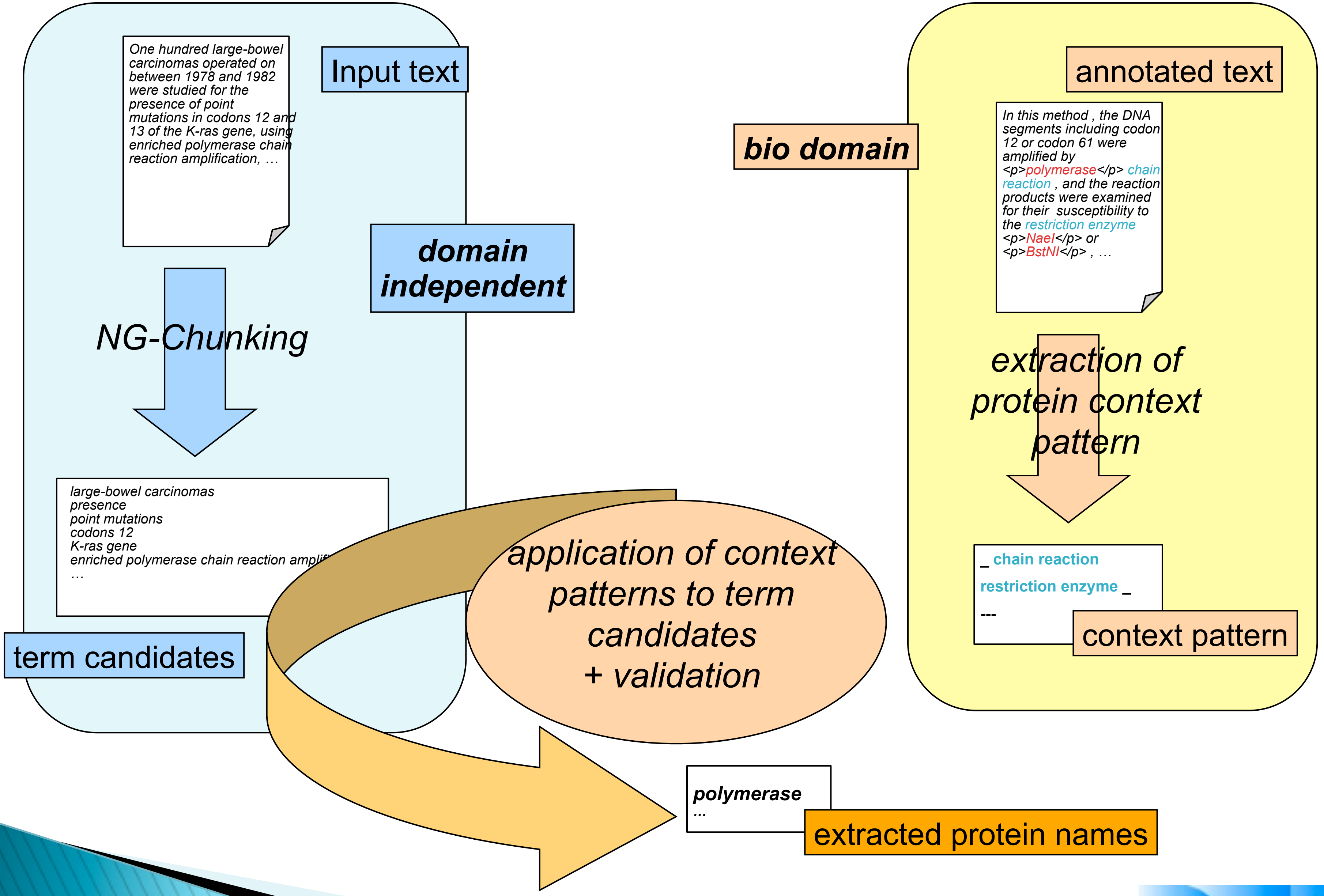
extraction of protein context pattern

_ chain reaction

restriction enzyme _

---

context pattern



Input text

One hundred large-bowel carcinomas operated on between 1978 and 1982 were studied for the presence of point mutations in codons 12 and 13 of the K-ras gene, using enriched polymerase chain reaction amplification, …

domain independent

NG-Chunking

large-bowel carcinomas
presence
point mutations
codons 12
K-ras gene
enriched polymerase chain reaction amplifi
…

term candidates

bio domain

annotated text

In this method , the DNA segments including codon 12 or codon 61 were amplified by <p>polymerase</p> chain reaction , and the reaction products were examined for their susceptibility to the restriction enzyme <p>NaeI</p> or <p>BstNI</p> , …

extraction of protein context pattern

_ chain reaction
restriction enzyme _
---

context pattern

application of context patterns to term candidates + validation

polymerase
…

extracted protein names

ccChunk – Output

- From the final tagged text all adjective-noun-sequences (JJ*NN+) are extracted as NGs

```
enriched polymerase chain reaction amplification
2,3,7,8-tetrachlorodibenzo-p-dioxin
murine
human B lymphocyte immunoglobulin
unknown mechanism
degradation rate
nitroso
surface
viable white blood cells
...
```

Extraction of protein names

- Findings:
  - protein names appear almost always as part of the extracted NGs, but are often nested in longer NGs, e.g.,

NG:	enriched	polymerase	chain reaction amplification
protein name:		polymerase	
left context:	enriched		
right context:			chain reaction amplification

→ Development of a method to detect which parts of the tokens of the extracted NG belong to the context

Extraction of context patterns

- Data: PennBioIE oncology-corpus (biomedical)
  - 1414 annotated PubMed-abstracts (u.a. protein names)
- For all protein names all left and right contexts have been extracted from the corpus text
- For example:

In this method , the DNA segments including codon 12 or codon 61 were amplified by polymerase chain reaction , and the reaction products were examined for their susceptibility to the restriction enzyme NaeI or BstNI , and by dot blot hybridization assay with oligonucleotide probes .

Protein name	left context	right context
polymerase	NIL	_ chain reaction
NaeI	restriction enzyme _	NIL
BstNI	NIL	NIL

Post-processing of the candidates

- Using the extracted context patterns, the candidate list is processed and the patterns are removed from the candidate until no matching patterns can be found.

- For example: extracted NG „enriched polymerase chain reaction amplification“

enriched polymerase chain reaction amplification

context pattern: * amplification

enriched polymerase chain reaction

context pattern: * chain reaction

enriched polymerase

context pattern: enriched *

polymerase

Post-processing of the candidates

- Using the extracted context patterns, the candidate list is processed and the patterns are removed from the candidate until no matching patterns can be found.
- For example: extracted NG „enriched polymerase chain reaction amplification“

enriched polymerase chain reaction amplification

context pattern: * amplification

enriched polymerase chain reaction

context pattern: * chain reaction

enriched polymerase

context pattern: enriched *

<u>polymerase</u>

Post-processing of the candidates

- Using the extracted context patterns, the candidate list is processed and the patterns are removed from the candidate until no matching patterns can be found.

- For example: extracted NG „enriched polymerase chain reaction amplification“

enriched polymerase chain reaction amplification

↓ context pattern: * amplification

enriched polymerase chain reaction

↓ context pattern: * chain reaction

enriched polymerase

context pattern: enriched *

polymerase

Post-processing of the candidates

- Using the extracted context patterns, the candidate list is processed and the patterns are removed from the candidate until no matching patterns can be found.
- For example: extracted NG „enriched polymerase chain reaction amplification“

enriched polymerase chain reaction amplification

↓ context pattern: * amplification

enriched polymerase chain reaction

↓ context pattern: * chain reaction

enriched polymerase

↓ context pattern: enriched *

polymerase

Post-processing of the candidates

- Using the extracted context patterns, the candidate list is processed and the patterns are removed from the candidate until no matching patterns can be found.

- For example: extracted NG „enriched polymerase chain reaction amplification“

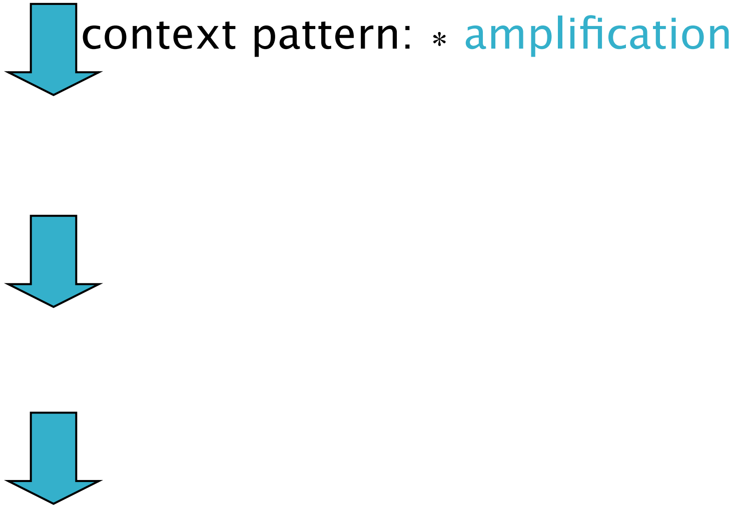
enriched polymerase chain reaction amplification

Post-processing of the candidates

- Using the extracted context patterns, the candidate list is processed and the patterns are removed from the candidate until no matching patterns can be found.

- For example: extracted NG „enriched polymerase chain reaction amplification“

enriched polymerase chain reaction amplification



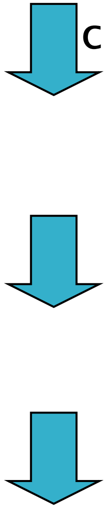
context pattern: * amplification

Post-processing of the candidates

- Using the extracted context patterns, the candidate list is processed and the patterns are removed from the candidate until no matching patterns can be found.

- For example: extracted NG „enriched polymerase chain reaction amplification“

enriched polymerase chain reaction amplification

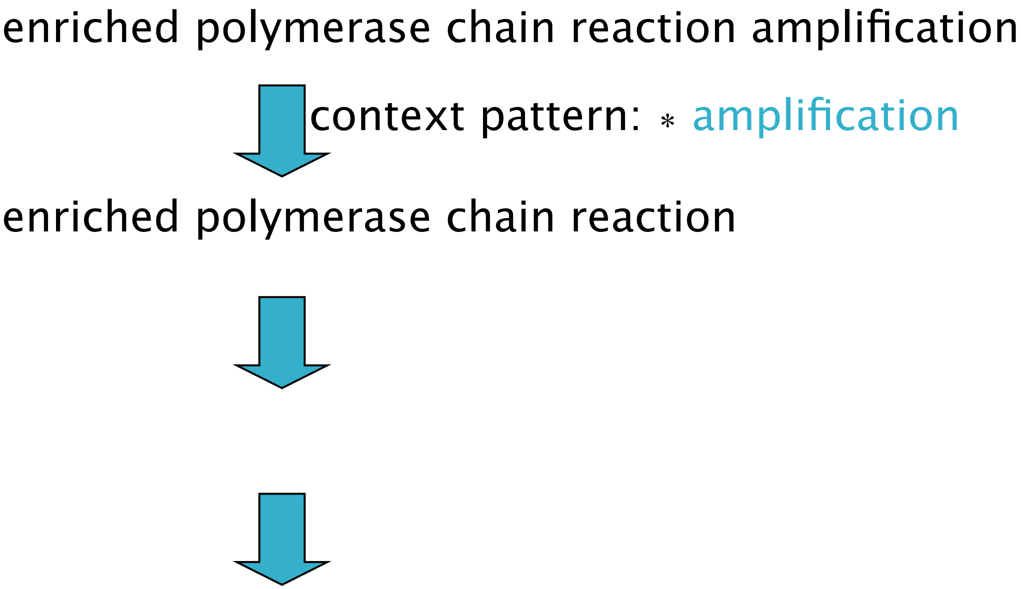


context pattern: * amplification

Post-processing of the candidates

- Using the extracted context patterns, the candidate list is processed and the patterns are removed from the candidate until no matching patterns can be found.

- For example: extracted NG „enriched polymerase chain reaction amplification“



enriched polymerase chain reaction amplification

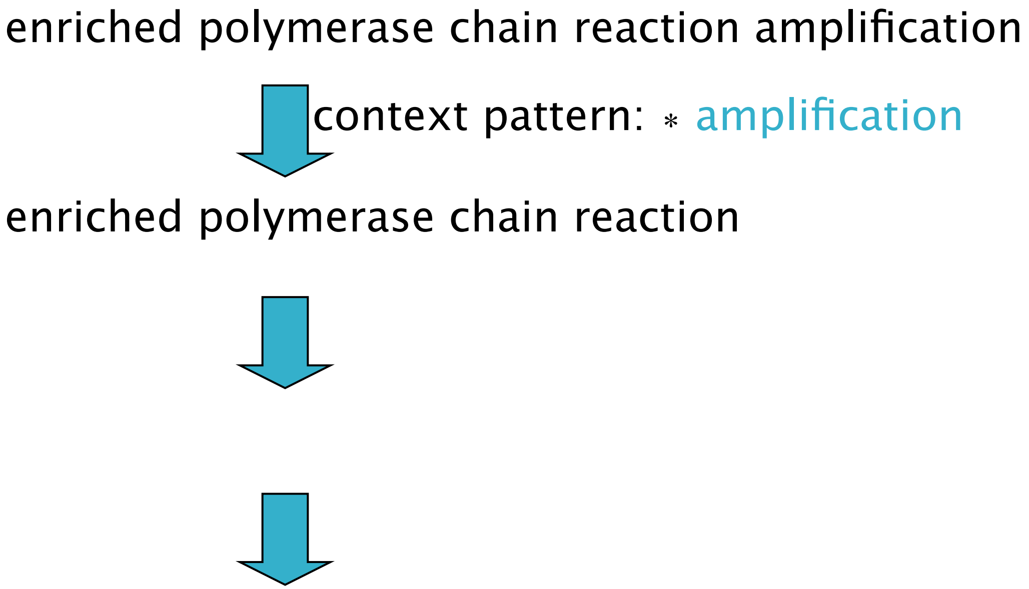
context pattern: * amplification

enriched polymerase chain reaction

Post-processing of the candidates

- Using the extracted context patterns, the candidate list is processed and the patterns are removed from the candidate until no matching patterns can be found.

- For example: extracted NG „enriched polymerase chain reaction amplification“



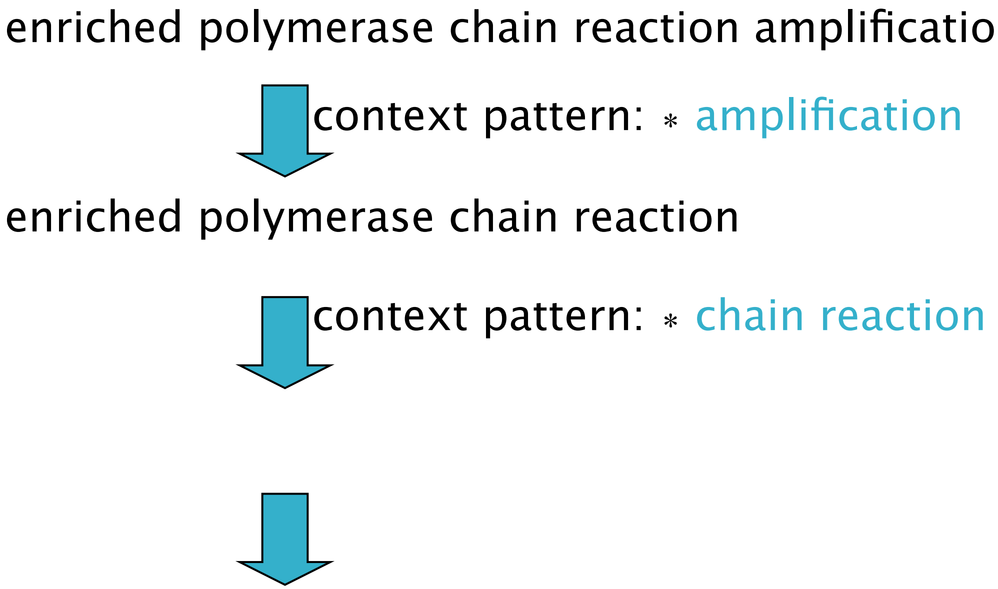
enriched polymerase chain reaction amplification

context pattern: $*$ amplification

enriched polymerase chain reaction

Post-processing of the candidates

- Using the extracted context patterns, the candidate list is processed and the patterns are removed from the candidate until no matching patterns can be found.
- For example: extracted NG „enriched polymerase chain reaction amplification“



enriched polymerase chain reaction amplification

context pattern: * amplification

enriched polymerase chain reaction

context pattern: * chain reaction

Post-processing of the candidates

- Using the extracted context patterns, the candidate list is processed and the patterns are removed from the candidate until no matching patterns can be found.

- For example: extracted NG „enriched polymerase chain reaction amplification“

enriched polymerase chain reaction amplification

context pattern: * amplification

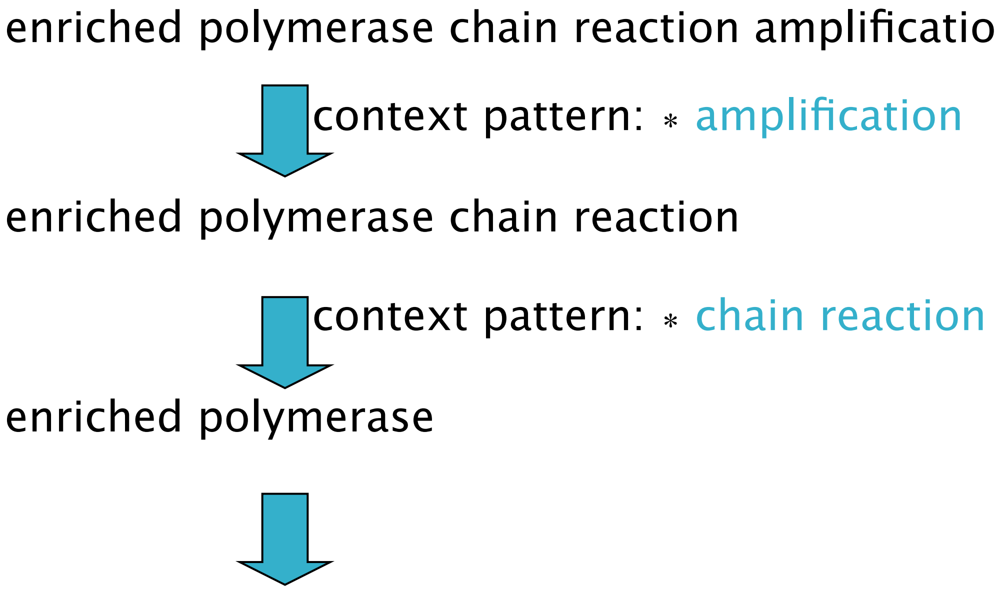
enriched polymerase chain reaction

context pattern: * chain reaction

Post-processing of the candidates

- Using the extracted context patterns, the candidate list is processed and the patterns are removed from the candidate until no matching patterns can be found.

- For example: extracted NG „enriched polymerase chain reaction amplification“



enriched polymerase chain reaction amplification

context pattern: * amplification

enriched polymerase chain reaction

context pattern: * chain reaction

enriched polymerase

Post-processing of the candidates

- Using the extracted context patterns, the candidate list is processed and the patterns are removed from the candidate until no matching patterns can be found.
- For example: extracted NG „enriched polymerase chain reaction amplification“

enriched polymerase chain reaction amplification

↓ context pattern: * amplification

enriched polymerase chain reaction

↓ context pattern: * chain reaction

enriched polymerase

↓ context pattern: enriched *

Post-processing of the candidates

- Using the extracted context patterns, the candidate list is processed and the patterns are removed from the candidate until no matching patterns can be found.

- For example: extracted NG „enriched polymerase chain reaction amplification“

enriched polymerase chain reaction amplification

↓ context pattern: * amplification

enriched polymerase chain reaction

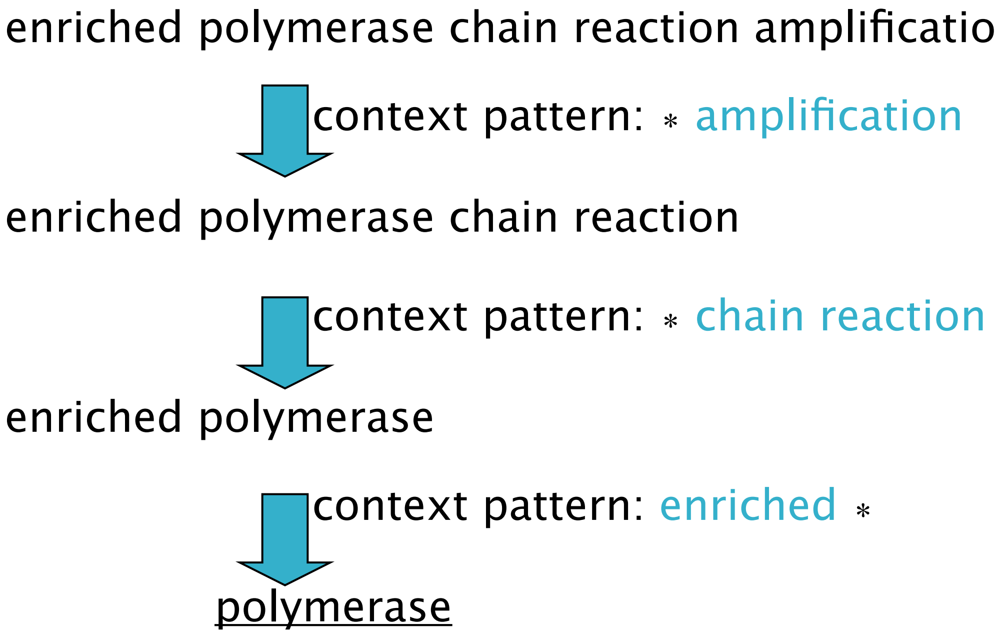
↓ context pattern: * chain reaction

enriched polymerase

↓ context pattern: enriched *

Post-processing of the candidates

- Using the extracted context patterns, the candidate list is processed and the patterns are removed from the candidate until no matching patterns can be found.
- For example: extracted NG „enriched polymerase chain reaction amplification“



enriched polymerase chain reaction amplification

context pattern: * amplification

enriched polymerase chain reaction

context pattern: * chain reaction

enriched polymerase

context pattern: enriched *

polymerase

Validation of candidates (1)

- morpho-syntactic patterns are used in combination with the candidate X as query for a search engine to determine the number of hits
- patterns used
  - hyponym patterns (Hearst 1992 & 1998)
  - additional patterns

<table>
<tr><th>left patterns</th><th>right patterns</th></tr>
<tr><td>proteins (like|such as| including|especially|except| namely|i.e.|e.g.|for example) X</td><td>X (among|and|or|unlike|like) other proteins
X is (a|the) protein
X are (the)? proteins</td></tr>
</table>

Validation of candidates (2)

- semantic similarity between X and „protein“:
  - PMI-IR(X) = Treffer(X, “protein”) / Treffer(X)
- Number of **different** left/right patterns for candidate:
  - p_left(x) = |{ m | m Є L Treffer(x,m) > 0}|
  - p_right(x) = |{ m | m Є R Treffer(x,m) > 0}|
- Value of candidate:
  - Wert(x) = PMI-IR(x) * p_left(x) * p_right(x)
- BNC-filtering:
  - words (stemmed) that appear in a frequent words list are removed.

Evaluation

- **Results from Aimed corpus**
  - **225 Medline abstracts (biomedical)**

	Prec.	Rec.	F1
Extracted NGs	11,55%	31,01%	16,83
Post-processed NGs	13,53%	36,33%	19,72
+ Validation	18,27%	35,63%	24,51
+ Brackets	19,55%	**39,43%**	**26,14**
only NGs with matching context pattern (valid./BNC-filter)	**31,18%**	17,92%	22,76

Generalisability

- Approach can easily be applied to other domains
- The only domain dependent input:

  list of instances of the named entity type to extract
- Domain independent steps
  - generation of candidate list (NG-chunker based on close classed word lists)
  - Extraction and use of context patterns
  - Validation of the post-processed candidates (based on frequency data from the Internet)

Relevant references

- Eichler, K., Hemsen, H. and Neumann G. (2009) Unsupervised and domain-independent extraction of technical terms from scientific articles in digital libraries. In proceedings of the Workshop Information Retrieval 2009 organized as part of LWA, Darmstadt, 2009.

- Eichler, K. and Neumann, G. (2010) Bootstrapping Noun Groups Using Closed-Class Elements Only. In KDML 2010: Knowledge Discovery, Data Mining, and Machine Learning, Kassel, Germany.