

On the evaluation of Polish definition extraction grammars

Adam Przepiórkowski, Łukasz Degórski, Beata Wójtowicz

Polish Academy of Sciences, Institute of Computer Science
ul. Ordona 21, 01-237 Warszawa, Poland

adamp@ipipan.waw.pl, {ldegorski,beataw}@bach.ipipan.waw.pl

Language Technology for eLearning

6th Framework IST Project

<http://www.lt4el.eu>

Use of multilingual language technology tools and semantic web techniques for improving the retrieval of learning material.

9 EU languages, incl. Polish

Subtasks:

- **keyword extraction**
- **identification of term definitions**
- **ontology-enhanced search**

Related work

Definition extraction: Used in terminology extraction, automatic creation of glossaries, question answering, learning lexical semantics relations, automatic construction of ontologies.

Tools:

- language-specific
- involve shallow or deep processing

Most of work - English and other Germanic languages

No previous attempts for Slavic (exception: Bulgarian)

Input

XML-encoded (adhering to XCES) morphosyntactically-annotated text

„Konstruktywizm kładzie nacisk na ...”
(*„Constructivism puts emphasis on ...”*)

```
<s id="s9">  
  <tok base="konstruktywizm" ctag="subst" id="t253" msd="sg:nom:m3">  
    Konstruktywizm  
  </tok>  
  <tok base="kłaść" ctag="fin" id="t254" msd="sg:ter:imperf">kładzie</tok>  
  <tok base="nacisk" ctag="subst" id="t255" msd="sg:acc:m3">nacisk</tok>  
  <tok base="na" ctag="prep" id="t256" msd="acc">na</tok>  
  [...] <tok base="." ctag="interp" id="t273">.</tok>  
</s>
```

Desired output

```
<s id="s9">
  <definingText def="mt1">
    <markedTerm id="mt1">
      <tok base="konstruktywizm" ctag="subst" id="t253" msd="sg:nom:m3">
        Konstruktywizm
      </tok>
    </markedTerm>
    <tok base="kłaść" ctag="fin" id="t254" msd="sg:ter:imperf">kładzie</tok>
    <tok base="nacisk" ctag="subst" id="t255" msd="sg:acc:m3">nacisk</tok>
    <tok base="na" ctag="prep" id="t256" msd="acc">na</tok>
  [...]
  <tok base="." ctag="interp" id="t273">.</tok>
  <definingText>
</s>
```

Shallow Grammars

Grammar: a regular grammar, implemented with the use of the `lxtransduce` tool.

An example rule:

```
<rule name="PP">
  <seq>
    <query match="tok[@ctag = 'prep']"/>
    <ref name="NP1">
      <with-param name="case" value=""/>
    </ref>
  </seq>
</rule>
```

Grammars built manually, based on the definitions annotated by humans in the training corpus.

Shallow Grammars

48 rules in a 16kB file

4 logical layers, each layer calling only rules of the same and previous layers.

- Top-level rules corresponding to various types of definitions (copular, parenthetical, structural, ...)
- Auxilliary rules (e.g. a possible term = an NP followed by genitive NPs, PPs etc.)
- Linguistically justified rules identifying nouns, NPs, PPs etc.
- Low-level, with reference to particular orthographic forms – *copulae*, *definitor verbs* etc.

Shallow Grammars

```
<!-- catches copulae -->
<rule name="copula">
  <seq>
    <query match="tok[@ctag = 'interp']" mult="?"/>
    <first>
      <seq>
        <query match="tok[@base = 'być' and @ctag ~ '^(fin|praet)$']"/>
        <query match="tok[. = 'to']"/>
      </seq>
      <query match="tok[. = 'to']"/>
    </first>
  </seq>
</rule>
```

```
<!-- catches a sequence non-interpunction elements, until full stop -->
<rule name="run_to_full_stop">
  <repeat-until name="tok_no_interp">
    <query match="tok[. = '.']"/>
  </repeat-until>
</rule>
```

Shallow Grammars

Some top-level patterns highly represented in the training corpus:

Pattern	% in training corpus
NP (...) are/is NP _{INS}	15,6%
NP -/: NP	15,2%
NP (are/is) to NP _{NOM}	10,6%
NP VP _{3pers}	9,8%
NP – i.e./or WH-question	4,3%

Corpora

Training: manually annotated definitions used for constructing the grammar; 12 texts

Held out: manually annotated definitions used for fine-tuning the grammar; 1 homogeneous text

Testing: manually annotated definitions unseen during the construction of the grammar; 12 texts

	training	held-out	testing	TOTAL
tokens	139039	84288	77309	300636
sentences	5218	2263	3349	10830
definitions	304	82	172	558
<i>(incl. split)</i>	21	9	24	54

Quantitative evaluation

Comparing **manually** annotated files with the same files annotated **automatically** by the grammar.

Comparison at **token level**:

precision = $\frac{\text{tokens marked in both annotations}}{\text{tokens marked automatically}}$

recall = $\frac{\text{tokens marked in both annotations}}{\text{tokens marked manually}}$

Quantitative evaluation

Comparison at **sentence level**: the sentence is marked in an annotation when it contains at least one marked token in this annotation. The precision and recall is then counted analogously.

Since recall is more important than precision, we use the F measure for the combined result. In general, $F_{\alpha} = \frac{(1+\alpha)pr}{\alpha p+r}$

The appropriate α value should be settled by user case evaluation experiments.

Quantitative evaluation

Baseline grammars used:

- **B1**: marks every sentence as a definition
- **B2**: marks every sentence containing a possible copula (*jest, są, to*), the abbreviation *tj.* 'i.e.' or the word *czyli* 'that is')
- **B3**: marks every sentence containing any of the 27 very simple patterns (mostly single-token keywords) manually identified on the basis of manually annotated definitions

Evaluated grammars:

- **GR**: the initial grammar, based on the training corpus
- **GR'**: improved basing on the held-out corpus

Quantitative evaluation – Training corpus

Token-level		P	R	F₁	F₂	F₅
	B1	4,49	100,00	8,59	12,36	22,00
	B2	6,83	76,56	12,54	17,38	28,33
	B3	6,61	97,29	12,37	17,45	29,29
	GR	16,84	66,49	26,87	33,53	44,58
	GR'	15,28	68,81	25,01	31,75	43,54

Sentence-level		P	R	F₁	F₂	F₅
	B1	5,37	100,00	10,19	14,54	25,39
	B2	10,01	73,57	17,63	23,61	35,75
	B3	9,79	94,64	17,75	24,34	38,72
	GR	23,66	72,50	35,68	42,95	53,94
	GR'	20,53	75,00	32,23	39,80	52,00

Quantitative evaluation – Testing corpus

Token-level	P	R	F ₁	F ₂	F ₅
B1	4,96	100,00	9,45	13,53	23,83
B2	7,82	64,20	13,95	18,87	29,17
B3	8,21	91,49	15,07	20,88	34,00
GR	19,27	48,96	27,65	32,34	38,95
GR'	18,77	56,55	28,18	33,84	42,34

Sentence-level	P	R	F ₁	F ₂	F ₅
B1	5,43	100,00	10,31	14,71	25,64
B2	9,69	61,54	16,74	22,11	32,53
B3	10,54	88,46	18,84	25,54	39,64
GR	19,34	51,65	28,14	33,18	40,40
GR'	18,69	59,34	28,42	34,39	43,55

Interannotator agreement

Cohen's kappa statistic with $\Pr(a)$ and $\Pr(e)$ calculated

at token level: $\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$

Assumption: two annotations agree on a token if it belongs to both or neither.

$$\Pr(e) = p_1 p_2 + (1 - p_1)(1 - p_2)$$

$$\Pr(a) = \frac{\text{number of tokens marked by both or neither}}{\text{number of all tokens}}$$

Interannotator agreement

Cohen's kappa statistic with $\Pr(a)$ and $\Pr(e)$ calculated

at token level: $\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$

Assumption: two annotations agree on a token if it belongs to both or neither.

$$\Pr(e) = p_1 p_2 + (1 - p_1)(1 - p_2)$$

$$\Pr(a) = \frac{\text{number of tokens marked by both or neither}}{\text{number of all tokens}}$$

Result: **0.26**

OUCH!

Interannotator agreement

But is token-based IAA a good model?

It simulates throwing a (weighed) coin for each token. This would result in many short (mostly 1 token long) “definitions”.

Sentence-level approach: we classify whole **sentences**.

Interannotator agreement

But is token-based IAA a good model?

It simulates throwing a (weighed) coin for each token. This would result in many short (mostly 1 token long) “definitions”.

Sentence-level approach: we classify whole **sentences**.

Result of our experiment on 83K subcorpus of the training corpus: **0.31**

Interannotator agreement – contingency tables

		def.	not def.	TOTAL
Token level	def.	1593	5702	7295
	not def.	1740	74197	75937
	TOTAL	3333	79899	83232
Sentence level		def.	not def.	TOTAL
	def.	127	419	546
	not def.	39	2968	3007
	TOTAL	166	3387	3553

Second annotator marked 595 definitions, while the first one – just 158.

Interannotator agreement – PABAK

Bias: different “probability of marking” a token/sentence by each annotator

Prevalence: probability of “yes” much different than “no” (here: much lower)

PABAK: *Prevalence-adjusted bias-adjusted kappa*

Token level: **0.82**

Sentence level: **0.74**

But aren't the effects of bias and prevalence actually meaningful?

Interannotator agreement - κ_{max}

Let's compare κ with the maximum value it could attain given the actual proportions of decisions by annotators.

Token-level $\kappa_{max} = 0.61$

Sentence-level $\kappa_{max} = 0.43$

One more argument for evaluating extraction at sentence level): interannotator agreement is much higher at sentence level (0.31 out of possible 0.43 = **0.72**) than at the token level (0.26 out of possible 0.61 = **0.43**).

The end

```
<definingText def="mtTheEnd">  
  <markedTerm id="mtTheEnd">  
    <tok>Applause</tok>  
  </markedTerm>  
  <tok>:</tok>  
  <tok>hand</tok>  
  <tok>clapping</tok>  
  <tok>as</tok>  
  <tok>a</tok>  
  <tok>demonstration</tok>  
  <tok>of</tok>  
  <tok>approval</tok>  
</definingText>
```