

Prague Dependency Treebank Annotation Errors: A Preliminary Analysis

Vojtěch Kovář, M. Jakubíček

Faculty of Informatics, Masaryk University Brno
Botanická 68a, 602 00 Brno, Czech Republic
xkovar3@fi.muni.cz

RASLAN 2009

Outline

- 1 Introduction, Prague Dependency Treebank
- 2 Error Analysis
- 3 Conclusions

Motivation

■ Manually annotated NL resources

- crucial for developing and testing NLP software
- created by humans → contain errors
- need for error analysis to improve the quality

■ Prague Dependency Treebank (PDT)

- the biggest syntactically annotated corpus of Czech
- functional generative description – 3 layers of annotation
- here we consider the analytical layer only
 - dependency trees
- extensive manual for annotators exists
- a lot of errors and inconsistencies

Motivation

- Manually annotated NL resources
 - crucial for developing and testing NLP software
 - created by humans → contain errors
 - need for error analysis to improve the quality
- Prague Dependency Treebank (PDT)
 - the biggest syntactically annotated corpus of Czech
 - functional generative description – 3 layers of annotation
 - here we consider the analytical layer only
 - dependency trees
 - extensive manual for annotators exists
 - a lot of errors and inconsistencies

Motivation

- Manually annotated NL resources
 - crucial for developing and testing NLP software
 - created by humans → contain errors
 - need for error analysis to improve the quality
- Prague Dependency Treebank (PDT)
 - the biggest syntactically annotated corpus of Czech
 - functional generative description – 3 layers of annotation
 - here we consider the analytical layer only
 - dependency trees
 - extensive manual for annotators exists
 - a lot of errors and inconsistencies

Motivation

- Manually annotated NL resources
 - crucial for developing and testing NLP software
 - created by humans → contain errors
 - need for error analysis to improve the quality
- Prague Dependency Treebank (PDT)
 - the biggest syntactically annotated corpus of Czech
 - functional generative description – 3 layers of annotation
 - here we consider the analytical layer only
 - dependency trees
 - extensive manual for annotators exists
 - a lot of errors and inconsistencies

Motivation

- Manually annotated NL resources
 - crucial for developing and testing NLP software
 - created by humans → contain errors
 - need for error analysis to improve the quality
- Prague Dependency Treebank (PDT)
 - the biggest syntactically annotated corpus of Czech
 - functional generative description – 3 layers of annotation
 - here we consider the analytical layer only
 - dependency trees
 - extensive manual for annotators exists
 - a lot of errors and inconsistencies

Motivation

- Manually annotated NL resources
 - crucial for developing and testing NLP software
 - created by humans → contain errors
 - need for error analysis to improve the quality
- Prague Dependency Treebank (PDT)
 - the biggest syntactically annotated corpus of Czech
 - functional generative description – 3 layers of annotation
 - here we consider the analytical layer only
 - dependency trees
 - extensive manual for annotators exists
 - a lot of errors and inconsistencies

Motivation

- Manually annotated NL resources
 - crucial for developing and testing NLP software
 - created by humans → contain errors
 - need for error analysis to improve the quality
- Prague Dependency Treebank (PDT)
 - the biggest syntactically annotated corpus of Czech
 - functional generative description – 3 layers of annotation
 - here we consider the analytical layer only
 - dependency trees
 - extensive manual for annotators exists
 - a lot of errors and inconsistencies

Motivation

- Manually annotated NL resources
 - crucial for developing and testing NLP software
 - created by humans → contain errors
 - need for error analysis to improve the quality
- Prague Dependency Treebank (PDT)
 - the biggest syntactically annotated corpus of Czech
 - functional generative description – 3 layers of annotation
 - here we consider the analytical layer only
 - dependency trees
 - extensive manual for annotators exists
 - a lot of errors and inconsistencies

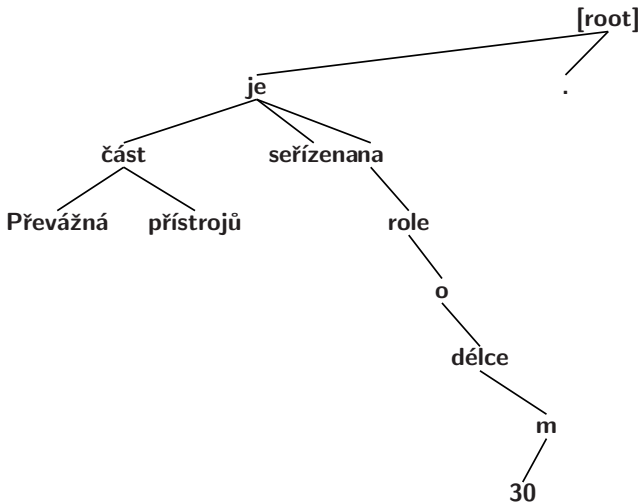
Motivation

- Manually annotated NL resources
 - crucial for developing and testing NLP software
 - created by humans → contain errors
 - need for error analysis to improve the quality
- Prague Dependency Treebank (PDT)
 - the biggest syntactically annotated corpus of Czech
 - functional generative description – 3 layers of annotation
 - here we consider the analytical layer only
 - dependency trees
 - extensive manual for annotators exists
 - a lot of errors and inconsistencies

Motivation

- Manually annotated NL resources
 - crucial for developing and testing NLP software
 - created by humans → contain errors
 - need for error analysis to improve the quality
- Prague Dependency Treebank (PDT)
 - the biggest syntactically annotated corpus of Czech
 - functional generative description – 3 layers of annotation
 - here we consider the analytical layer only
 - dependency trees
 - extensive manual for annotators exists
 - **a lot** of errors and inconsistencies

Převážná část přístrojů je seřízena na role o délce 30 m.



Error Analysis

■ Background

- developing and testing a new rule-based parser for Czech
- tried to approximate PDT data
- problems with the quality of the data

■ Preliminary analysis

- sample = 60 sentences from the `train-1` set
- basic error classification
- estimation of overall percentage of error
- finding examples

Error Analysis

■ Background

- developing and testing a new rule-based parser for Czech
- tried to approximate PDT data
- problems with the quality of the data

■ Preliminary analysis

- sample = 60 sentences from the `train-1` set
- basic error classification
- estimation of overall percentage of error
- finding examples

Error Analysis

■ Background

- developing and testing a new rule-based parser for Czech
- tried to approximate PDT data
- problems with the quality of the data

■ Preliminary analysis

- sample = 60 sentences from the `train-1` set
- basic error classification
- estimation of overall percentage of error
- finding examples

Error Analysis

■ Background

- developing and testing a new rule-based parser for Czech
- tried to approximate PDT data
- problems with the quality of the data

■ Preliminary analysis

- sample = 60 sentences from the `train-1` set
- basic error classification
- estimation of overall percentage of error
- finding examples

Error Analysis

■ Background

- developing and testing a new rule-based parser for Czech
- tried to approximate PDT data
- problems with the quality of the data

■ Preliminary analysis

- sample = 60 sentences from the `train-1` set
- basic error classification
- estimation of overall percentage of error
- finding examples

Error Analysis

■ Background

- developing and testing a new rule-based parser for Czech
- tried to approximate PDT data
- problems with the quality of the data

■ Preliminary analysis

- sample = 60 sentences from the train-1 set
- basic error classification
- estimation of overall percentage of error
- finding examples

Error Analysis

■ Background

- developing and testing a new rule-based parser for Czech
- tried to approximate PDT data
- problems with the quality of the data

■ Preliminary analysis

- sample = 60 sentences from the train-1 set
- basic error classification
- estimation of overall percentage of error
- finding examples

Error Analysis

■ Background

- developing and testing a new rule-based parser for Czech
- tried to approximate PDT data
- problems with the quality of the data

■ Preliminary analysis

- sample = 60 sentences from the train-1 set
- basic error classification
- estimation of overall percentage of error
- finding examples

Error Analysis

■ Background

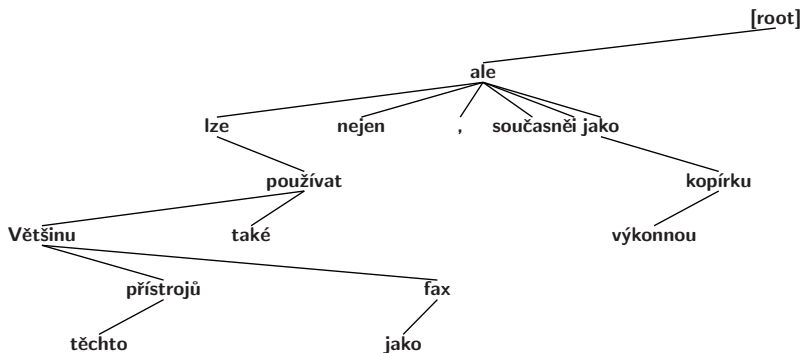
- developing and testing a new rule-based parser for Czech
- tried to approximate PDT data
- problems with the quality of the data

■ Preliminary analysis

- sample = 60 sentences from the train-1 set
- basic error classification
- estimation of overall percentage of error
- finding examples

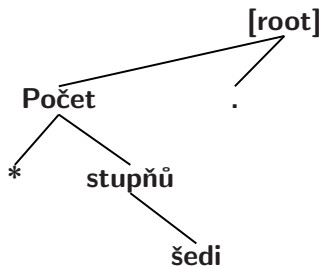
Random Errors

Většinu těchto přístrojů lze také používat nejen jako fax,
ale současně i jako výkonnou kopírku...



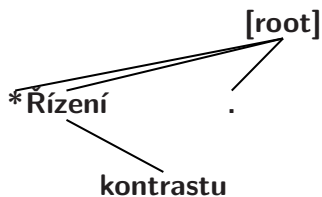
Inconsistencies

* Počet stupňů šedi.



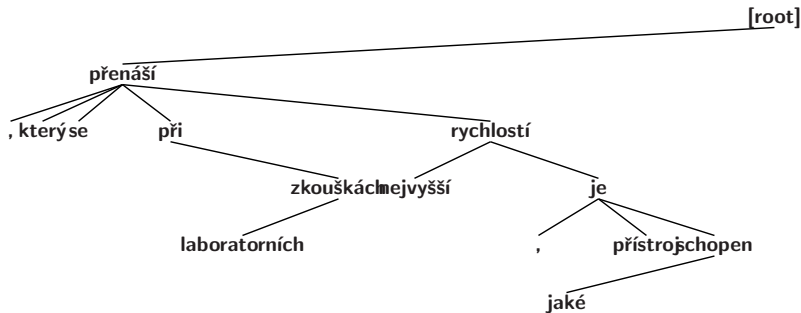
Inconsistencies

* Řízení kontrastu.



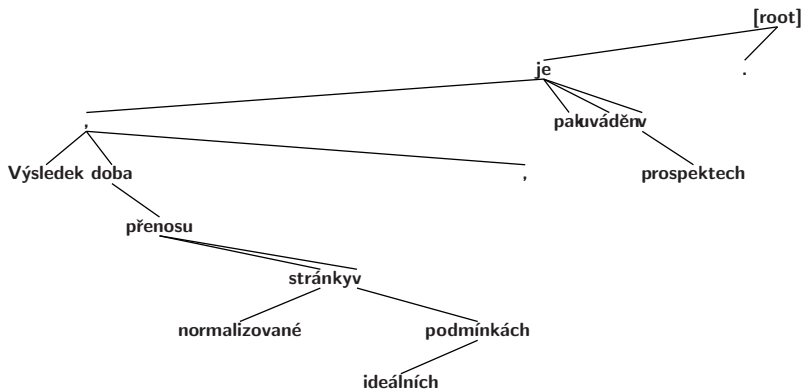
Inconsistencies – passive verb phrases

..., který se přenáší při laboratorních zkouškách nejvyšší rychlostí, jaké je přístroj schopen...



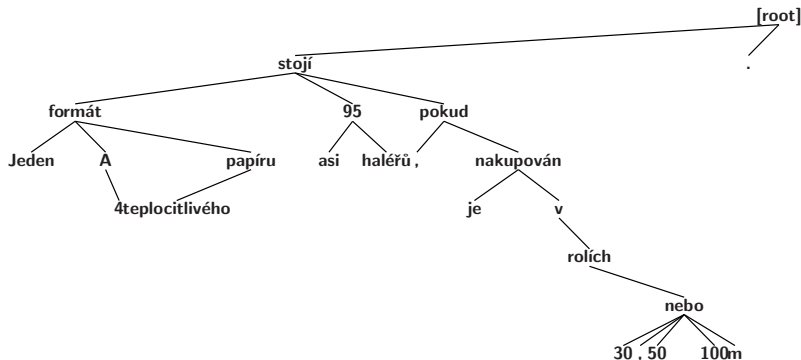
Inconsistencies – passive verb phrases

Výsledek, doba přenosu normalizované stránky v ideálních podmínkách, je pak uváděn v prospektech.



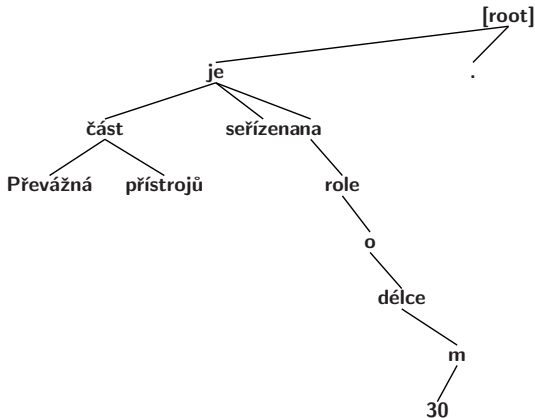
Inconsistencies – passive verb phrases

Jeden formát A 4 teplocitlivého papíru stojí asi 95 haléřů, pokud je nakupován v rolích 30, 50 nebo 100 m.



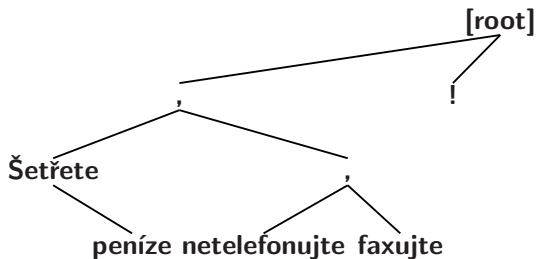
Inconsistencies – numerals

Převážná část přístrojů je seřizena na role o délce 30 m.



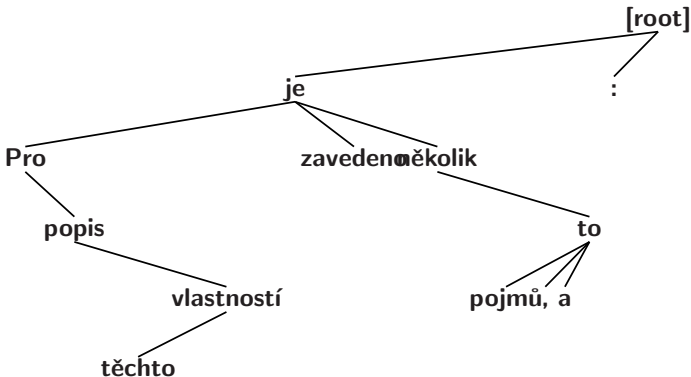
The “Dirty” Cases

Šetřete peníze, netelefonujte, faxujte!



The “Dirty” Cases

Pro popis těchto vlastností je zavedeno několik pojmů, a to:



Conclusions

■ Annotation errors

- 5 to 10 percent of annotation is erroneous
- single mistake can cause structural changes
- disputable usability of the resource

■ On parser evaluation

- really use treebanks?
- application-driven testing

Conclusions

■ Annotation errors

- 5 to 10 percent of annotation is erroneous
- single mistake can cause structural changes
- disputable usability of the resource

■ On parser evaluation

- really use treebanks?
- application-driven testing

Conclusions

■ Annotation errors

- 5 to 10 percent of annotation is erroneous
- single mistake can cause structural changes
- disputable usability of the resource

■ On parser evaluation

- really use treebanks?
- application-driven testing

Conclusions

■ Annotation errors

- 5 to 10 percent of annotation is erroneous
- single mistake can cause structural changes
- disputable usability of the resource

■ On parser evaluation

- really use treebanks?
- application-driven testing

Conclusions

■ Annotation errors

- 5 to 10 percent of annotation is erroneous
- single mistake can cause structural changes
- disputable usability of the resource

■ On parser evaluation

- really use treebanks?
- application-driven testing

Conclusions

■ Annotation errors

- 5 to 10 percent of annotation is erroneous
- single mistake can cause structural changes
- disputable usability of the resource

■ On parser evaluation

- really use treebanks?
- application-driven testing

Conclusions

■ Annotation errors

- 5 to 10 percent of annotation is erroneous
- single mistake can cause structural changes
- disputable usability of the resource

■ On parser evaluation

- really use treebanks?
- application-driven testing