Treelet model for HPSG-parsing with error correction

Angelina Ivanova
Project carried out at University of Groningen

PhD Seminar in Language Technology
08.10.2013
Oslo, Norway

Motivation

- HPSG-parsing with error-correction
- Cross-sentence comparability of parse tree probabilities

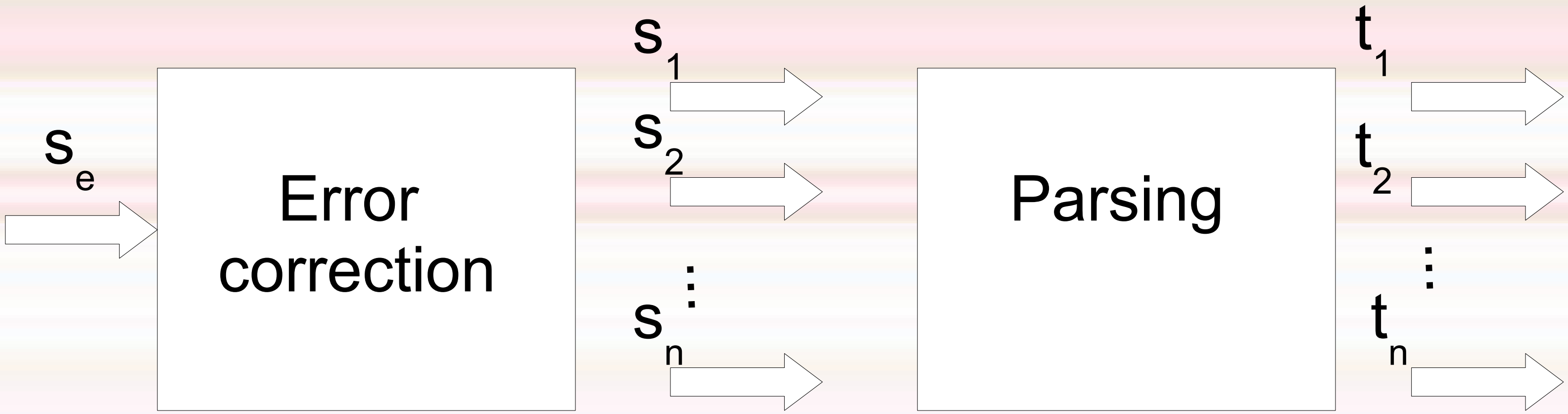
Related work

- Shared tasks on grammatical error correction: HOO 2011, 2012; CoNLL 2013
- Influence of errors on parse tree probabilities: Wagner and Foster (2009)
- Treelet model for error correction: Pauls and Klein (2012), Yoshimoto et al. (2013)
- HPSG-parsing with error correction: Flickinger and Yu (2013)

System

- Generate weighted versions of a sentence with a grammatical error
  Approaches: n-gram (Lee and Seneff, 2006), Levenshtein-distance kernel (Levy, 2008)
- Parse candidate sentences with PET
- Choose the best version by the highest joint probability of the version and its parse tree

System



$$s = s_i \mid \max(P(s_i, t_i))$$

s_e – erroneous sentence

$s_1 \ldots s_n$ – versions of s_e with error correction

t_1... tn – parse trees of versions

s – best corrected version of s_e

System

- Sentence with an error:
  Am I feeding my ***prt*** *enough?*

- Corrected sentence versions:

- *Am I feeding my* ***pet*** *enough?*
- *Am I feeding my* ***put*** *enough?*
- *Am I feeding my* ***part*** *enough?*

Requirement

- **Generative** probabilities of parse trees
- Reason:

  we need to compare probabilities of parse trees of different sentences (corrected versions of the sentence with an error)

Parse ranking in PET

- PET exploits maximum entropy model for parse selection (**discriminative**)
- Only parse trees of the same sentence could be compared by the scores
- To obtain generative probabilities unpacking of the whole forest is required, which is not easily feasible

Possible solution

Apply treelet model to compute generative probabilities of the parse trees

Treelet model (Pauls and Klein, 2012)

r = P $\rightarrow$ C1, …, Cd

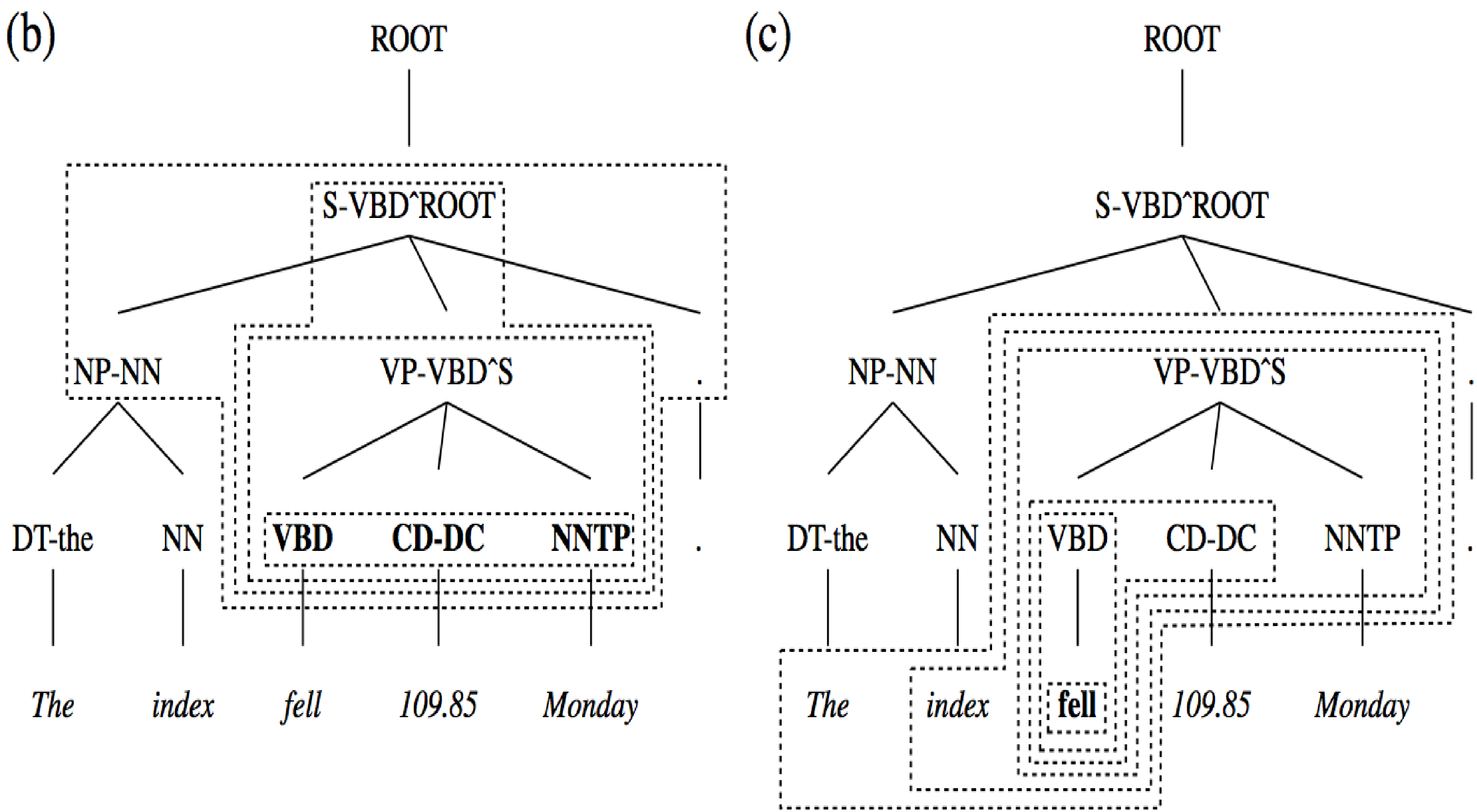
r – parent symbol
C1, …, Cd – children

Probability of the parse tree:

$$p(T) = \prod_{r \in T} p(C_1^d | h)$$

h - context

Context
(Pauls and Klein, 2012)



Zero probabilities: non-terminal producitons

- Backing-off

$p(C_1^d|r', P', P)$ ➡ $p(C_1^d|P', P)$ ➡ $p(C_1^d|P)$ ➡

➡ $$\lambda \prod_{i=1}^{d} p(C_i|P) + (1-\lambda) \prod_{i=1}^{d} p(C_i)$$ ➡

➡ $$P_{WB}(C_i) = \frac{c_h(\epsilon)}{c_h(\epsilon) + N_{1+}(\epsilon)} P_{MLE}(C_i) + \\ + \frac{N_{1+}(\epsilon)}{c_h(\epsilon) + N_{1+}(\epsilon)} \frac{1}{|V|}$$

Back-off parameters

- Estimation-maximization algorithm

The algorithm searches for λ_j that would minimize

$$-\frac{1}{|H|}\sum_{i\ldots|H|}\log_2(p'_\lambda(w_i|h_i))$$

where H is the size of the development set.

Algorithm finds maximum likelihood estimates of the parameters of the statistical model. It alternates between the two steps: estimation and maximization.

Zero probabilities: lexical level

- We have seen all the lexical items of which the ngram is composed on the training set, but we haven't seen such ngram.
  Solution: smoothing

- We haven't seen one or several lexical items of which the ngram is composed.
  Solution: <UNK> token

Modified Kneser-Ney Smoothing

$$p_{KN}(w_i|w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i}) - D(c(w_{i-n+1}^{i}))}{\sum_{w_i} c(w_{i-n+1}^{i})} + \gamma(w_{i-n+1}^{i-1}) p_{KN}(w_i|w_{i-n+2}^{i-1})$$

where

$$D(c) = \begin{cases} 0 & \text{if } c = 0 \\ D_1 & \text{if } c = 1 \\ D_2 & \text{if } c = 2 \\ D_{3+} & \text{if } c \geq 3. \end{cases}$$

$$\gamma(w_{i-n+1}^{i-1}) = \frac{D_1 N_1(w_{i-n+1}^{i-1}\cdot) + D_2 N_2(w_{i-n+1}^{i-1}\cdot) + D_{3+} N_{3+}(w_{i-n+1}^{i-1}\cdot)}{\sum_{w_i} c(w_{i-n+1}^{i})}$$

Modified Kneser-Ney Smoothing

$$N_1(w_{i-n+1}^{i-1}\cdot) = |\{w_i : c(w_{i-n+1}^{i-1}w_i) = 1\}|$$

the number of words that appear after the context w_{i-n+1}^{i-1} exactly once.

$$N_2(w_{i-n+1}^{i-1}\cdot) = |\{w_i : c(w_{i-n+1}^{i-1}w_i) = 2\}|$$

the number of words that appear after the context w_{i-n+1}^{i-1} exactly twice.

$$N_{3+}(w_{i-n+1}^{i-1}\cdot) = |\{w_i : c(w_{i-n+1}^{i-1}w_i) \geq 3\}|$$

the number of words that appear after the context w_{i-n+1}^{i-1} three or more times.

Modified Kneser-Ney Smoothing

$$D_1 = 1 - 2Y\frac{n_2}{n_1}$$

$$D_2 = 2 - 3Y\frac{n_3}{n_2}$$

$$D_{3+} = 3 - 4Y n_4 n_3$$

$$Y = \frac{n_1}{n_1 + 2n_2}$$

where n_1, n_2, n_3 and n_4 are the total number of n-grams with exactly one, two, three and four respectively, in the training data.

Modified Kneser-Ney Smoothing

If

$$\sum_{w_i} c(w_{i-n+1}^{i}) = 0$$

1) full backoff to the lower level n-gram
2) setting the probability to a small constant

$\mu = 0.000001$

<UNK> tokens

Hapax to model unknown words

Choose vocabulary in advance and replace other words in the training corpus with <UNK> (12%)

ERG data

- **36,918** sentences from DeepBank 1.0 (sections 0-21 of PTB in ERG representation)
- **50,997** sentences from WeSearch, SemCor, Verbmobil and other resources

ERG data

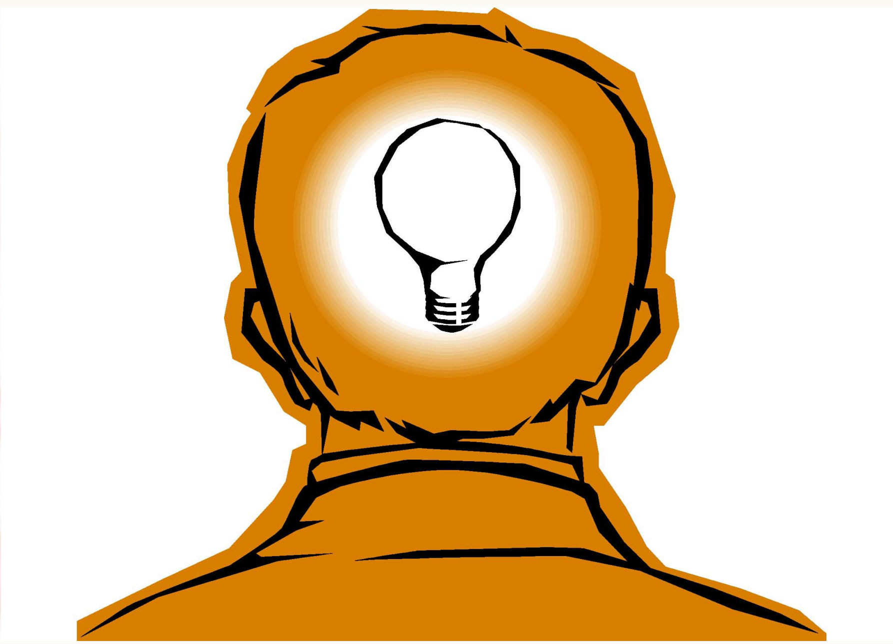
Train	63,298 sentences
Development	7,034 sentences
Test	17,583 sentences

NUS Corpus

- NUS Corpus of Learner English
- We collected only non-overlapping corrections that are in the scope of one paragraphs
- **2,181** sentence pairs

Wikipedia

- **3,959** pairs of aligned sentences from Wikipedia 2012 and Wikipedia 2013



Hypothesis

Sent. from Wiki 13 are corrections for sent. from Wiki 12

Experiments

- Parse selection
- Scoring parse trees of erroneous and corrected sentences

Treelet model for parse selection

Upper-bound	12,311 sent.	100%
Treelet	4,487 sent.	**36.45%**
PCFG	2,905 sent.	23.60%
Random	621 sent.	5.04%

Treelet model for parse selection

- Treelet model gives 36.44% exact match.
- Zhang et al. (2007): 56.83% exact match for selective unpacking.
- Differences:
  1) Multiple domains vs. one domain
  2) Size of datasets:
  63,298 vs 8,000 for training
  12,255 vs 1,603 for testing

Treelet model for scoring parse trees of erroneous and corrected sentences

Model	NUS corpus			Wikipedia		
	Corrected	Equal	Original	Corrected	Equal	Original
Oracle	2,223		0	4,604		0
Treelet	1,449	11	763	1,884	994	1,726
PCFG	1,304	11	908	1,835	996	1,773
Trigram	1,249	80	894	1,732	1,294	1,578
Random	1,112		1,111	2,302		2,302

Statistical significance

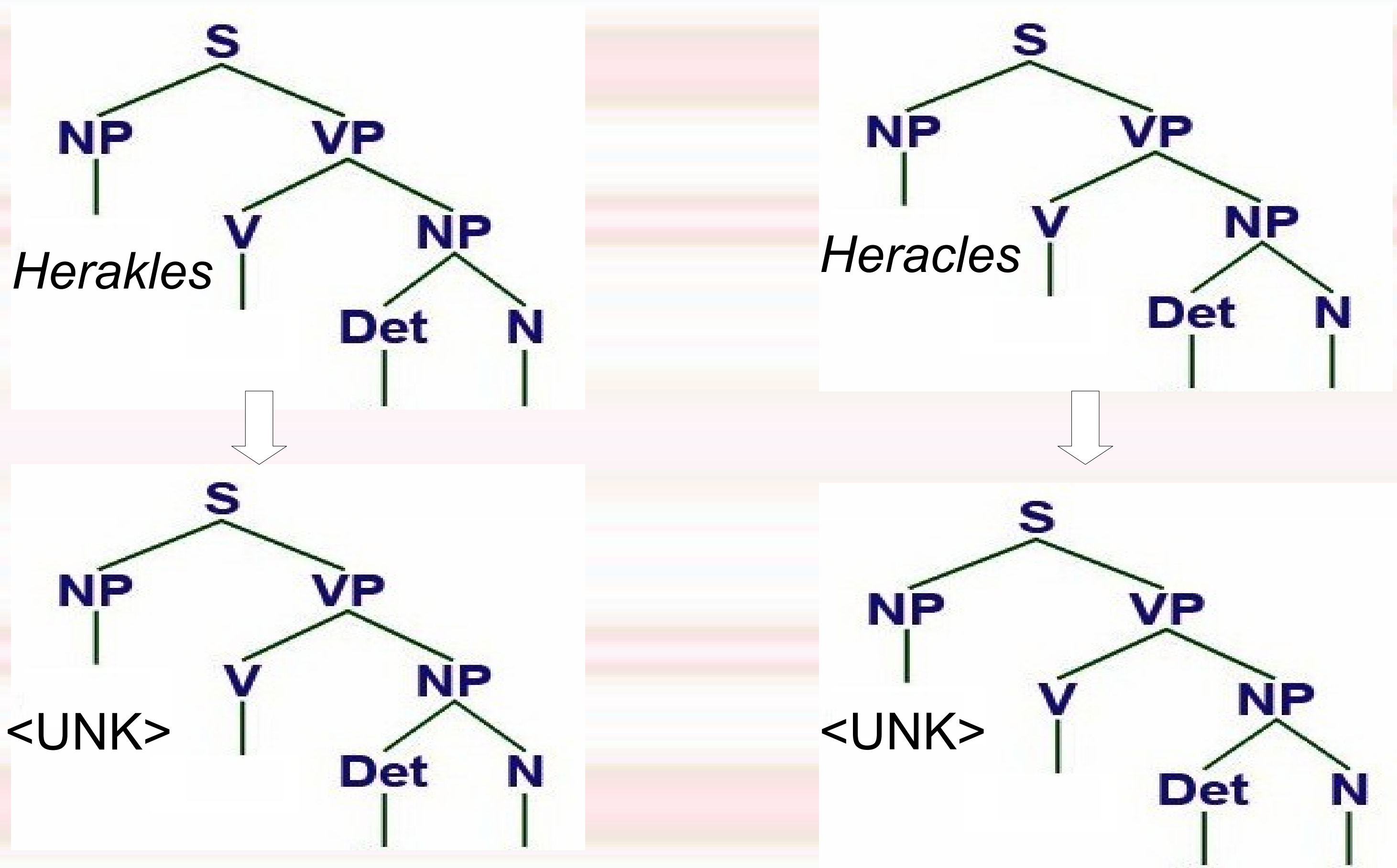
- Binomial test
- Population proportions
- Analysis of variance

- Results on the NUS corpus are significant
- Results on the Wikipedia dataset are insig-nificant

Wikipedia errors

- Noise
  will be created - were be created
- Proper nouns
  Christobal – Christóbal
  Herakles – Heracles
  Stuebing – Stübing
- Semantic errors
  most – many
  youth days after his birth – four days after his birth
- Stylistic errors
  you – one
- Discourse-level errors
  his daughter – their daughter

Proper nouns



Possible solution: add lists of proper nouns to vocabulary

Conclusions

- The treelet model outperforms PCFG for parse selection but is probably weaker than ME
- The treelet model scores parse trees of corrected sentences more often than PCFG and trigram on the NUS corpus
- The treelet, PCFG, trigram and random models perform similarly on the Wikipedia dataset
- Results on Wikipedia are related to the types of errors present in the resource

Contributions

- The Wikipedia dataset (pairs of parallel sentences from Wiki12 and Wiki13)
- Application of the treelet model to the two tasks