Approximating Parse Probabilities with Simple Probabilistic Models

Joachim Wagner
2006-02-08
Supervisors: Josef van Genabith and Monica Ward

Layout of the Talk

- Motivation
  - grammar checker
  - grammaticality and parse probability
  - my approach to detecting ungrammatical sentences
  - requirements
- Models
  - simple models
  - language modelling
  - combining models
- Conclusions

Motivation

- Grammar checkers are useful in
  - Word processors
  - Computer-assisted language learning
- Current grammar checkers
  - Low-level techniques from computational linguistics, e.g. part-of-speech patterns
  - Hand-crafted grammars → parsing

Hand-Crafted Grammars

- Grammar writing is labour-intensive
  - Needs to be repeated for each language
- Only few grammars with good coverage available for English
- For grammar checkers:
  - Reject ungrammatical sentences
  - 2nd stage: add rules to analyse the error

Data-Driven Methods

- Various methods to train or induce grammars from
  - Labelled corpora, especially treebanks
  - Unlabelled corpora
- Over-generalisation
  - Fail to reject ungrammatical sentences
  - Produce many analysis
- To date, probability models address the latter
- In my research, they will address the former

Research Question

- Can the output of existing probabilistic, data-driven parsers be exploited to:
  - judge grammaticality of sentences
  - locate errors within sentences?

Grammaticality and Parse Probability

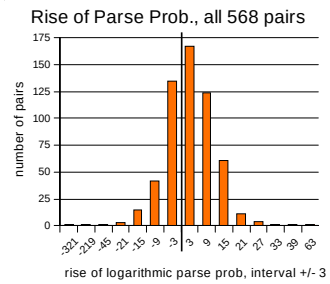
- How does grammaticality influence the probability of the most likely parse?
- Parallel error corpus (Foster 2005)
  - 923 ungrammatical sentences
  - 1 or 2 corrections each
  - 2048 sentences in total
- 50 % development set

Parallel Error Corpus - Example

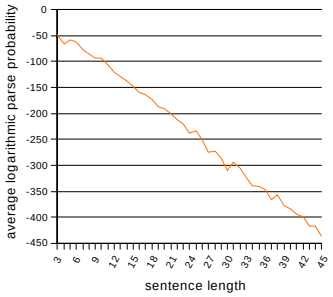
- Ungrammatical sentence
  "**Does** your circles overlap?"
  → 3.3 x 10^{-28}
- Corrected sentence
  "**Do** your circles overlap?"
  → 23.0 x 10^{-28}

[Student assignment 22/12/04]

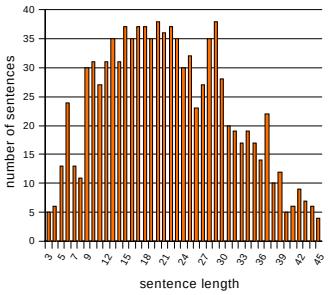
Observations 1



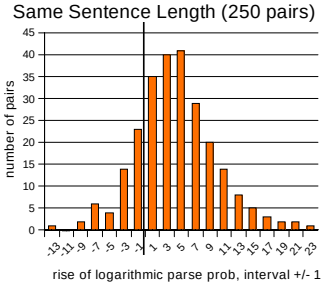
Influence of Sentence Length



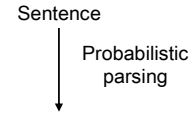
Influence of Sentence Length



Observations 2

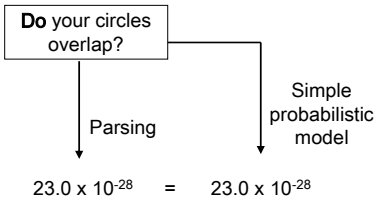


Summary

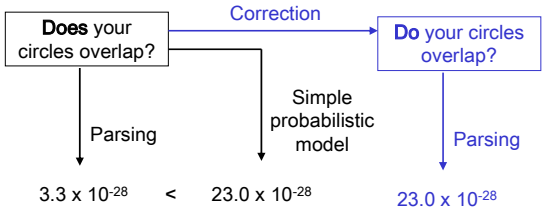


- Manually correcting an ungrammatical sentence increases probability
- Big variance among different sentences
- Too much overlap for a simple threshold method

Detecting Grammatical Sentences



Detecting Ungrammatical Sentences



Requirements for the Probabilistic Model

- Predict parse probability of grammatical sentence accurately (log. error < 5)
- Fail to emulate the lower probability of ungrammatical sentences
- May look at the parse of the sentence
- Not really predicting the probability of a hypothecical correction

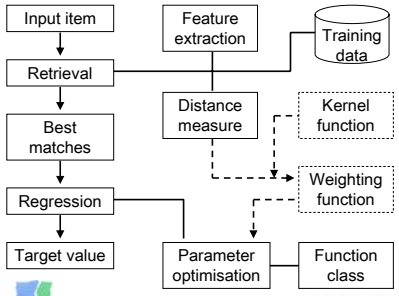
Layout of the Talk

- Motivation
  - grammar checker
  - grammaticality and parse probability
  - my approach to detecting ungrammatical sentences
  - requirements
- Models
  - simple models
  - language modelling
  - combining models
- Conclusions

Instance-Based Learning

- Retrieve similar sentence from training corpus
- Choose parse probability based on these sentences, e.g. average
- *k*-nearest neighbour method
  - simple implementation
  - few parameters
  - assumes Euclidean space

k-Nearest Neighbour Method



Example: **Do** your circles overlap ?

Distance	Sentence	Log. Prob.
0.24	Is Mr Fatuzzo there ?	-60.33
0.42	Is Burma really isolated ?	-62.00
0.68	Should embryos be cloned ?	-57.50
0.73	(Mr Crowley refused)	-59.11
0.74	Should we reprimand ministers ?	-59.84
0.76	Subject : Phare - Poland	-71.65
0.77	Subject : ASEAN and Burma	-70.43
0.80	Is Mr Duisenberg present ?	-64.05
0.81	Structural Funds (continuation)	-57.08
0.81	Have I understood correctly ?	-49.56
		-61.16

Example: **Does** your circles overlap ?

Distance	Sentence	Log. Prob.
0.05	Is Mr Fatuzzo there ?	-60.33
0.44	Is Burma really isolated ?	-62.00
0.45	(Mr Crowley refused)	-59.11
0.57	Structural Funds (continuation)	-57.08
0.60	Euro-Mediterranean cooperation (continuation)	-67.12
0.60	Have I understood correctly ?	-49.56
0.60	Have I understood correctly ?	-49.56
0.60	Have I understood correctly ?	-49.56
0.60	Should we reprimand ministers ?	-59.84
0.63	(Loud sustained applause)	-57.68
		-57.18

Training Data

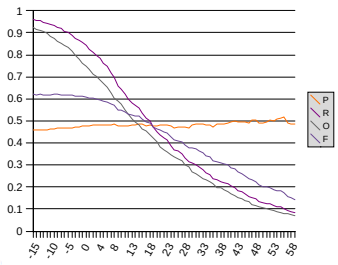
- ½ of English EuroParl (1.1M sentences)
  - different domain
- Presumably grammatical
- Excluded sentences containing quotes
- Parser crashes and hang-ups
- 409,736 sentences

Evaluation Measures

- Mean square error of prediction of logarithmic Parse probability of grammatical sentences
  - EuroParl corpus, cross-validation N=10
- Precision and recall in task of classifying sentences
  - Parallel error corpus

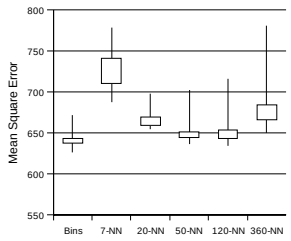
Results (1a)

Sentence Lenth only



Results (1b)

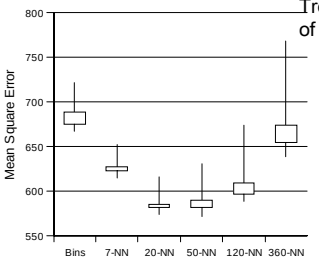
Sentence Length Only



Results (2)

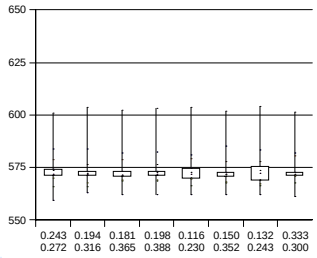
Using All 3 Features

(Sentence Length, Tree Height and # of internal nodes)



Results (3)

Scaling TH & IN



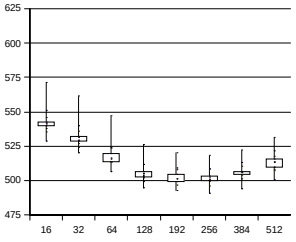
Scaling or Weighting

Character Trigrams

- Easy to integrate into k-NN model (vector of normalised trigram counts)
- Prov
- Ideally, add all trigrams
- K-NN slow with high-dimensional data

Character Trigrams (2)

Trigram Counts / N



Scaling or Weighting

Summary – Simple Models

Model	Mean Square Error
Sentence length only	640
Adding tree height and number of internal nodes	580
Scaling / Weighting	570
Adding Trigrams	502

Statistical Language Modelling

- Works on raw string of tokens
- Markov assumption: probability of token only depends on previous (n-1) tokens
- N = 1 and MLE: token frequencies
- Unseen events
  - discounting / smoothing

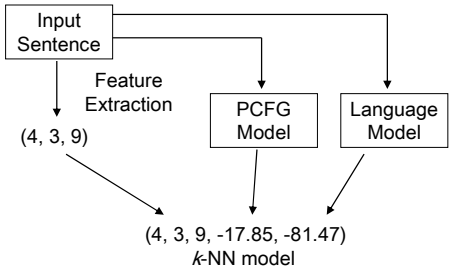
Results

- Mean Square Error
  - 1-gram: 3,464
  - 2-gram: 20,890
  - POS tagged token: 6,778

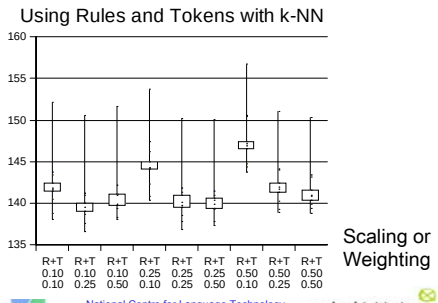
Terminal Rules of a PCFG

- Condition token probability on POS tag
- Corresponds to terminal rule of PCFG
- Motivation:
  - might be more related to terminal probabilities in parse tree than ordinary language models
- Mean Square Error: 23,352

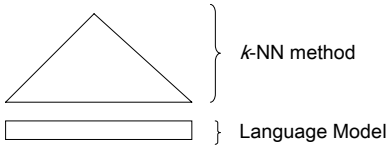
Combining Models – Method 1



Results (Method 1)



Combining Models – Method 2



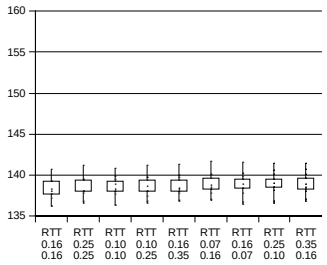
- Training data: extract (divide by) language model output
- Making predictions: include (multiply with) language model output

Results (Method 2)

- Combined with simple model (SL/TH/IN)
- Mean Square Error
  - 1-gram: 200
  - 2-gram: 610
  - PCFG rules: 220
  - POS tagged: 690

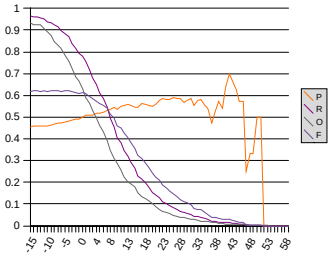
Combining Both Methods

R+T (k-NN) and T extracted



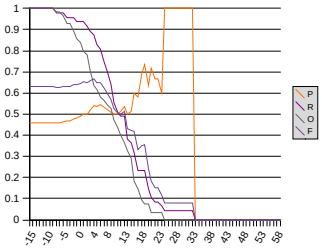
Results (P&R)

R+T / T



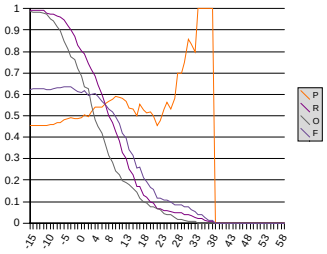
Result Broken Down by Sentence Length

R+T/T, SL 0-9



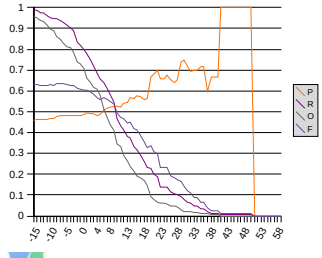
Result Broken Down by Sentence Length

R+T/T, SL 10-19



Result Broken Down by Sentence Length

R+T/T, SL 20-29



Conclusion

- Each idea improved MSE
- Eventually precision increased noticeable above baseline
- Precision not yet very useful
- Many possible ways to further improve the model

Thank you!

- Any questions?