Introduction:

Exploring Evidence for Shallow Parsing

By Xin Li and Dan Roth

Department of Computer Science

University of Illinois, USA

Overview

- This overview
- Introduction
- Experimental Design
- Experimental Results
- Conclusion

Introduction

- Recent effort towards partial analysis of natural language sentences
- Earlier work concentrated on manual construction of rules

Introduction

- Research motivated by
  - observation that shallow syntactic information can be extracted using local information
  - psycholinguistics arguments that full parsing is not a realistic strategy for sentence processing and analysis in many scenarios

Introduction

- Evaluate whether this is worthwhile in comparison to full sentence parsing
- Could shallow parsing be more accurate and more robust than full parsing?

Experimental Design

- For accuracy: identify the phrase structure of sentences
  - atomic phrases
  - “chucking”
- For robustness: two different types of data
  - Standard Penn Treebank data
  - Switchboard data

Experimental Design

- Def^n for phrases used in the chunking competition in CoNLL-2000 (Tjong Kim Sang and Buchholz, 2000). Full parse trees in a flat form, e.g.:

  [NP He] [VP reckons] [NP the current account deficit] [VP will narrow] [PP to] [NP only $ 1.8 billion] [PP in] [NP September] .

Experimental Design

- The goal is to accurately predict 11 different types of phrases. The phrases are:
  - adjective phrase (ADJP),
  - adverb phrase (ADVP),
  - conjunction phrase (CONJP),
  - interjection phrase (INTJ),
  - list marker (LST),
  - noun phrase (NP),
  - preposition phrase (PP),
  - particle (PRT),
  - subordinated clause (SBAR),
  - unlike coordinated phrase (UCP),
  - verb phrase (VP)

Experimental Design

- An atomic phrase represents the most basic phrase with no nested sub-phrases.
- ((S (NP (NP Pierre Vinken) , (ADJP (NP 61 years) old) ,) (VP will (VP join (NP the board) (PP as (NP a nonexecutive director)) (NP Nov. 29))) .))

Experimental Design

- Used two state-of-the-art parsers
- Full parser by Michael Collins
  - Represents a full parse tree as a set of basic phrases and a set of dependency relationships between them
- SNoW-based CSCL shallow parser
  - A multi-class classifier that is specifically tailored for learning in domains in which the potential number of information sources taking part in decisions is very large

Experimental Design

Table 1: **Rankings of Shallow Parsers in CoNLL-2000.** See (Kim-Sang and Buchholz, 2000) for details.

Parsers	Precision(%)	Recall(%)	F_β(%)
[KM00]	93.45	93.51	93.48
[Hal00]	93.13	93.51	93.32
[CSCL]*	93.41	92.64	93.02
[TKS00]	94.04	91.00	92.50
[ZST00]	91.99	92.25	92.12
[Dej00]	91.87	91.31	92.09
[Koe00]	92.08	91.86	91.97
[Osb00]	91.65	92.23	91.94
[VB00]	91.05	92.03	91.54
[PMP00]	90.63	89.65	90.14
[Joh00]	86.24	88.25	87.23
[VD00]	88.82	82.91	85.76
Baseline	72.58	82.14	77.07

Experimental Design

- Testing on two types of data
  - Wall Street Journal for accuracy
  - Switchboard data for robustness
- The sections used for testing had not previously been used for training

Experimental Design

- The following sentence is a typical example of the SWB data.
- *Huh/UH ,/, well/UH ,/, um/UH ,/, you/PRP know/VBP ,/, I/PRP guess/VBP it/PRP 's/BES pretty/RB deep/JJ feelings/NNS ,/, uh/UH ,/, I/PRP just/RB ,/, uh/UH ,/, went/VBD back/RB and/CC rented/VBD ,/, uh/UH ,/, the/DT movie/NN ,/, what/WP is/VBZ it/PRP ,/, GOOD/JJ MORNING/NN VIET/NNP NAM/NNP ./.*

Experimental Design

The results are reported in terms of precision, recall, and $F_\beta(\beta = 1)$ as defined below:

$$\text{Precision} = \frac{\text{Number of correct proposed patterns}}{\text{Number of proposed patterns}}$$

$$\text{Recall} = \frac{\text{Number of correct proposed patterns}}{\text{Number of correct patterns}}$$

$$F_\beta = \frac{(\beta^2 + 1) \cdot \text{Recall} \cdot \text{Precision}}{\beta^2 \cdot \text{Precision} + \text{Recall}}$$

Experimental Results

Table 2: **Precision & Recall for phrase identification (chunking)** for the full and the shallow parser on the WSJ data. Results are shown for an (weighted) average of 11 types of phrases as well as for two of the most common phrases, NP and VP.

	Full Parser			Shallow Parser		
	P	R	F_β	P	R	F_β
Avr	91.71	92.21	91.96	93.85	95.45	94.64
NP	93.10	92.05	92.57	93.83	95.92	94.87
VP	86.00	90.42	88.15	95.50	95.05	95.28

Experimental Results

Table 3: **Precision & Recall for atomic phrase identification** on the WSJ data. Results are shown for an (weighted) average of 11 types of phrases as well as for the most common phrase, NP. VP occurs very infrequently as an atomic phrase.

	Full Parser			Shallow Parser		
	P	R	F_β	P	R	F_β
Avr	88.68	90.45	89.56	92.02	93.61	92.81
NP	91.86	92.16	92.01	93.54	95.88	94.70

Experimental Results

Table 4: **Switchboard data:** Precision & Recall for phrase identification (chunking) on the Switchboard data. Results are shown for an (weighted) average of 11 types of phrases as well as for two of the most common phrases, NP, VP.

	Full Parser			Shallow Parser		
	P	R	F_β	P	R	F_β
Avr	81.54	83.79	82.65	86.50	90.54	88.47
NP	88.29	88.96	88.62	90.50	92.59	91.54
VP	70.61	83.53	76.52	85.30	89.76	87.47

Experimental Results

Table 5: **Robustness:** Relative degradation in F_β results for Chunking. For each parser the result shown is the ratio between the result on the "noisy" SWB data and the "clean" WSJ corpus data.

	Full Parser	Shallow Parser
	F_β	F_β
Avr	.89	.93
NP	.95	.96
VP	.86	.92

Table 6: **An example: a parsing mistake**

WORD	POS	TRUE	Collins
Um	UH	B-INTJ	B-INTJ
COMMA	COMMA	O	I-INTJ
Mostly	RB	O	I-INTJ
COMMA	COMMA	O	O
um	UH	B-INTJ	B-INTJ
COMMA	COMMA	O	O
word	NN	B-NP	**B – NP**
processing	NN	I-NP	**B – VP**
applications	NNS	I-NP	**B – NP**
and	CC	O	O
COMMA	COMMA	O	O
uh	UH	B-INTJ	B-INTJ
COMMA	COMMA	O	O
just	RB	B-ADVP	B-PP
as	IN	B-PP	I-PP
a	DT	B-NP	B-NP
dumb	JJ	I-NP	I-NP
terminal	NN	I-NP	I-NP
..	.	O	O

Conclusion

- The authors Li and Roth conclude that, although their work is preliminary, their results show that the shallow parser, based on the specific task of identifying several kinds of phrases for which they trained it, performs more accurately and robustly than the full parser.

Thank you…

- Any questions?