On the Role of Style in Parsing Speech with Neural Models

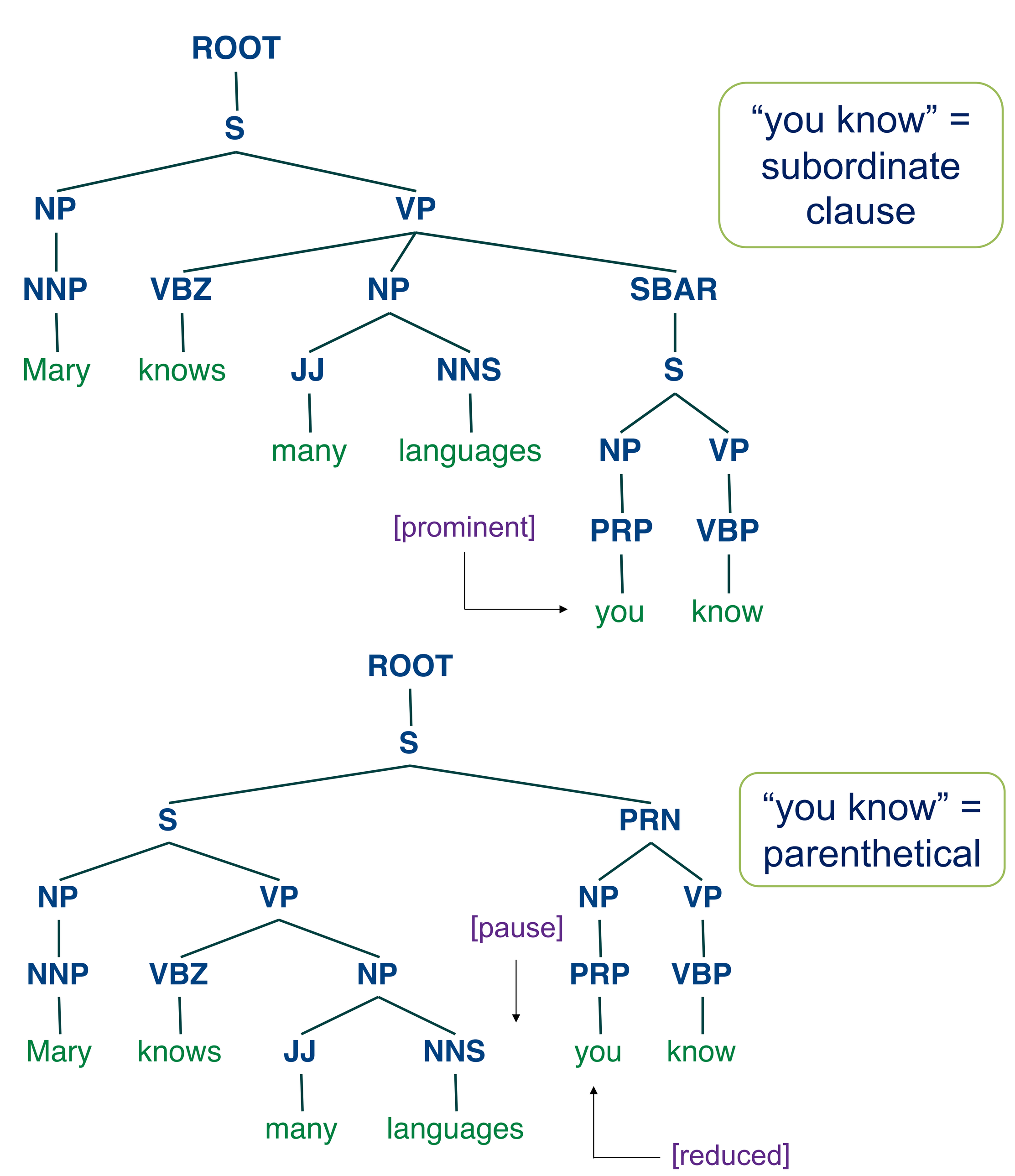
Trang Tran, Mari Ostendorf – University of Washington
Jiahong Yuan, Yang Liu – LAIX Inc.
ELECTRICAL & COMPUTER ENGINEERING
LAIX Inc.

Overview

- Parsing: core technology for intermediate language understanding
- Focus of parsing research & resources: written text
- Problem: many applications (dialog systems, assistive devices, translation, …) involve spoken language
- This work studies impact of **style** difference
  - Written text ≠ spontaneous speech (wording)
  - Spontaneous speech ≠ Read speech (prosody)

Background

- Parsing: identify syntactic structure
- Speech vs. text:
  - lacks conventional written cues (case, punctuations); has disfluent components
  - has prosody: characteristics beyond words; acoustic correlates (intonation, energy, timing) signal structure



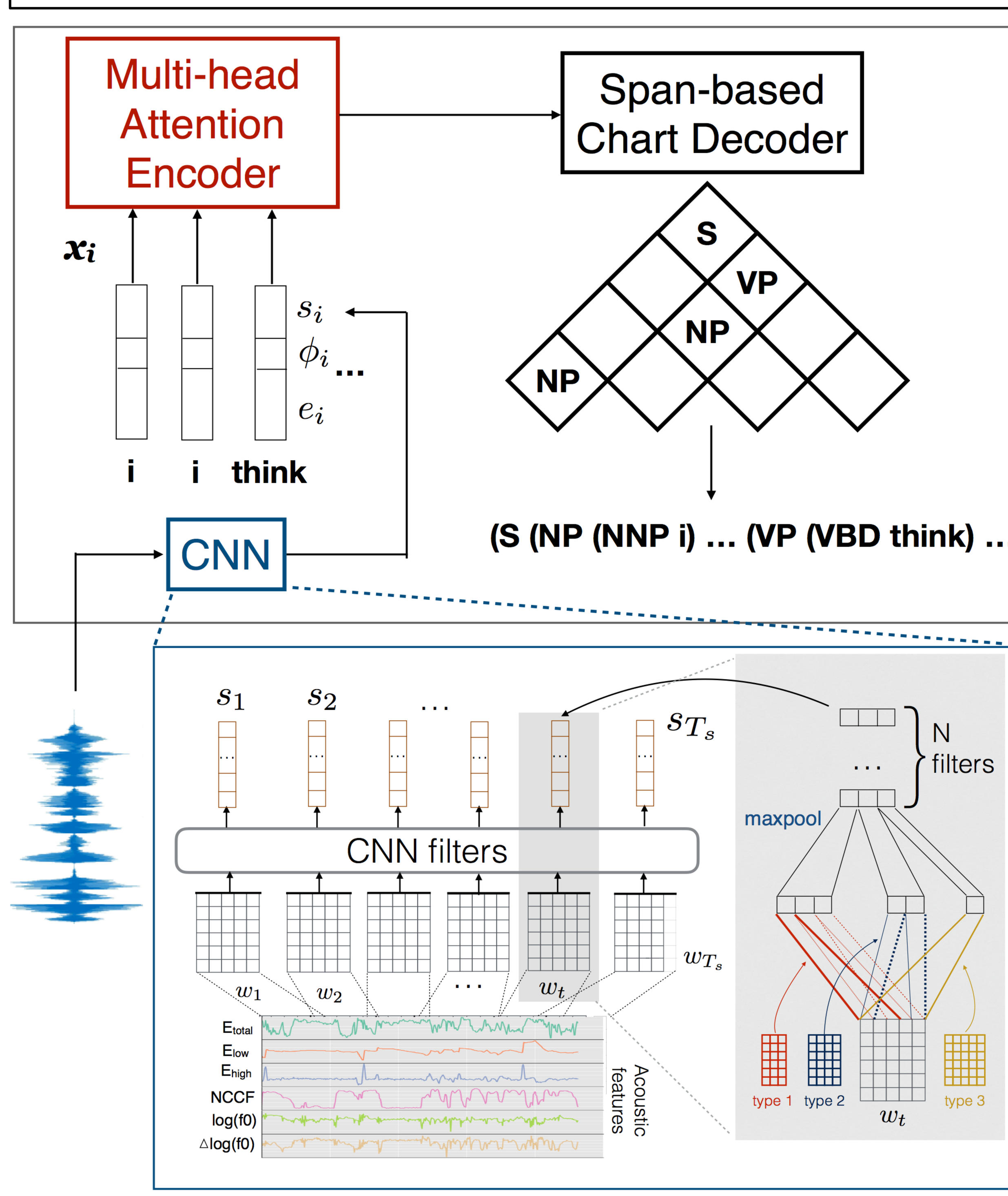
- Recent advances:
  - 2018: prosody benefits neural parsing on spontaneous speech
  - 2018, 2019: contextual embeddings give significant benefit in neural text parsers (SOTA on WSJ Treebank)

Questions

1. Do contextualized word representations learned for written text also benefit spontaneous speech parsers? [Yes!]
2. Does prosody improve further on top of the rich text information in neural parsers for spontaneous speech? [Yes!]
3. How is the use of prosody affected by mismatch between read and spontaneous speech styles? [Read on…]

Approach

- Input representation
  - word-level features $[x_1, x_2, \ldots]$
  - $\boldsymbol{x}_i = [\boldsymbol{e}_i, \boldsymbol{s}_i, \boldsymbol{\phi}_i]$
  - $\boldsymbol{e}_i$: word embeddings
  - $\boldsymbol{s}_i$: acoustic feature embeddings
  - $\boldsymbol{\phi}_i$: pause, duration features
- Output:
  - Set of labeled spans $[(a_i, b_i, l_i), \ldots]$
  - (a_i, b_i, l_i) = (start_idx, end_idx, label)
- Self-attentive encoder + chart decoder (self-attn) (Kitaev & Klein, 2018)
- Integrate prosody into via a convolutional neural network (CNN) (Tran et al., 2018)
- Metric: Parseval F1 (label and span)



Data

Data	Style	Available Material	Split	# Sentences	Used in
WSJ	news text	(gold) parses	train, dev	40k	Q1
SWBD	conversational speech (C)	audio, (gold) parses	train, dev, test	96k	Q1, Q2, Q3
CSR	read news (R)	audio, (silver) parses	train (tune), dev	8k	Q2, Q3
GT-N	read news/article (R)	audio, (gold) parses	test	6k (3k unique)	Q3
GT-SW	read version of SWBD (RC)	audio, (gold) parses	test	31 (13 unique)	Q3

Results

Q1

Train	Embedding	F1
WSJ (W)	BERT	77.5
SWBD (S)	Learned	91.0
	GloVe (Fisher)	91.0
	GloVe (Gword)	91.2
	ELMo	92.7
	BERT	93.2
S+W	BERT	93.4

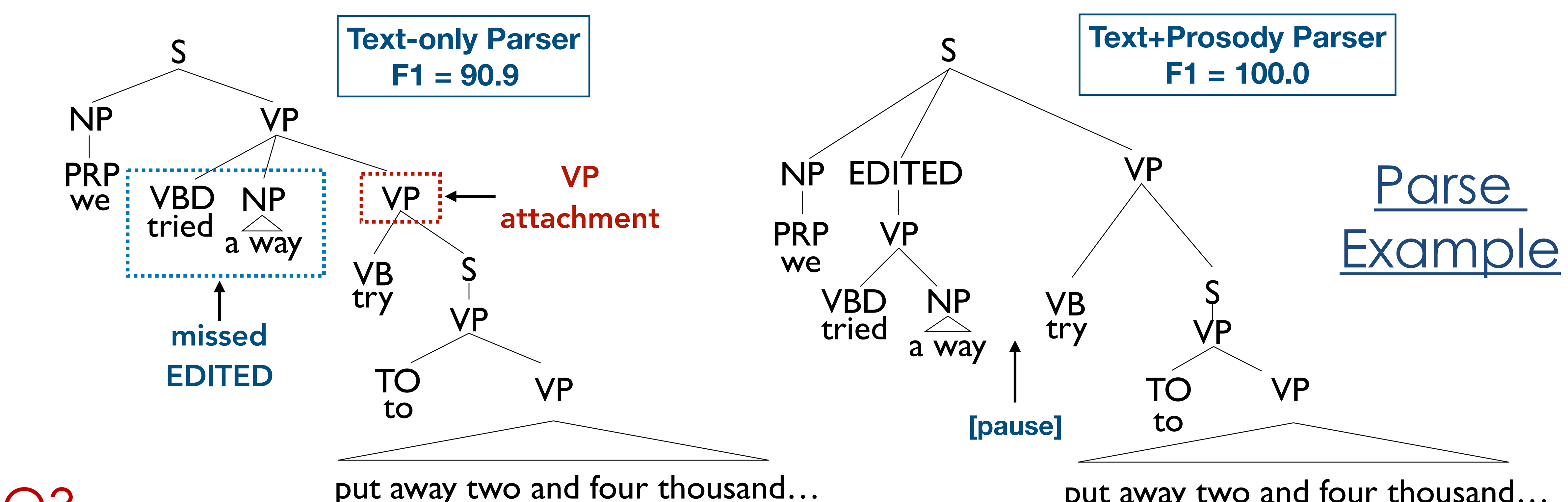
- Training with text alone doesn't work, even with BERT embeddings
- Pretraining on large written text benefits parsing speech
- Training on both (SWBD+WSJ) gives marginal gain

Q2

Model		all	disfluent	fluent
text	ELMo	92.5	91.5	94.6
	BERT	92.9	91.9	94.9
+prosody	ELMo	92.7*	91.7*	94.9*
	BERT	93.0*	92.1	95.2*

- SWBD test sentences: 3823 disfluent (with EDITED, INTJ), 2078 fluent
- (*): statistically significant at p<0.05
- Using prosody:
  - helps in disfluent and long sentences
  - further improves performance over strong text-only parsers: current best SWBD parsing result
  - reduces edit errors, 19% fewer VP attachment errors

Parse Example



Q3

Train/Tune	Model	SWBD (C)	GT-N (R)	GT-SW (RC)
SWBD (C)	text	92.9	92.4	98.0
CSR (R)	text	80.6	93.9	91.4
SWBD (C)	+prosody	93.0*	92.6*	98.0
CSR (R)	+prosody	80.4	94.2*	90.3

- Training on conversational (C) speech: minimal degradation on read (R) speech
- Training on (R): significant degradation on (C) → (C) more useful for general training
- Use of prosody differs in (R) vs. (C): style mismatch is both in terms of words and acoustic cues

Conclusion

- Pretrained contextualized word embeddings on text helps constituency parsing of speech
- Using prosody gives further gains, especially in long and disfluent sentences; reducing attachment errors
- Conversational prosody ≠ read prosody
  Conversational prosody is more general, better for training

- Acknowledgements: NSF Grant IIS-1617176; opinions our own
- Code: `github.com/trangham283/prosody_nlp/tree/master/code/self_attn_speech_parser`