Incorporating ASR Errors with Attention-based, Jointly Trained RNN for Intent Detection and Slot Filling

Raphael Schumann[1]   Pongtep Angkititrakul[2]

[1] UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Institute for Computational Linguistics
Heidelberg University

[2] BOSCH

Human Machine Interaction
Robert Bosch LLC

Outline

Outline

Spoken Language Understanding Pipeline

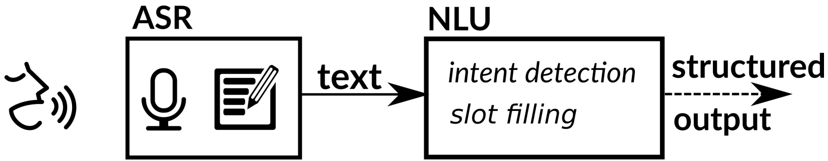


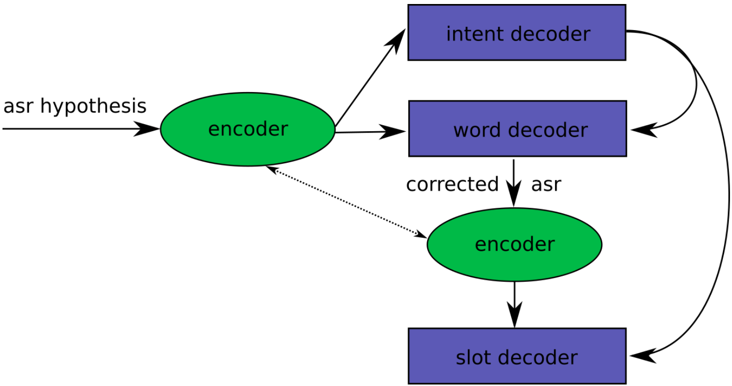
Figure: icons: [1]

- ASR errors get propagated to NLU component
- leverage information from intent detection and slot filling to correct ASR errors
- train as joint model

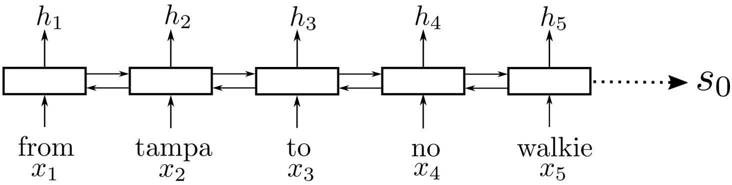
Outline

1. Introduction
2. Proposed Model
3. Data
4. Baseline
5. Evaluation Metrics
6. Results
7. Conclusion
8. Future Work

High Level Architecture

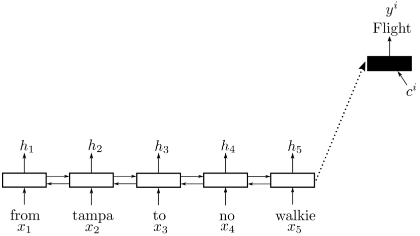


Encoder



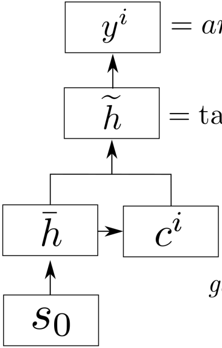
- bidirectional RNN with LSTM cell
- $h_t = [fh_t, bh_t]$ at each timestep $t = \{1, ..., T_x\}$
- encodes input sequence $\mathbf{x}$ to vector s_0 [2]:
  - $s_0 = tanh(W_s[fh_{T_x}, bh_1])$

Intent Decoder



- text classification on encoded input sequence $\mathbf{x}$
- intent attention vector c^i weighted sum over all h_t
- intent label y^i predicted by feed-forward network on $[c^i, s_0]$

Intent Decoder Detail



$$y^i = argmax(softmax(W_p \widetilde{h}_i))$$

$$\widetilde{h} = \tanh(W_c[c_i, \bar{h}_i])$$

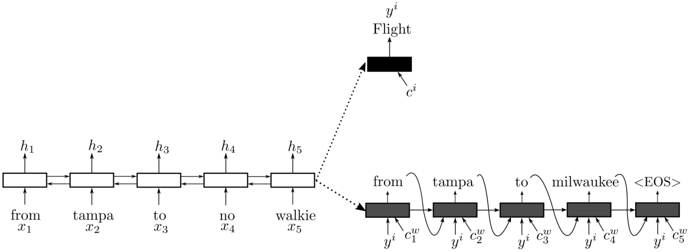
$$W_i s_0 = \bar{h}$$

$$c^i = \sum_{j=1}^{T_x} a_{ij} h_j$$

global attention [Luong et al., 2015]

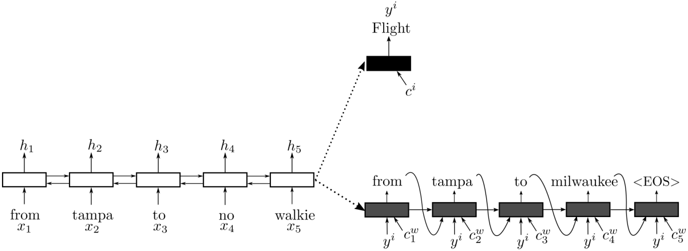
$$a_{ij} = \frac{\exp\left(\bar{h}_i^\top W_a h_j\right)}{\sum_{j'} \exp\left(\bar{h}_i^\top W_a h_{j'}\right)}$$

Corrected-Word Decoder



- RNN with LSTM cell
- initial state is set to s_0

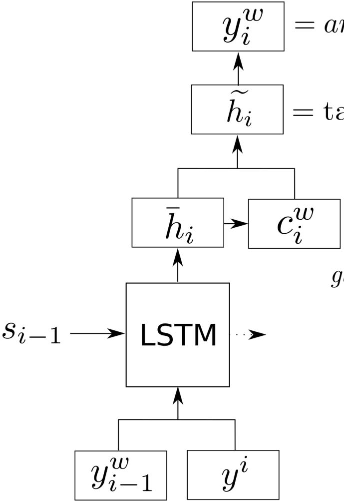
Corrected-Word Decoder



Input at each decoding timestep i:

- predicted intent label y^i
- attention vector c_i^w weighted sum over all h_t
- previous emitted corrected word y_{i-1}^w

Corrected-Word Decoder Detail



$$y_i^w = argmax(softmax(W_p \widetilde{h}_i))$$

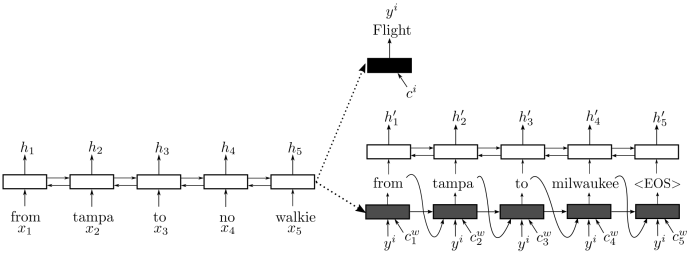
$$\widetilde{h}_i = \tanh(W_c[c_i, \bar{h}_i])$$

$$c_i^w = \sum_{j=1}^{T_x} a_{ij} h_j$$

global attention [Luong et al., 2015]

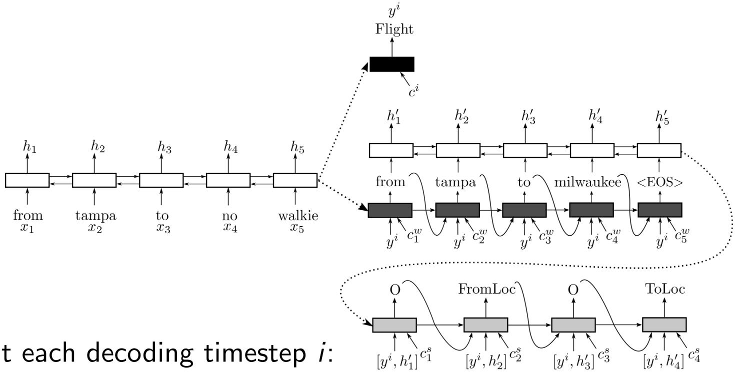
$$a_{ij} = \frac{\exp(\bar{h}_i^\top W_a h_j)}{\sum_{j'} \exp(\bar{h}_i^\top W_a h_{j'})}$$

Slot Decoder



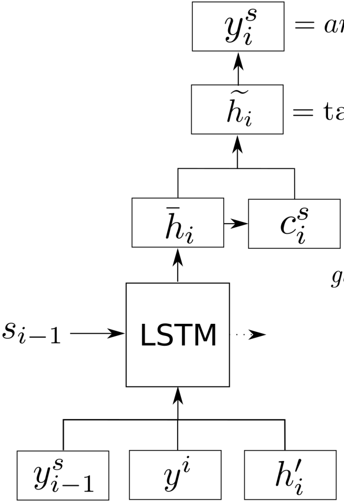
- slots are tagged on the corrected word sequence
- apply same encoder, resulting in hidden states h' and s'_0

Slot Decoder



Input at each decoding timestep i:

- predicted intent label y^i
- attention vector c^s_i weighted sum over all h'_t
- previous emitted slot token y^s_{i-1}
- corrected word encoder hidden state h'_i



$$y_i^s = argmax(softmax(W_p \widetilde{h}_i))$$

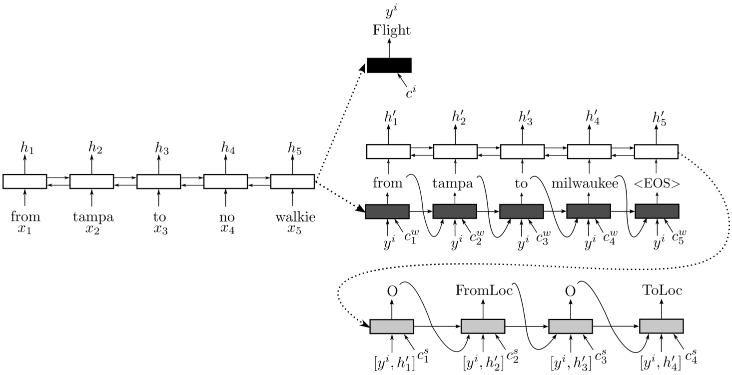
$$\widetilde{h}_i = \tanh(W_c[c_i, \bar{h}_i])$$

$$c_i^s = \textstyle\sum_{j=1}^{T_x} a_{ij} h_j$$

global attention [Luong et al., 2015]

$$a_{ij} = \frac{\exp(\bar{h}_i^\top W_a h_j)}{\sum_{j'} \exp(\bar{h}_i^\top W_a h_{j'})}$$

Joint Model



- use information about predicted intent during word correction
- shared word embeddings for all tasks
- weights shared between both encoders
- scheduled sampling [3] for corrected ASR sequence

Outline

ATIS

- Airline Travel Information Systems (ATIS) dataset [4]
- 18 different intent labels
- 128 different slot labels

ATIS Instance

Input:

words	show	me	flights	from	boston	to	new	york

Labels:

intent	flight							
slots	O	O	O	O	B-fromloc .city_name	O	B-toloc .city_name	I-toloc .city_name

Extended ATIS

- create audio samples by TTS
- add noise to reach ASR performance of $\sim$ 14% word error rate
- use top3 ASR hypotheses as input and form new instances

Extended ATIS Instance

Input:

words	show	flights	from	boston	to	no	work

Labels:

intent	flight							
words	show	me	flights	from	boston	to	new	york
slots	O	O	O	O	B-fromloc .city_name	O	B-toloc .city_name	I-toloc .city_name

Data

	train	dev	test	unique words
ATIS	4085	893	893	950
extended	11841	2583	2606	3178

Outline

Baseline

subsequent models:

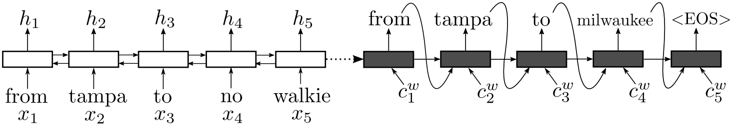


Figure: ASR Correction

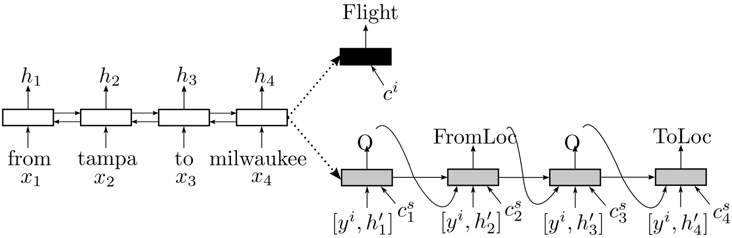


Figure: Intent Detection + Slot Filling [5]

Outline

Evaluation Metrics

- **WER:** word error rate
- **Slot F1:** F1-score following CoNLL Chunking Shared Task [6] using the in/out/begin schema [7]
- **Intent Error**: percentage of wrongly predicted intent labels

Outline

Results

Models	WER (%)	Slot (F1)	Intent Error (%)
Joint Slot&Detection	14.55	84.26	5.80
ASR Correction + Joint Slot&Detection	10.43	86.85	5.20
Proposed Joint Model	10.55	87.13	5.04

Table: Experimental results on the extended ATIS dataset.

- average of 10 runs
- joint model beats subsequent model

Outline

Conclusion

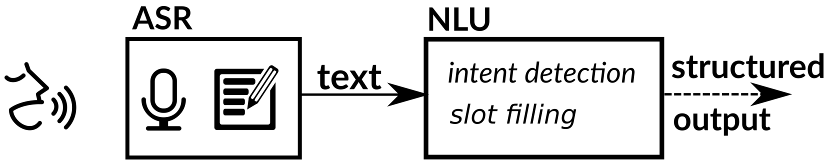
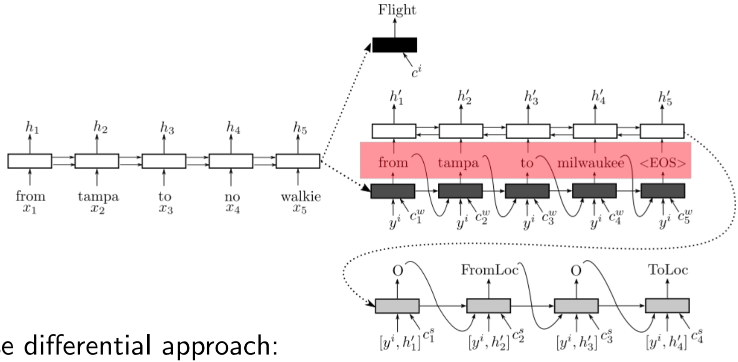


Figure: icons: [1]

- joint model for ASR error correction, intent detection and slot tagging performs better than subsequent models
- reducing the gap between ASR and NLU component

Outline



- use differential approach:
  - Gumbel Softmax [8]
  - Soft Argmax [9]

Bibliography I

[1] M. Aguilar, A. Shirazi, and S. Keating, *Voice, voice, write*,

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014. arXiv: 1409.0473. [Online]. Available: http://arxiv.org/abs/1409.0473.

[3] S. Bengio, O. Vinyals, N. Jaitly, and N. M. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems, NIPS*, 2015. [Online]. Available: http://arxiv.org/abs/1506.03099.

Bibliography II

[4] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The atis spoken language systems pilot corpus," in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT '90, Hidden Valley, Pennsylvania: Association for Computational Linguistics, 1990, pp. 96–101. DOI: 10.3115/116580.116613. [Online]. Available: https://doi.org/10.3115/116580.116613.

[5] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," *CoRR*, vol. abs/1609.01454, 2016. [Online]. Available: http://arxiv.org/abs/1609.01454.

Bibliography III

[6] E. F. Tjong Kim Sang and S. Buchholz, "Introduction to the conll-2000 shared task: Chunking," in *Proceedings of the 2Nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning - Volume 7*, ser. ConLL '00, Lisbon, Portugal: Association for Computational Linguistics, 2000, pp. 127–132. DOI: 10.3115/1117601.1117631. [Online]. Available: https://doi.org/10.3115/1117601.1117631.

[7] L. A. Ramshaw and M. P. Marcus, "Text chunking using transformation-based learning," *CoRR*, vol. cmp-lg/9505040, 1995. [Online]. Available: http://arxiv.org/abs/cmp-lg/9505040.

[8] E. Jang, S. Gu, and B. Poole, *Categorical reparameterization with gumbel-softmax*, cite arxiv:1611.01144, 2016. [Online]. Available: http://arxiv.org/abs/1611.01144.

Bibliography IV

[9] K. Goyal, C. Dyer, and T. Berg-Kirkpatrick, "Differentiable scheduled sampling for credit assignment," *CoRR*, vol. abs/1704.06970, 2017. arXiv: 1704.06970. [Online]. Available: http://arxiv.org/abs/1704.06970.