Sentence Simplification for SRL: A Neural Approach

Logan Born
Simon Fraser University
loborn@sfu.ca

1 Nov 2016

Intent

- To investigate the effectiveness of neural nets on SRL tasks.

Intent

- To investigate the effectiveness of neural nets on SRL tasks.
- Replicate the results of Vickrey & Koller 2008:

Intent

- To investigate the effectiveness of neural nets on SRL tasks.
- Replicate the results of Vickrey & Koller 2008:
  - Jointly learn to simplify input sentences and to label simplified sentences.

Vickrey & Koller 2008

- Combined system for sentence simplification and SRL.

Vickrey & Koller 2008

- Combined system for sentence simplification and SRL.
- Four stages:

Vickrey & Koller 2008

- Combined system for sentence simplification and SRL.
- Four stages:
  - Simplify: repeatedly apply transformation rules to the input parse to obtain a set of simpler parses.

Vickrey & Koller 2008

- Combined system for sentence simplification and SRL.
- Four stages:
  - Simplify: repeatedly apply transformation rules to the input parse to obtain a set of simpler parses.
  - Extract role patterns: what argument structures are possible for a given simplified parse?

Vickrey & Koller 2008

- Combined system for sentence simplification and SRL.
- Four stages:
  - Simplify: repeatedly apply transformation rules to the input parse to obtain a set of simpler parses.
  - Extract role patterns: what argument structures are possible for a given simplified parse?
    - Each simple sentence–role pattern pair defines a possible labeling.

Vickrey & Koller 2008

- Combined system for sentence simplification and SRL.
- Four stages:
  - Simplify: repeatedly apply transformation rules to the input parse to obtain a set of simpler parses.
  - Extract role patterns: what argument structures are possible for a given simplified parse?
    - Each simple sentence–role pattern pair defines a possible labeling.
  - Extract other features from the candidate labelings: POS tags, argument positions, and head words.

Vickrey & Koller 2008

- Combined system for sentence simplification and SRL.
- Four stages:
  - Simplify: repeatedly apply transformation rules to the input parse to obtain a set of simpler parses.
  - Extract role patterns: what argument structures are possible for a given simplified parse?
    - Each simple sentence–role pattern pair defines a possible labeling.
  - Extract other features from the candidate labelings: POS tags, argument positions, and head words.
  - Learn a set of weights for these features which assign high likelihood to good labelings.

Example

Input: "I was given a sandwich by John."

Example

Input: "I was given a sandwich by John."

Simplification: $\xrightarrow[\textit{Depassivize}]{}$ "John gave me a sandwich"

Example

Input:	"I was given a sandwich by John."
Simplification:	$\xrightarrow[\textit{Depassivize}]{}$ "John gave me a sandwich"
Role pattern:	{ARG0 = Subj NP, ARG1 = PV NP2, ARG2 = PV NP1}

Example

Input:	"I was given a sandwich by John."
Simplification:	$\xrightarrow[\textit{Depassivize}]{}$ "John gave me a sandwich"
Role pattern:	{ARG0 = Subj NP, ARG1 = PV NP2, ARG2 = PV NP1}
Feature set:	`Rule = Depassivize`
	`Pattern = {ARG0=Subj NP, ARG1=PV NP2, ARG2=PV NP1}`
	`Role = ARG0, Head Word = John`
	`Role = ARG1, Head Word = sandwich`
	`Role = ARG2, Head Word = I`
	`Role = ARG0, Category = NP`
	`Role = ARG1, Category = NP`
	`Role = ARG2, Category = NP`
	`Role = ARG0, Position = Subject NP`
	`Role = ARG1, Position = Postverb NP2`
	`Role = ARG2, Position = Postverb NP1`

Training Weights

- For each candidate simplification c_k of an input sentence c, construct a feature vector f_k (last slide).

Training Weights

- For each candidate simplification c_k of an input sentence c, construct a feature vector f_k (last slide).
- Define the probability of a candidate labeling as

$$P(c_k) = \frac{e^{w^\top f_k}}{\sum_{k'} e^{w^\top f_{k'}}}$$

Training Weights

- For each candidate simplification c_k of an input sentence c, construct a feature vector f_k (last slide).
- Define the probability of a candidate labeling as

$$P(c_k) = \frac{e^{w^\top f_k}}{\sum_{k'} e^{w^\top f_{k'}}}$$

- Objective function seeks to maximize total probability of all correct labelings:

$$F(w) = \Sigma_s \left(\log \Sigma_{c_k^s \in K^s} e^{w \cdot f_k^s} - \log \Sigma_{c_{k'}^s \in C^s} e^{w \cdot f_{k'}^s} \right) - \underbrace{\frac{w^\top w}{2\sigma^2}}_{\text{L2 regularization}}$$

Neural SRL

- Idea: generate simplified sentences as in Vickrey & Koller 2008, but train a neural net to select the best labeling.

Neural SRL

- Idea: generate simplified sentences as in Vickrey & Koller 2008, but train a neural net to select the best labeling.
- Sequence-to-sequence learning problem: given a sequence of simplified parses as input, output a sequence of role labels.

Neural SRL

- Idea: generate simplified sentences as in Vickrey & Koller 2008, but train a neural net to select the best labeling.
- Sequence-to-sequence learning problem: given a sequence of simplified parses as input, output a sequence of role labels.
  - Suitable for RNN encoder-decoder network.

Network Architecture

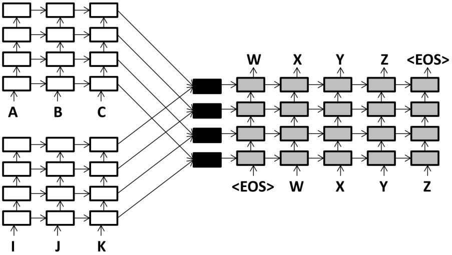
- Series of RNN encoders, one per simplified parse.

Network Architecture

- Series of RNN encoders, one per simplified parse.
- Following Zoph & Knight 2016, combine these encodings and pass them as inputs to an RNN decoder.

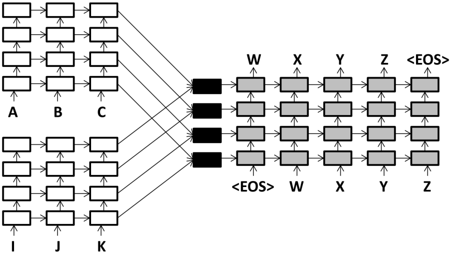
Network Architecture

- Series of RNN encoders, one per simplified parse.
- Following Zoph & Knight 2016, combine these encodings and pass them as inputs to an RNN decoder.
- Decoder will produce role labels as output.



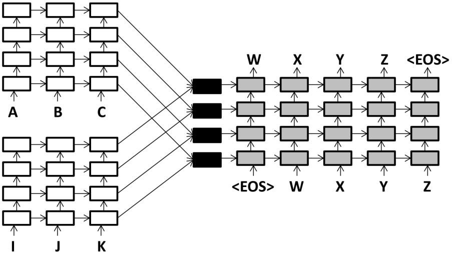
Network Architecture

- Series of RNN encoders, one per simplified parse.
- Following Zoph & Knight 2016, combine these encodings and pass them as inputs to an RNN decoder.
- Decoder will produce role labels as output.
  - Output sequence format?



Network Architecture

- Series of RNN encoders, one per simplified parse.
- Following Zoph & Knight 2016, combine these encodings and pass them as inputs to an RNN decoder.
- Decoder will produce role labels as output.
  - Output sequence format?



- Remark: no need to pass lexical features, role patterns, etc as inputs.

Evaluation

- Following Vickrey & Koller 2008, evaluate on the PropBank WSJ dataset, using the CoNLL 2005 evaluation metrics.

Evaluation

- Following Vickrey & Koller 2008, evaluate on the PropBank WSJ dataset, using the CoNLL 2005 evaluation metrics.
- Compare against Vickrey & Koller 2008 and previous SOTA (Punyakanok et al. 2008); also compare with recent neural SRL like FitzGerald et al. 2015.

Contributions

- Novel application of the multi-source encoder-decoder framework from Zoph & Knight 2016.

Contributions

- Novel application of the multi-source encoder-decoder framework from Zoph & Knight 2016.
- Novel approach to neural SRL: contrast FitzGerald et al. 2015 (no simplification), Zhou & Xu 2015 (ignores syntactic features).

References

Alberti, Chris, David Weiss, Greg Coppola, Slav Petrov. 2015. Improved Transition-Based Parsing and Tagging with Neural Networks. EMNLP 2015.

Firat, Orhan, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism. NAACL 2016.

Jia, Robin and Percy Liang. 2016. Data Recombination for Neural Semantic Parsing. ACL 2016

Vickrey, David and Daphne Koller. 2008. Sentence Simplification for Semantic Role Labeling. ACL 2008.

Vickrey, David. 2010. Learning Structured Probabilistic Models for Semantic Role Labeling. Dissertation, Stanford University.

Vinyals, Oriol, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. NIPS 2015.

Zhou, Jie and Wei Xu. 2015. End-to-end Learning of Semantic Role Labeling Using Recurrent Neural Networks. ACL 2015.

Zoph, Barret and Kevin Knight. 2016. Multi-Source Neural Translation. NAACL 2016.