Word Ordering Without Syntax

Allen Schmaltz Alexander M. Rush Stuart M. Shieber

Harvard University

EMNLP, 2016

Outline

1. Task: Word Ordering, or Linearization
2. Models
3. Experiments
4. Results

Word Ordering

- Task: Recover the original order of a shuffled sentence

Given a bag of words

{ the, ., Investors, move, welcomed }

$\Downarrow$

Goal is to recover the original sentence

Investors welcomed the move .

Word Ordering

- Task: Recover the original order of a shuffled sentence

Variant: Shuffle, retaining base noun phrases (BNPs)

{ the move, ., Investors, welcomed }

$\Downarrow$

Goal is to recover the original sentence

Investors welcomed the move .

Word Ordering

Early work

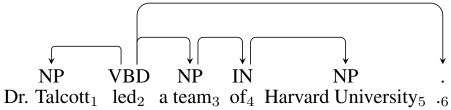
Jeffrey Elman ("Finding Structure in Time." *Cognitive Science*, 1990):

The order of words in sentences reflects a number of constraints. . . Syntactic structure, selective restrictions, subcategorization, and discourse considerations are among the many factors which join together to fix the order in which words occur. . . [T]here is an abstract structure which underlies the surface strings and it is this structure which provides a more insightful basis for understanding the constraints on word order. . . . It is, therefore, an interesting question to ask whether a network can learn any aspects of that underlying abstract structure.

The word ordering task also appears in Brown et al. (1990) and Brew (1992).

Word Ordering, Recent Work (Zhang and Clark, 2011; Liu et al., 2015; Liu and Zhang, 2015; Zhang and Clark, 2015)

- Liu et al. (2015) (known as ZGEN)
  - State of art on PTB
  - Uses a transition-based parser with beam search to construct a sentence and a parse tree



- Liu and Zhang (2015)
  - Claims syntactic models yield improvements over pure surface n-gram models
    - Particularly on longer sentences
    - Even when syntactic trees used in training are of low quality

Revisiting comparison between syntactic & surface-level models

Simple takeaway:

- Prior work: Jointly recovering explicit syntactic structure is important, or even required, for effectively recovering word order
- We find: Surface-level language models with a simple heuristic give much stronger results on this task

Models - Inference

- Scoring function:

$$f(x, y) = \sum_{n=1}^{N} \log p(x_{y(n)} \mid x_{y(1)}, \dots, x_{y(n-1)})$$
$$y^* = \arg\max_{y \in \mathcal{Y}} f(x, y)$$

- Beam search: Maintain multiple beams, as in stack decoding for phrase-based MT
- Include an estimate of future cost in order to improve search accuracy: Unigram cost of uncovered tokens in the bag

Beam Search ($K = 3$): Unigram Future Cost Example

Shuffled bag

{ the, ., Investors, move, welcomed }

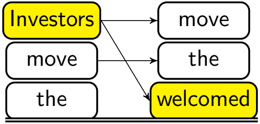
Investors
move
the

- Timestep 1:
  - score(Investors) $= \log p(\text{Investors} \mid \text{START}) + \log p(\text{the}) + \log p(.) + \log p(\text{move}) + \log p(\text{welcomed})$

Beam Search ($K = 3$): Unigram Future Cost Example

Shuffled bag

{ the, ., Investors, move, welcomed }

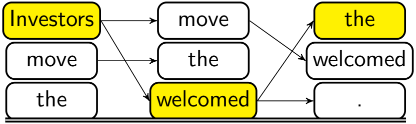


- Timestep 2

Beam Search ($K = 3$): Unigram Future Cost Example

Shuffled bag

{ the, ., Investors, move, welcomed }



- Timestep 3:
  - score(Investors welcomed the) $= \log p(\text{Investors} \mid \text{START}) + \log p(\text{welcomed} \mid \text{START, Investors}) + \log p(\text{the} \mid \text{START, Investors, welcomed}) + \log p(.) + \log p(\text{move})$

Experiments

- Data, matches past work:
  - PTB, standard splits, Liu et al. (2015)
  - PTB + Gigaword sample (GW), Liu and Zhang (2015)
  - WORDS and WORDS+BNPS tasks
- Baseline: Syntactic ZGEN model (Liu et al., 2015)
  - With/without POS tags
- Our LM models: NGRAM and LSTM
  - With/without unigram future costs
  - Varying beam size (64, 512)

Test Set Performance (BLEU), WORDS task

Model	BLEU
ZGEN-64	30.9
NGRAM-64 (NO FUTURE COST)	32.0
NGRAM-64	37.0
NGRAM-512	38.6
LSTM-64	40.5
LSTM-512	42.7

Test Set Performance (BLEU), Words+BNPs task

Model	BLEU
ZGen-64	49.4
ZGen-64+pos	50.8
NGram-64 (no future cost)	51.3
NGram-64	54.3
NGram-512	55.6
LSTM-64	60.9
LSTM-512	63.2
ZGen-64+lm+gw+pos	52.4
LSTM-64+gw	63.1
LSTM-512+gw	65.8

Performance by sentence length

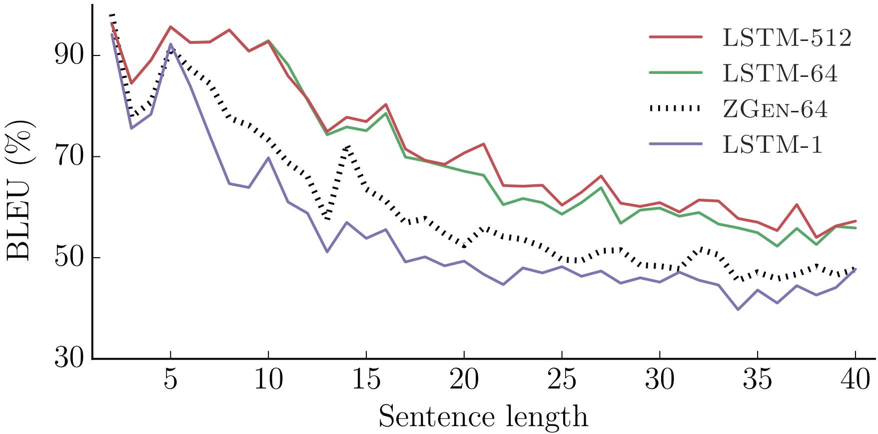


Figure: Performance on PTB validation by length (Words+BNPs models)

Additional Comparisons

BNP	g	GW	1	10	64	128	256	512
					LSTM			
•			41.7	53.6	58.0	59.1	60.0	60.6
•	•		47.6	59.4	62.2	62.9	63.6	64.3
•	•	•	48.4	60.1	64.2	64.9	65.6	66.2
			15.4	26.8	33.8	35.3	36.5	38.0
	•		25.0	36.8	40.7	41.7	42.0	42.5
	•	•	23.8	35.5	40.7	41.7	42.9	43.7
					NGRAM			
•			40.6	49.7	52.6	53.2	54.0	54.7
•	•		45.7	53.6	55.6	56.2	56.6	56.6
			14.6	27.1	32.6	33.8	35.1	35.8
	•		27.1	34.6	37.5	38.1	38.4	38.7

Conclusion

- Strong surface-level language models recover word order more accurately than the models trained with explicit syntactic annotations
- LSTM LMs with a simple future cost heuristic are particularly effective

Conclusion

- Strong surface-level language models recover word order more accurately than the models trained with explicit syntactic annotations
- LSTM LMs with a simple future cost heuristic are particularly effective

- Implications
  - Begin to question the utility of costly syntactic annotations in generation models (e.g., grammar correction)
  - Part of larger discussion as to whether LSTMs, themselves, are capturing syntactic phenomena

Code

Replication code is available at
`https://github.com/allenschmaltz/word_ordering`