Prescient Language Models: Multitask Learning for Long-term Planning in LSTM's

Joyce Xu

Problem

Where do language models go wrong? E.g. in generation:

- Repetition
- Truncation
- Generic, unspecific outputs

In each case, appears to be a failure in the language model's capacity to "plan" well for future content.

Hypothesis:

1) If a language model indeed "plans" for the future, some representation of downstream content should be extractable from the LSTM's hidden states
2) "Attending" to future content in the form of making predictions can serve as an auxiliary task to a robust, multitask language model

Data and Tasks

Dataset: Wikitext-2 for eval + base language model training
2-mil subset of Wikitext-103 for future module training

Standalone Prediction Tasks (from hidden state):

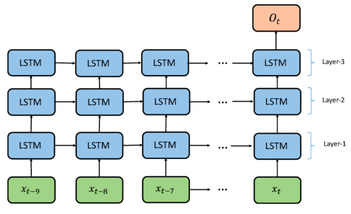
1. How well can we predict +1, +2, +3, etc. words ahead?
2. How well can we predict an *average word embedding* of the next 1, 3, 10 words?

Language Modeling with Multitask Learning:

1. Train LSTM's with an auxiliary "future prediction" loss?
2. Train LSTM's with an auxiliary "future prediction" loss AND incorporate the predicted future vector back in to the model?

Approach: Loss Formulation

1. Train stacked LSTM base language model
2. Train future prediction module from hidden states of an LSTM layer
3. (Optionally) feed in future prediction to final layer of modified LM
4. Sum losses: LM + future



Multitask with "+n" single-word prediction:
Auxiliary loss = **cross-entropy** loss with +n target word

$$l_{+n}(\theta_{LM}, \theta_F) = -\frac{1}{T}\sum_{t=1}^{T} \log(p_{x_t}|x_{t-c}, ..., x_{t-1}; \theta_{LM}) + \lambda \log(q_{x_{t+n}}|x_{t-c}, ..., x_{t-1}; \theta_F)$$

Multitask with average future "bag-of-words" embedding prediction:
Auxiliary loss = **cosine similarity** with gold future embedding

$$\hat{x}_{bow} = \text{FutureModel}(x_{t-c}, ..., x_{t-1}; \theta_F)$$

$$l_{BOW}(\theta_{LM}, \theta_F) = -\frac{1}{T}\sum_{t=1}^{T} \log(p_{x_t}|x_{t-c}, ..., x_{t-1}; \theta_{LM}) + \lambda(1 - \cos\_\text{sim}(x_{bow}, \hat{x}_{bow}))$$

Approach: Model Architecture

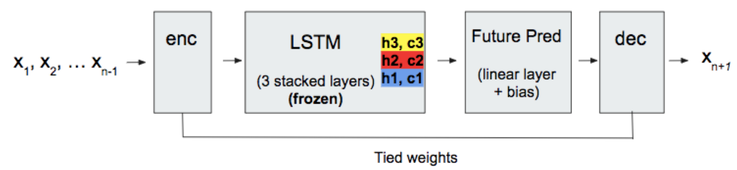


Fig 1. Standalone "+1" prediction: freeze base LSTM weights, train future prediction module to predict "n+1" word from one of the 3 LSTM hidden state layers

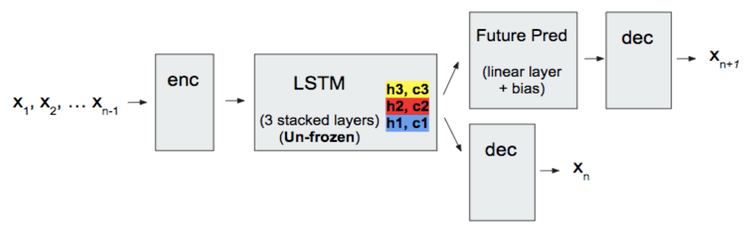


Fig 2. Multitask language model A: un-freeze base LSTM weights, fine-tune with original LM loss and auxiliary "+1" future prediction loss

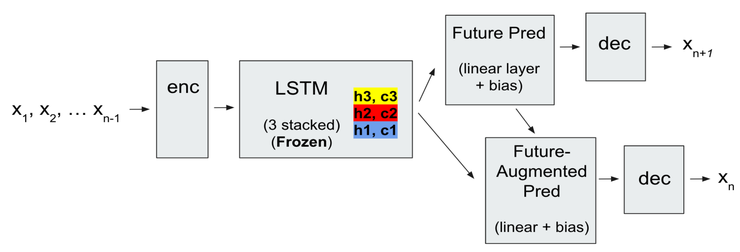


Fig 3. Multitask language model B: keep base LSTM weights frozen, feed predicted future vector *and* LSTM hidden states to augmented prediction module

Results

+n	Perplexity
1	243.67
2	418.58
3	529.24

Table 1. Standalone "+n" future predictions

Window	Cosine sim
1	0.251
3	0.516
10	0.712

Table 2. Standalone "BOW" future word embedding predictions

Model	Perplexity
Base LM	71.73
Base LM + fine-tuning	69.51
Multitask A (n1h2): aux loss	69.25
Multitask B (rand): feed-in	66.6
Multitask B (n1h2): aux loss + feed-in	**65.6**

Table 3. Language modeling perplexities for base LM and multitask A and B

Analysis

Standalone future predictions:

- Best perplexities achieved using h2: **long-term planning occurs *before* next-word prediction** in stacked LSTM's
- Consistently outperform other baselines (unigram; weighted word embed.)
- "+1" able to learn part of speech, structures such as lists, numbers/units: **LSTM's do form rough syntactic plan for at least a few upcoming words**

gold context | gold x_n x_{n+1} x_{n+2}	Base LM preds (top 5)	Multitask LM preds (top 5)	+1 Future preds (top 10)
and parts of the Black Sea . It is closely | related to the	found, related, distributed, classified, associated	related, distributed, found, considered, endemic	to, in, ,, the, from, and, that, by, with, about
animals . <eos> The first pair of <unk> is armed | with a large	with, in, ,, by, .	with, in, by, at, on	the, a, <unk>, four, two, three, with, The, which, and
wrestlers <unk> chose . <unk> 's four wrestlers were James | Storm , Matt	<unk>, ,, and, Hardy, Moore	<unk>, ,, ", and, R.	,, and, <unk>, ., who, 's, (, from, in, Williams
<unk> chose . <unk> 's four wrestlers were James Storm | , Matt Morgan	,, and, (, ., who	,, and, (, ., <unk>	<unk>, and, the, ,, who, a, but, The, Williams, David
chose . <unk> 's four wrestlers were James Storm , | Matt Morgan ,	who, <unk>, the, a, Kevin	<unk>, who, the, a, and	<unk>, and, ,, the, of, was, Ryan, had, @-@, Michael
. <unk> 's four wrestlers were James Storm , Matt | Morgan , Robert	<unk>, and, ,, Lee, Hardy	<unk>, ,, Hardy, and, Lee	,, and, <unk>, (, from, @-@, of, 's, on, was
to 60 centimetres (24 in) and weighing up | to 5 -	to, in, and, with, by	to, with, by, at, and	4, 2, 1, 60, 150, 230, 0, 8, 12, 70
60 centimetres (24 in) and weighing up to | 5 - 6	2, 1, 3, 5, 12	2, 5, 3, 4, 12	@.@, cm, mm, %, centimetres, feet, @,@, meters, inches, kilograms

Multitask LM:

- 1 point perplexity gain over control – good but not groundbreaking
- **Rare that future predictions specific or informative enough to change LM's original word choice**

gold context | gold x_t x_{t+1} x_{t+2}	Base LM preds (top 5)	Multitask LM preds (top 5)	+1 Future preds (top 10)
was bombed by the Japanese for the first time on | 26 January 1942	the, 20, 23, 18, 24	20, 18, 1, 19, 24	November, April, February, January, December, October, June, March, May, August

Conclusion

- Well-trained LSTM's *do* form syntactic plan for upcoming few words
- More salient information about future content in hidden states than in input embeddings
- "Planning" happens in earlier layers than next-word prediction
- Predicting future words does not usually change base LM behavior

References

[1] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. 2017.
[2] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. An Analysis of Neural Language Modeling at Multiple Scales. arXiv preprint arXiv:1803.08240, 2018.
[3] Trieu H. Trinh, Andrew M. Dai, Thang Luong, and Quoc V. Le. Learning longer-term dependencies in rnns with auxiliary losses. CoRR, abs/1803.00144, 2018.