The Importance of Being Recurrent for Modeling Hierarchical Structure

Summarizing and contextualizing this high-level AI research paper into byte-sized pieces

Nathaniel Watkins
Dec 12 · 7 min read

Photo by Nathaniel Shuman on Unsplash

This is a summary of

> *The Importance of Being Recurrent for Modeling Hierarchical Structure*

by Ke Tran, Arianna Bisazza & Christof Monz found here:

aclweb.org/anthology/D18-1503

· · ·

Two Sentence Takeaway

Recurrent Neural Networks (RNNs), such as Long Short-Term Memory networks (LSTMs), currently have performance limitations, while newer methods such as Fully Attentional Networks (FANs) show potential for replacing LSTMs without those same limitations. So the authors set out to compare the two approaches using standardized methods and found that LSTMs universally surpass FANs in prediction accuracy when applied to the hierarchy structure of language.

. . .

RNNs have inherent performance limitations

For a while, it seemed that RNN's were taking the Natural Language Processing (NLP) world by storm (from about 2014–17). However, we've recently started realizing the limitations of RNN's, primarily that they are "inefficient and not scalable". While there is great promise in overcoming these limitations by using more specialized processing hardware, such as Field Programmable Gate Arrays, solutions are at least one hardware generation away. This means that it's worth exploring other options, such as Convolutional Neural Networks (CNN) or Transformers, for text comprehension to see if we can achieve similar or better results using another technique that is more optimal with the current status quo in hardware.

Comparing the performance of an LSTM vs a FAN transformer

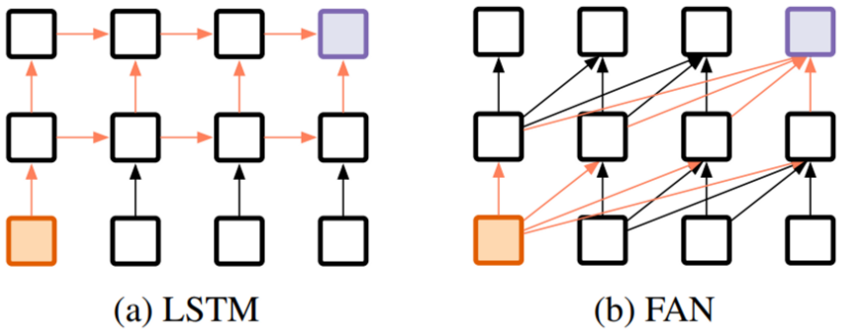
Due to this need, the authors of this paper have elected to benchmark test two promising methods of Natural Language Understanding (NLU), comparing the results between the two using objective criteria. Specifically, they gauged how well the models seemed to understand the hierarchical nature of language by testing relationships between subject and verb as well as testing the models on logical inference tasks.

Recurrent Neural Networks and more specifically Long Short-Term Memory networks are the gold-standard when it comes to NLP/NLU. So the authors start there as the baseline, then compare it to a Fully Attentional Network: a new model architecture from the cutting edge paper *Attention is all you need*. An LSTM is a sequential framework that takes inputs in one at a time, such as reading a sentence one word at a time; it varies from other RNNs by each node remembering dependencies for a longer period of time. See this post by Rohith Gandhi for an explanation of RNNs and some popular variants.

While LSTMs have a longer term memory than other RNNs, hence the Long in LSTM, they still struggle when there is a large distance between relevant data points such as the subject and the verb in the next sentence. Furthermore, LSTMs, due to their directional nature and consistent tweaking of their cell state at each time step, tend to have trouble when the context of an earlier portion of a sentence is dependant on information in a later part of a sentence. An example from Rohith's post above:

> *"He said, Teddy bears are on sale" and "He said, Teddy Roosevelt was a great President". In the above two sentences, when we are looking at the word "Teddy" and the previous two words "He said", we might not be able to understand if the sentence refers to the President or Teddy bears*

Enter the FAN transformer, which solves these problems by looking at the entire input, such as the entire sentence, at once, instead of sequentially, and it features an attention layer that helps preserve the context between relevant data points, no matter the distance. On top of these gains, the FAN architecture is highly parallelizable, which helps it overcome or avoid the aforementioned performance limitations of RNNs.



The difference in the architectures can be seen by the way information flows through the nodes (indicated with orange arrows). This graph was originally included in the subject paper.

Currently, the state of the art in sentence-embedding models is a bidirectional LSTM (bi-LSTM) with an attention layer, which was published after this subject paper was published, but bi-LSTMs and attention layers have been well developed by the release of this paper. Bidirectional LSTMs are basically 2 LSTMs (one reading the text left to right, and the other reading right to left) that compare notes to make a collective prediction. An attention layer is similar to described above for a FAN, but in an RNN, it sits outside of the sequential portions of the model allowing much more context to be preserved between time-steps. The authors chose to use a plain LSTM, without either of these upgrades (which address all the previously mentioned downsides, except for parallelization), but that turned out not to matter much, as the LSTM still achieved better accuracy than the FAN transformer.

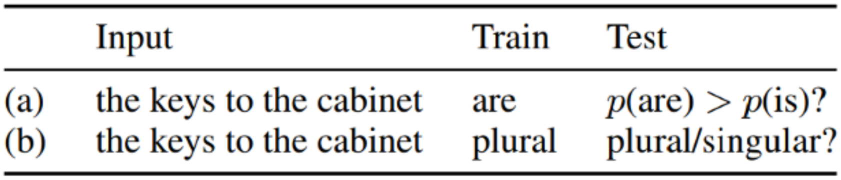
Why Hierarchy?

While the *Attention is all you need* paper focused on general language to language translation performance, the authors of this paper chose to study the models' comprehension of hierarchy in language. Hierarchy is critical to truly understanding the meaning of a sentence and is a necessary step to achieving anything near human-level NLU. Here are some examples highlighting the difficulty in understanding hierarchy, even for humans, and how it changes the meaning of sentences, from a talk on this paper presented by Rachel Tatman:

> *"I saw the person with the binoculars"*

> *I put the keys on the stand, on the table, by the couch, next to the desk…"*

Testing subject/verb agreement

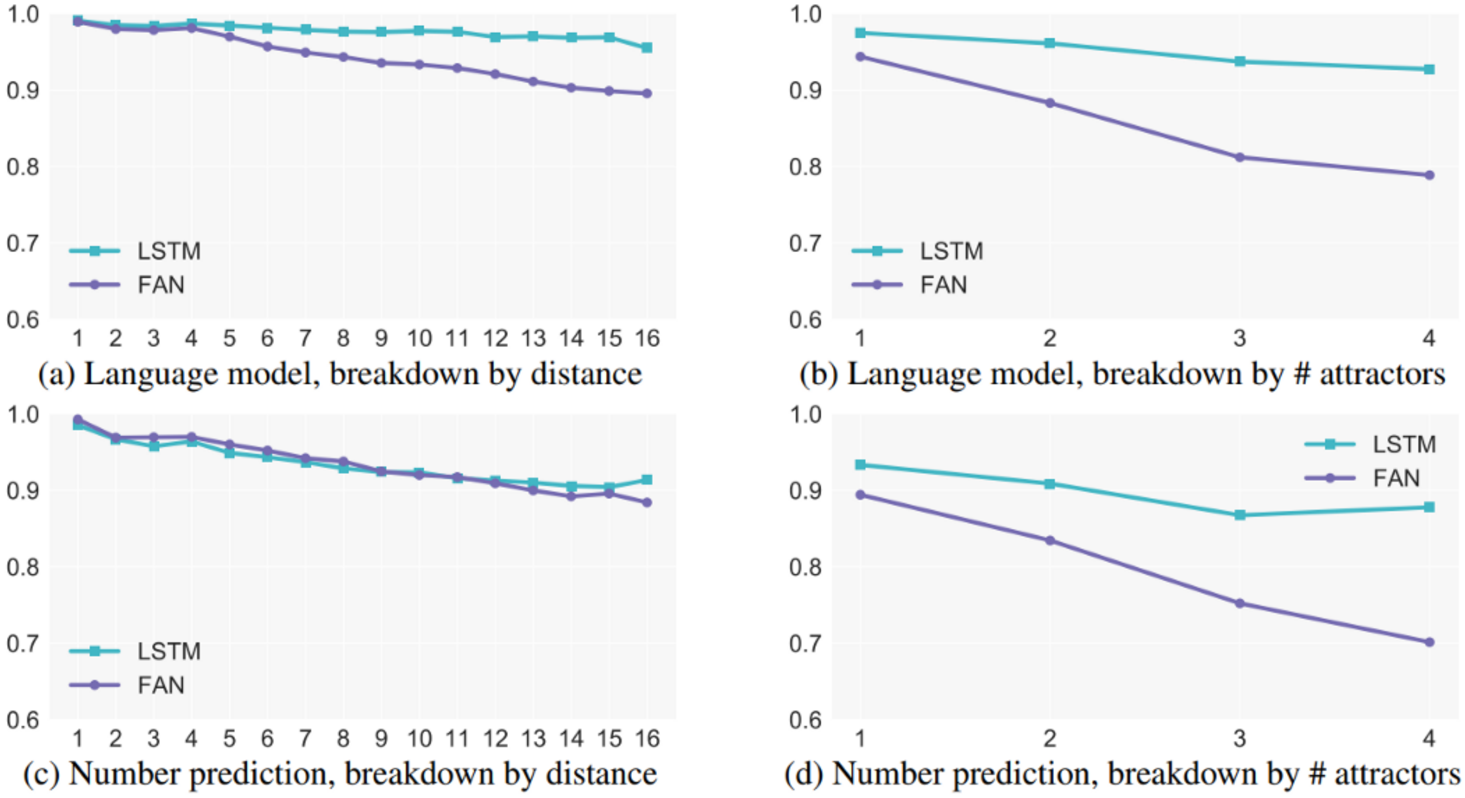


	Input	Train	Test
(a)	the keys to the cabinet	are	$p(\text{are}) > p(\text{is})$?
(b)	the keys to the cabinet	plural	plural/singular?

This diagram was originally included in the subject paper.

One big way to ensure that a model is understanding hierarchy is to make sure that it predicts an appropriate singular/plural verb given a singular/plural subject. The chart above shows an example sentence, including the input and how the verb plurality might be used in training the model and how it might be used in testing the prediction accuracy.

For this task, the LSTM outperformed the FAN in 3 of the tests and tied on the 3rd objective.



These graphs were originally included in the subject paper.

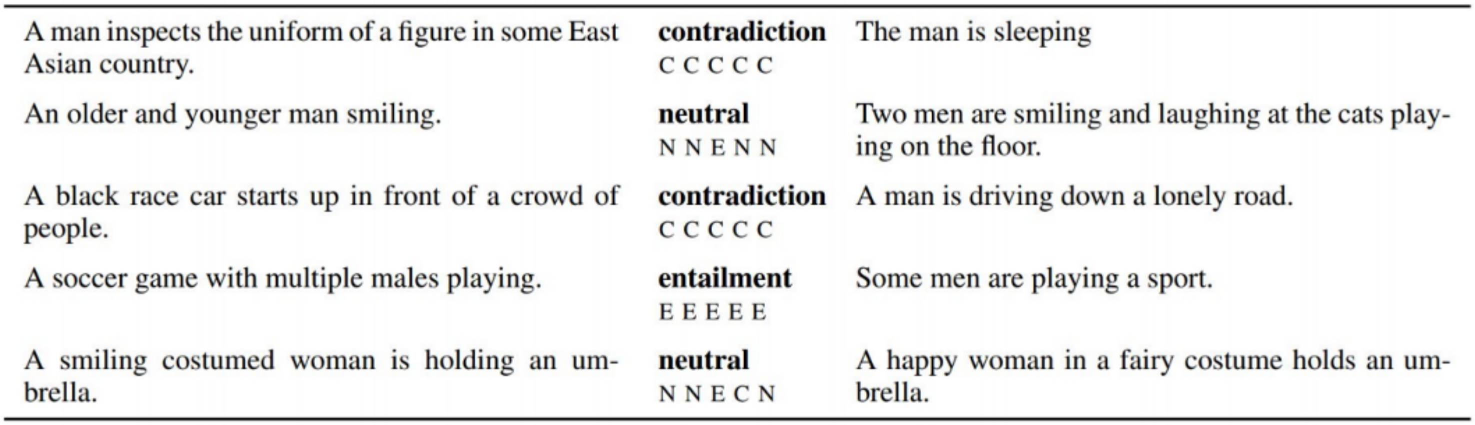
Note that the "distance" mentioned above is the number of words in between the subject and the verb. And the "attractors" are the number of nouns between the subject and verb, which might throw off the model's understanding of what word is the subject. Another example by Rachel:

> *The* ***bus*** *always* ***comes*** *late.* | *Attractors = 0*

> *The* ***bus*** *with broken* windows *always* ***comes*** *late.* | *Attractors = 1*

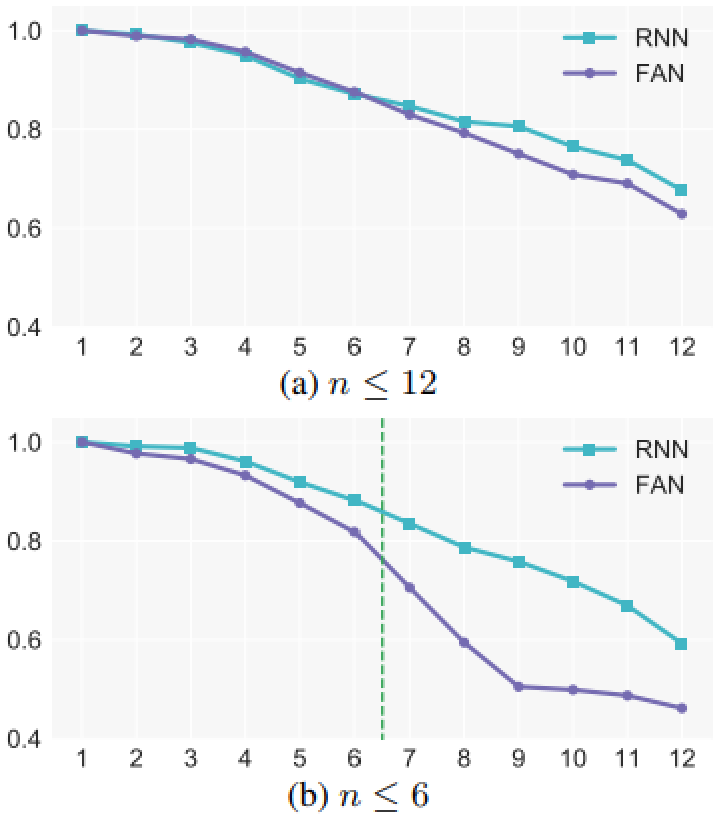
Testing logical inference

To avoid getting too hung up on the subtleties and variations that using sample text might introduce, the authors utilized a simplified language from *Bowman et al. (2015b)* using just six word types, three logic operators and some symbols to perform this task. They generated the training and test dataset using a rule-based system to ensure 100% accuracy of the data. While the letters and symbols in the paper might not seem to make much sense, this example should help:



A man inspects the uniform of a figure in some East Asian country.	**contradiction** C C C C C	The man is sleeping
An older and younger man smiling.	**neutral** N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	**contradiction** C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	**entailment** E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	**neutral** N N E C N	A happy woman in a fairy costume holds an umbrella.

These examples were originally included in Bowman et al (2015b).



(a) $n \leq 12$

(b) $n \leq 6$

These graphs were originally included in the subject paper.

And once again, we see the plain vanilla LSTM match or exceed the FAN on all accounts. Note that *n* is the number of operators on the left side of the equations.

So why is the LSTM a clear winner?

The paper doesn't venture to answer or even explore why these results were observed. They do a fantastic job explaining their process and detailing the hyperparameters used for repeatability, but they seem to purposefully avoid taking it further than pure empiricism. Furthermore, their code is clean, readable and documented so that you can try it all yourself: https://github.com/ketranm/fan_vs_rnn

Some theories that might explain these results:

- LSTMs are a heavily refined model architecture, with many years of research, while FAN transformers are about a year old right now and still on the bleeding edge of research. So perhaps the hyperparameters chosen for the grid search were outside the optimal range for a FAN on this task.

- Perhaps the sequential nature of an LSTM inherently is more in-tune with the sequential nature of human language. After all, we speak and write in a sequential manner.
- Indeed FANs might just be more suited to everything that goes into tasks like language to language translation, while LSTMs are better at understanding the structure in language.

Some interesting opportunities to explore these results further could include tuning the transformer (such as trying a weighted transformer and/or experimenting with more hyperparameters) or adding a 3rd architecture for comparison, such as Convs2S a Convolutional Neural Network based framework.

In theory, a Fully Attentional Network transformer seems like it should outperform a simple Long Short-Term Memory network on all accounts, but through thorough testing, this doesn't seem to be the case. While we should continue researching and tweaking FANs to explore this new option, we shouldn't discount the venerable LSTM just yet. Please read the paper for more detail.

· · ·

If this type of thing interests you, and you're in the Seattle area, I highly encourage you to join the Puget Sound Programming Python (PuPPy) Meetup for our monthly Advanced Topics on Machine Learning (AToM) discussion night, which inspired me to write this (Thanks to Rachel Tatman who recently gave a great talk on this paper). And if you're not in the area, I recommend looking for, or even starting, something similar wherever you are.

· · ·

This type of article is a new thing I'm going to be doing, born out of an apparent lack of approachable content that can quickly inform anyone about the status of cutting-edge AI research, with some background to see it within context. So I look forward to hearing any feedback or questions you have on this article or the topics discussed, either in the responses here or on social media. Feel free to connect with me:

twitter.com/theNathanielW

linkedin.com/in/theNathanielWatkins