What kind of grammar do LSTMs learn?

Thesis Project Proposal - Year 2

Ludovica Pannitto

Thu, Mar 26th 2019

34th cycle PhD student - CIMeC - Center for Mind/Brain Sciences

Do RNNs learn grammar?

A popular question, relating to **productivity** and **compositionality**[1]. Can machines master these fundamental traits of natural language?

How come such a simple architecture, fed with unrealistic input, with no access to perceptual information or hard-coded syntax can learn such a fundamental part of language?[2]

[1]"Linguistic generalization and compositionality in modern artificial neural networks" (Baroni 2020)

[2]"Colorless green recurrent networks dream hierarchically" (Gulordava et al. 2018), "The Emergence of Number and Syntax Units in LSTM Language Models" (Lakretz et al. 2019)

Our Setup

How much language (L) can be learnt from a certain level of computational complexity (C) with a certain type of data (I)?

$$C \times I \xrightarrow{f} L \tag{1}$$

All aspects of the equation are of paramount importance in linguistic discussion:

complexity of the learning mechanism C - how much has to be *innate* or *hard-coded* in the function?

quality and quantity of the stimuli I - how do stimuli differ? What are the most relevant features?

language L - what is the *grammar* that best explains the bits of language we experience?

Simplistically, we could say (child, child-directed-input) $\mapsto \ell_c$

What is innate?

Concerning what needs to be innate in order for language learning to happen, very different claims have been made:

Chomsky[3]: *the language faculty contains innate knowledge of various linguistic rules, constraints and principles [...] This knowledge is essential to our ability to speak and understand a language.*

Chomsky and Katz[4]: *rationalists and empiricists alike attribute innate structure and principles to the mind [...] purely combinatorial devices for putting together items of experience*

Fillmore[5]: *Some frames are undoubtedly innate, in the sense that they appear naturally and unavoidably in the cognitive development of every human [...] Others are learned through experience or training*

[3] "Innateness and Language" (Cowie 2017)
[4] "On innateness: A reply to Cooper" (Chomsky and Katz 1975)
[5] "Frames and the semantics of understanding" (Fillmore 1985)

What it takes to learn (a) language

Innatist theories have posited the existence of a specific ability for processing of hierarchical structures[6], while cognitive theories have stressed how linguistic hierarchies can emerge from the linear signal through general-purpose memory and cognitive mechanisms[7].

The **recurrent structure** of LSTMs has been shown to play a crucial role in the abstraction process[8].

[6] "The faculty of language: what is it, who has it, and how did it evolve?" (Hauser, Chomsky, and Fitch 2002)

[7] Christiansen and Chater 2016; Cornish et al. 2017; Lewkowicz, Schmuckler, and Mangalindan 2018

[8] "The Importance of Being Recurrent for Modeling Hierarchical Structure" (Tran, Bisazza, and Monz 2018), "Recurrent Memory Networks for Language Modeling" (Tran, Bisazza, and Monz 2016)

Research Questions

$$(\text{LSTM}, \{I_i\}) \overset{f}{\mapsto} \ell \tag{2}$$

- we fix the level of computational complexity to a vanilla LSTM (character-based)
- we explore different sources of input in a specific range $\{I_i\}$ selected based on their complexity level
- we want to explore the features of the produced language $\ell \in L$

Our questions are the following:

RQ1: *How much grammar is learned overall by the system?*

RQ2: *What is the influence of the complexity of the input on the learning process?*

Hypotheses

The network is only a (sophisticated) mechanism to find patterns in the data, with no bias towards the syntactic structure of sentences.

If the network is able to *abstract* some grammatical knowledge from raw data, then:

H1 [incrementality]: the learning process must be incremental and hierarchical

H2 [categories]: the structures learned can be described through data-driven categories

Data and Methods

Child-motivated input

We've collected a portion of existing corpora, with specific attention at developmental language.

CHILDES - Child-directed utterances of the NA and UK portions of the CHILDES database.

Gutenberg - Books and newspapers from 18 children-related bookshelves of Project Gutenberg (incl. literature, instructional books and others).

Opensubtitles - Movie and TV series subtitles from the OpenSubtitle corpus, filtered on the content-rating label (G for movies and TV-Y, TV-Y7. TV-G for tv series).

Simplewikipedia - 2019 dump of Simple English Wikipedia, written in basic and learning English.

Processing and Stats

stats	childes	opensub	simplewiki	gutenberg
$\|C\|$	14.1M	7.6M	23.4M	85.6M
$\|V\|$	53K	119K	451.8K	338.4K
#sentences	3M	1M	1.5M	4.2M
$\|C\|$/#sent	4.7	7.6	15.6	20.3
TTR	.0038	.016	.019	.004
HTR [:6M]	.002	.006	.014	.005

Next steps:

- Compute some indexes of syntactic complexity (i.e., *average dependency length*, *number of subordinate structures*...)
- Comparison of the subcorpora[9] based on perplexity and distributional information (i.e., *representation of words motivated by language acquisition literature*)

[9] including the corpus used in Gulordava et al. 2018

Which grammar?

We want to compare:

$$\ell_H \overset{?}{\rightleftharpoons} \ell_{LSTM} \tag{3}$$

We need a representation structure that lets us compare the subset of human language ($\ell_H \subseteq L_H$) and that produced or conceptualized by the LSTM ($\ell_{LSTM} \subseteq L_{LSTM}$).

We are actually comparing:

$$G(\ell_H) \overset{?}{\rightleftharpoons} G(\ell_{LSTM}) \tag{4}$$

for some grammar G, through its specific set of categories and assumptions.

Strings, catenae, constituents...

We take into consideration dependency-structure representation of syntactic trees.

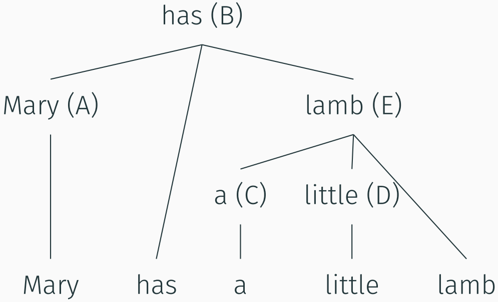
Fundamental units that describe how elements relate in the structure are[10]:

String: word or combination of words that is continuous with respect to **precedence**

Catena: word or combination of words that is continuous with respect to **dominance**

Constituent: **catena** that consists of a word plus all the words that that word dominates

[10] "Catenae: Introducing a novel unit of syntactic analysis" (Osborne, Putnam, and Groß 2012)



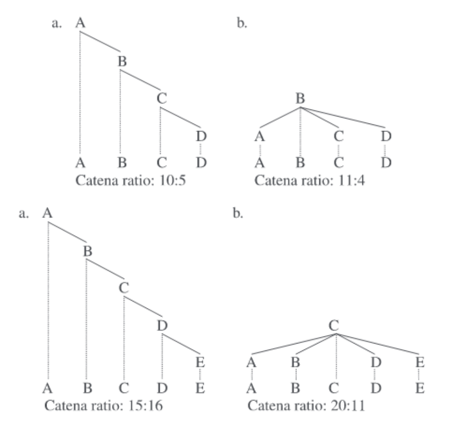
Strings: n^2 - (A, AB, ABC, … B, BC, … E)

Catenae: (A, B, C, D, E, AB, ABCE, ABDE, ABCDE, ABE, BCE, BDE, BE, CE, DE, CDE)

Constituents: n - (**A**, A**B**CDE, **C**, **D**, CD**E**)

Relation to structure[11]

The number of catenae depends on the structure of the tree.



[11]Trees from “Catenae: Introducing a novel unit of syntactic analysis” (Osborne, Putnam, and Groß 2012)

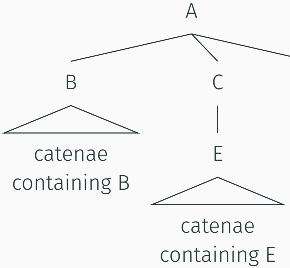
Strings (Chunks) have been central in collocation-based approaches to language modeling: it is unclear how to scale to the acquisition of discontinuous patterns.

Constituents seem to have too strong constraints (i.e., also *adjacency* is among them), and are not suited to model many phenomena (e.g., *idioms*, *ellipsis*...)

Catenae are based on a more inclusive definition than constituents, and show a number of interesting properties for capturing both linear and hierachical relations.

Extracting catenae

Extraction Algorithm: sketch



1. If the node is a leaf (i.e., D), the only possible catena is D itself
2. If the node is not a leaf (i.e., A), then it is the mother of subtrees (i.e., rooted in B, C and D). So (1) there can't be a catena that bridges any set of subtrees without A being in it and (2) any catena including A must also include a subset of his children nodes (i.e., a catena including A and E without including C is impossible).

Extraction Algorithm: pseudocode

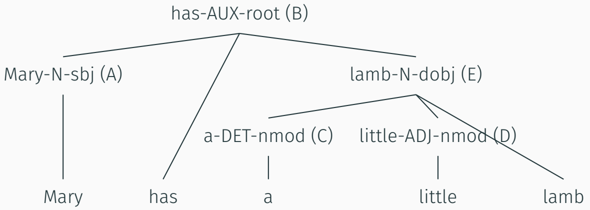
```
function extract_catenae (A: tree) ->
    returns (catenae_containing_A, catenae_in_this_tree):

    if A is leaf:
        return [new Catena(A)], [new Catena(A)]
    else:
        all_catenae = []
        C_catenae = [ [new Catena(A)] ]

        for child in A.children:
            c, all = extract_catenae(child)
            all_catenae.extend(all)
            C_catenae.append(c + EmptyCatena)

        A_catenae = []
        for comb in cartesian_product(C_catenae):
            A_catenae.append (new Catena(comb))

        return A_catenae, A_catenae+all_catenae
```



Catenae:

AB: Mary has, Mary AUX, N has, N AUX, Mary root, sbj has, N root, sbj AUX, sbj root

CDE: a little lamb, DET little lamb, a ADJ lamb, a little N, DET ADJ lamb ..., nmod ADJ lamb, DET nmod lamb...

ABE: Mary has lamb, Mary AUX lamb, Mary has N ..., sbj AUX N, ... Mary root N, ...

CHILDES

@nsubj @root, _PRON @root, @nsubj _VERB, _PRON _VERB, _VERB @obj, _AUX @root, **@root @obj**, _VERB _NOUN, **@nsubj _VERB @obj**, _DET _NOUN, @root _NOUN, @det _NOUN, **_PRON _VERB @obj**, _AUX _VERB, **you @root**, @aux _VERB, **you _VERB**, _PRON _AUX @root, **_VERB _PRON**, @nsubj _AUX @root, **@nsubj @root @obj**, @aux @root, **_PRON @root @obj**, **@advmod @root**, @nsubj _VERB _NOUN, **_PRON _NOUN**, **@root _PRON**, **_VERB _VERB**, **@root _VERB**, **_PRON _VERB _NOUN**

OPENSUBTITLES

@nsubj @root, _DET _NOUN, @det _NOUN, @nsubj _VERB, _AUX @root, **_NOUN _NOUN**, _VERB _NOUN, _PRON @root, _PRON _VERB, @root _NOUN, **@case _NOUN**, **_ADP _NOUN**, _AUX _VERB, @nsubj _AUX @root, _VERB @obj, @aux _VERB, **the _NOUN**, @aux @root, **_ADJ _NOUN**, _PRON _AUX @root, **_NOUN @root**, **@amod _NOUN**, **_NOUN _VERB**, **@nsubj _AUX _VERB**, **@nsubj @aux _VERB**, **@nsubj _VERB _NOUN**, **@case @obl**, **_ADP @obl**, **@nsubj @root _NOUN**

Extraction Algorithm: needs some tweaking

- longer catenae shouldn't be penalized $\rightarrow$

  promote longer ones? consider top k for each class of lengths?

- catenae with lexemes should be preferred over more the more abstract ones (i.e., f(*the dog*) = f(*DET N*)) $\rightarrow$

  introduce a penalty for PoS or syntactic relations?

- information shouldn't be replicated (i.e., f(*the dog barks*) = f(*the dog*))$\rightarrow$

  remove subcatenae with comparable frequency?

- frequency is probably not the most informative measure $\rightarrow$

  Mutual Information?

- how to compare the sets of extracted catenae? $\rightarrow$

  Size of overlap? Jaccard measure? Edit Distance? Average Precision?

RQ1: How much grammar is learned overall by the system?

H1 (incrementality)

Exp1: The **quantity** of learned structures grows with training

Given $s_1, s_2, ..., s_i, .., s_n$ steps during training, we should find:

$$|G(\ell_{s_1})| \leq |G(\ell_{s_2})| \leq ... \leq |G(\ell_{s_i})| \leq ... \leq |G(\ell_{s_n})| \tag{5}$$

Exp2: The **quality** of learned structures changes with training

Given $s_1, s_2, ..., s_i, .., s_n$ steps during training and $G : G_L \cup G_C$ with $G_L(\ell)$ being the structures composed mostly by lexemes and $G_C(\ell)$ being the structures composed mostly by grammatical categories, we should find:

$$|G_L(\ell_{s_1})| \geq |G_L(\ell_{s_2})| \geq ... \geq |G_L(\ell_{s_i})| \geq ... \geq |G_L(\ell_{s_n})| \tag{6}$$

$$|G_C(\ell_{s_1})| \leq |G_C(\ell_{s_2})| \leq ... \leq |G_C(\ell_{s_i})| \leq ... \leq |G_C(\ell_{s_n})| \tag{7}$$

H2 (categories)

Exp3: The distributional properties of structures at timestep t help explaining the distribution at timestep $t + j$.

As in **Exp2**, we can assume $G(\ell) = G_L(\ell) \cup G_C(\ell)$. The expectation is that, given two time steps $(i, j, i < j)$ during learning, the distributional properties observes on lexical patterns on ℓ_i (i.e., in $G_L(\ell_i)$) will be partially transferred on more abstract patterns in ℓ_j (i.e., in $G_C(\ell_j)$).

Given the structures $x_l \in G_L(\ell)$, $x_c \in G_C(\ell)$ and a measure of distributional similarity $\phi : G \times G \rightarrow [0, 1]$, we expect that $\phi(x_l, x_c)$ decreases over time steps.

$$\phi_i(x_l, x_c) \leq \phi_j(x_l, x_c) \tag{8}$$

According to Goldberg[12], this is for instance the case of the emergence of the ditransitive pattern *Sbj V Obj Obj2*, which is highly associated with *give* in child-directed speech: the idea is that part of the distributional meaning of *give* remains attached to the pattern *Subj V Obj Obj2*, that should therefore show a similar distributional behaviour once abstracted.

We expect that the distributional properties of the pattern *Sbj V Obj Obj2* will change during learning: in the frist phases its properties will overlap with those of a lower level pattern such as *Sbj give Obj Obj2* and it will gradually shift to a more autonomous distribution.

[12] *Constructions at work: The nature of generalization in language* (Goldberg 2006)

RQ2: What is the influence of the complexity of the input on the learning process?

H1 (incrementality)

Exp4: Abstraction is faster if the input is given with progressive levels of complexities

Given a family of input data $I_m = \iota_1, .., \iota_m$ and a measure of complexity $c : I \rightarrow \mathbb{R}$, we expect:

$$G(\ell_i) \subseteq G(\ell_j) \iff c(\iota_i) \leq c(\iota_j) \tag{9}$$

In other words, we expect that, if learning happens through a defined order of complexity, the network learns at step i some structures that will be needed at step $i+1$, thus optimizing learning. If the input is not ordered, the network might learn some structure that will no longer be needed afterwards and therefore be forgotten, leading to a non-optimized learning curve.

Evaluation

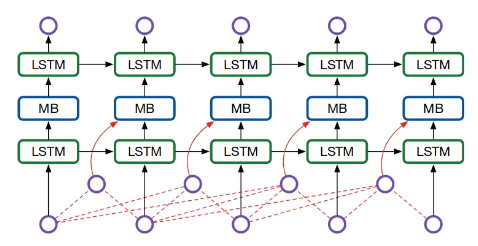
For any kind of input, at any point in the training process, we want to be able to assess how much grammatical competence has been acquired by the network.

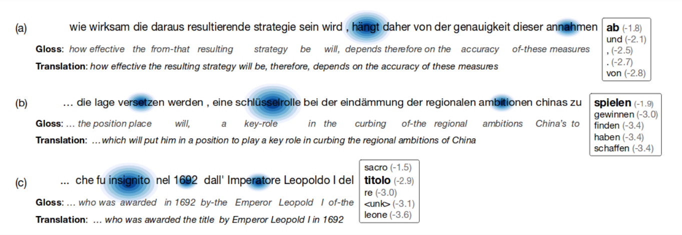
More specifically:

- what is the network focusing on →
  - **Attention over the hidden states of the network**[13]
- what does the acquired grammar look like →
  - **Overlap between sets of structures**
  - **Distributional properties of structures**

[13] "Recurrent Memory Networks for Language Modeling" (Tran, Bisazza, and Monz 2016)

Tran, Bisazza, and Monz 2016: Attention over hidden states





Overlap between sets of structures

Given some training data and some generated output from the RNN, processed with the same formalism, we can investigate which structures the RNN can reproduce, and compare their distribution to the original data.

Actual CHILDES data

"What kind of food did you buy?"

```
        buy
  ______|________
 |      |        kind
 |      |    ____|____
 |      |   |         of
 |      |   |         |
did    you what      food
```

RNN-generated data

"What kind of little girl?"

```
      kind
  ____|_____
 |          of
 |          |
 |         girl
 |          |
what      little
```

The trees show some catenae in common, specifically:

what,W,nmod - kind,N,root - of,I,nmod

what,W,nmod - kind,N,root - of,I,nmod - ∅,N,pmod

We can do this check even when the RNN produces partially nonsensical sentences.

Actual CHILDES data

“Are you teasing me?”, “It’s a steering wheel.”

```
      teasing
   _____|_____
are    you     me
```

```
       's
   ___|____
  |      wheel
  |    ____|______
 it   a        steering
```

RNN-generated data

It’s a jelly graving me.

```
      graving
  _____|______
's   jelly    |
|      |      |
it     a      me
```

Common Cateane:

∅,∅,sbj - ∅,V,root - me,PR,dobj

it,PR,sbj - is,V,∅

Distributional Properties

Given a piece of language ℓ and the corresponding set of catenae $G(\ell)$, we can define a distributional model over the structures contained in $G(\ell)$, considering $targets = contexts = G(\ell)$ and each sentence as the context window.

We can either:

- extend the construction of a simple count-based model
- employ a prediction algorithm for the construction of word embedding that allows to use arbitrary context features[14]

[14] "Dependency-based word embeddings" (Levy and Goldberg 2014)

Progress

So far...

			Experiments
Data	☑	Collection	1,2,3,4
	☑	Basic stats	4
	☐	Syntactic complexity stats	4
	☐	Comparison of subcorpora	2,3,4
Catenae	☑	Extraction	1,2,3,4
	☐	Correction for length/abstractness	1,3,4
	☐	Association measure	3
	☐	Overlap	1,2,3,4
LSTM	☐	Text processing	1,2,4
	☐	Text generation and processing	1,2,3,4
	☐	Attention	1,2,4
	☐	Time steps	1,2,3
DM	☐	Count-based model	3
	☐	Prediction model	3
	☐	Comparison among different spaces	3

References

Baroni, Marco (2020). "Linguistic generalization and compositionality in modern artificial neural networks". In: *Philosophical Transactions of the Royal Society B* 375.1791, p. 20190307.

Chomsky, Noam and Jerrold J Katz (1975). "On innateness: A reply to Cooper". In: *The Philosophical Review* 84.1, pp. 70–87.

Christiansen, Morten H and Nick Chater (2016). "The now-or-never bottleneck: A fundamental constraint on language". In: *Behavioral and brain sciences* 39.

Cornish, Hannah et al. (2017). "Sequence memory constraints give rise to language-like structure through iterated learning". In: *PloS one* 12.1.

Cowie, Fiona (2017). "Innateness and Language". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Fall 2017. Metaphysics Research Lab, Stanford University.

Fillmore, Charles J (1985). "Frames and the semantics of understanding". In: *Quaderni di semantica* 6.2, pp. 222–254.

Goldberg, Adele E (2006). *Constructions at work: The nature of generalization in language*. Oxford University Press on Demand.

Gulordava, Kristina et al. (2018). "Colorless green recurrent networks dream hierarchically". In: *Proceedings of NAACL-HLT*, pp. 1195–1205.

Hauser, Marc D, Noam Chomsky, and W Tecumseh Fitch (2002). "The faculty of language: what is it, who has it, and how did it evolve?" In: *science* 298.5598, pp. 1569–1579.

Lakretz, Yair et al. (2019). "The Emergence of Number and Syntax Units in LSTM Language Models". In: *Proceedings of NAACL-HLT*, pp. 11–20.

Levy, Omer and Yoav Goldberg (2014). “Dependency-based word embeddings”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 302–308.

Lewkowicz, David J, Mark A Schmuckler, and Diane MJ Mangalindan (2018). “Learning of hierarchical serial patterns emerges in infancy”. In: *Developmental psychobiology* 60.3, pp. 243–255.

Osborne, Timothy, Michael Putnam, and Thomas Groß (2012). “Catenae: Introducing a novel unit of syntactic analysis”. In: *Syntax* 15.4, pp. 354–396.

Tran, Ke M, Arianna Bisazza, and Christof Monz (2016). “Recurrent Memory Networks for Language Modeling”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 321–331.

— (2018). “The Importance of Being Recurrent for Modeling Hierarchical Structure”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4731–4736.