LT2213-VT20

Seminar assignment (VG part)
Semantic Compositionality in Deep Learning

Mohamed Zarzoura

December 2, 2020

SUMMARY: This report tries to touch the semantic compositionality in neural networks (NN) by examining [1, 2]. The first article discusses how compositional representation is computed, therefore it is mathematically oriented. It discusses first the compositional semantics in vector space between two constituents (binary function). After that, it discusses the mathematical side of NNs architecture, taking into consideration the compositional semantics in vector space. I try to write a summary showing how the NNs can be seen as an extension to the binary compositional functions.

The second article reviews past studies that examine the ability of NNs to learn the syntactic structure. It concludes that NNS goes beyond the simple heuristic learning but fail to be compositional. Furthermore, these failures are not systematic and hard to be analysised. Still the strategies that NNs use to generalize have not been revealed yet.

AN INTERESTED REMARK: In the second paper, the author criticizes the trials to introduce the tree structure, which linguists preferred to the NN. The author provides two reasons:

1. He claims that tree-NN did not provide a good performance and referred to an old article [3]. However, some studies claim the opposite like [4].
2. The author stated that recursive-NNs drive tree structures that are different from those introduced by linguists[5]. This finding may indicate that NNs do not have to generalize or learn in the same way that we expect.

The second reason seems to be convincing; there is no need to inject priors to the NNs design. Moreover, researchers found that during human brain processing of the natural language, neural signals are correlated to the semantics (meaning), not structure[6]. This last finding introduces a question: how can humans judging the correctness of sentences even if they have no meaning?

1 Compositional Semantics in Vector Space: Binary composition

In the vector space model, one can compress a word's semantic meaning into a real-valued vector, named word vector, using its context information. The question now is how to construct a similar numerical representation for a complex semantic structure like a phrase, sentence, paragraph, etc. Here the principle of compositionality comes to answer this question. The principle of compositionality states that a complex unit's meaning is a function only of two things: first, the semantic meanings of its constituents, and second the syntactic rules that combine those constituents[7]. However, [8] states that [9] suggests that knowledge about the language and the real world contributes to the linguistic unit's meaning.

There are multiple semantic vector space models[10], however in this short introduction we will focus on the additive model introduced in [11]. [11] formulates the semantic composition as follows:

$$p = f(u, v, R, K), \tag{1}$$

where u, v denote the constituents vector representation, R denotes the syntactic rules combine both u and v, K denotes the background knowledge, f denotes the semantic composition function and p denotes the compositional representation. In the additive model, the f is a vectors addition of u and v. [11] ignored the K terms to investigate what the model can achieve in the absence of background knowledge. Furthermore, [11] fixed the relation term R to study the verb-subject relation, Thus, the equation 1 will be as follow:

$$p = u + v \tag{2}$$

However, as we can see, equation 2 does not account for the word order as $u \cdot v = v \cdot u$. To overcome this problem, equation 2 can be rewritten as follow[1]:

$$p = W_u \times u \cdot W_v \times v, \tag{3}$$

where W_u and W_v are two matrices that weight the importance of the u components and the v components to p.

In [1], the authors drive that the compositional representation may be dominated by the semantic unit that has the deeper parsing tree as $\|u+v\| <= \|u\| + \|v\|$. The semantic unit that has a deeper parsing tree will have a higher value of the vector norm. However, this is not the case for the similarity between a compositional representation and an arbitrary word (w), see [1] equations 3.11 till 3.15.

2 Compositionality in Neural Network: An extension to the binary composition

Artificial neural networks (NN) are used to process a sequence of words to perform different NLP tasks. What we have discussed above is a binary function applied to a pair of constituent units. As we will see, the NN applies a binary compositional function in sequential order; $f_2(f_1(w1, w2), w3)$ to drive a compositional representation for a complex unit. Here, we will look over two types of NN; the recurrent NN (RNN) and the recursive-NN.

2.1 Recurrent NN

The RNN processes the words in steps starting with the first word w_1 to the last word w_n, assuming that the sequence has n words. At step t, the RNN generates a compositional representation h_t of the current sequence (from w_1 to w_t, $t < n$) by applying binary compositional functions f to the current word vector x_t and all previous words' composoitional representation h_{t-1} generated so far. Assuming forward direction, the binary compositional function will be:

$$h_t = tanh(W_h \times h_{t-1} \cdot W_x \times x_t) \tag{4}$$

2.1.1 LSTM

The LSTM is a gated variant of RNNs. Each gate apply a compositional functions as follow:

$$f^{(n)}(h_{t-1}, x_t) = W_h^{(n)} h_{t-1} + W_x^{(n)} x_t + b^{(n)} \tag{5}$$

As we can see, the functions used in recurrent RNNs; equations 4 and 5, are comparable with the additive model (equation 3).

2.2 Recursive NN

In recursive-NN, the words are not proceed in sequential order. Instead, they are processed according to their location in a tree data structure representing the complex linguistic unit, like a constituency parsing tree. The main idea here is that you encode constituent units according to the syntactic rules that govern their structure.

2.2.1 Recursive Matrix-Vector Model (MV-RNN)

In MV-RNN[3] and in addition to the word vector representation, each word is assigned a transformation matrix that describes how the word modifies the meaning of the other word it combines with. For example, if we have two words (a and b) that are child's of a linguistic unit p, the compositional functions are:

$$f_p = W_1 \begin{bmatrix} Ba \\ Ab \end{bmatrix} \tag{6a}$$

$$f_P = W_2 \begin{bmatrix} A \\ B \end{bmatrix}, \tag{6b}$$

where a and b denotes the word vector representation for words a and b, A and B denotes the matrix representation for words a and b, and f_p and f_P denotes the compositional vector and matrix representation for the constituent unit p.

2.2.2 Tree-LSTM

Tree-LSTM[4] is a recursive variant of LSTM that works over either dependency tree or constituency tree. In Tree-LSTM, the compositional functions are applied to the current word vector x_t, and all children's compositional representation generated so

far. There are two types of treeLSTM, Child-Sum, and N-ary. In the child-sum LSTM, the children vector representation is summed. Assume that a constituent unit p is a parent of k words, the compositional function is:

$$\tilde{h}_p = \sum_{l=1}^{k} h_l \tag{7a}$$

$$f^{(n)}(\tilde{h}_p, x_t) = W_h^{(n)} \tilde{h}_p + W_x^{(n)} x_t + b^{(n)} \tag{7b}$$

In the N-ary Tree-LSTM, each child's representation has his weight matrix. Thus, the number of children for each constituent is at most k.

$$f^{(n)}([h_p], x_t) = \sum_{l=1}^{k} W_l^{(n)} h_l + W_x^{(n)} x_t + b^{(n)} \tag{8}$$

Again, the functions used in recursive-NNs; equations 6, 7, and 8, are comparable with the additive model (equation 3).

3 Are deep networks compositional?

To answer this question, the author reviewed three experiments that investigated whether the NNs generalize based on shallow heuristics or capturing syntactic-based structure[2].

NOT A SHALLOW HEURISTIC: The authors of [12] investigated the RNN network used in [13] revealing a sub-network that is sensitive to syntactic structure.

PRODUCTIVE NOT COMPOSITIONAL: The RNNs show a remarkable performance with dataset random split experiments. However, they fail with controlled split experiments indicating that NNs fails to generalize based on syntactic structure[12].

ARCHITECTURE: An intersting finding is that CNN with attention provide better results than LSTM[12, 14].

ERRORS ARE NOT TRACTABLE: NNs failures to generalize are fuzzy and not systematic. This finding shows that the current understanding of NN generalization strategies is very limited. Interestingly, We know that latent tree learning models and recursive-NN drive tree structures that are different from what the linguist hypothesized[15, 16].

References

[1] Zhiyuan Liu, Yankai Lin, and Maosong Sun. "Compositional Semantics". In: *Representation Learning for Natural Language Processing*. Singapore: Springer Singapore, 2020, pp. 43–57. DOI: 10.1007/978-981-15-5573-2_3. URL: https://doi.org/10.1007/978-981-15-5573-2%5C_3.

[2] Marco Baroni. "Linguistic generalization and compositionality in modern artificial neural networks". In: *arXiv* (2019). DOI: 10.1098/rstb.2019.0307. eprint: 1904.00157.

[3] Richard Socher et al. "Semantic Compositionality through Recursive Matrix-Vector Spaces". In: *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. 2012, pp. 1201–1211.

[4] Kai Sheng Tai, Richard Socher, and Christopher D Manning. "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks". In: *arXiv* (2015). eprint: 1503.00075.

[5] Jonathan R. Brennan and Andrea E. Martin. "Phase synchronization varies systematically with linguistic structure composition". In: *Philosophical Transactions of the Royal Society B* 375.1791 (2020), p. 20190305. ISSN: 0962-8436. DOI: 10.1098/rstb.2019.0305.

[6] Liina Pylkkanen and Jonathan R Brennan. "Composition: The neurobiology of syntactic and semantic structure building". In: (2019).

[7] Francis Jeffry Pelletier. "The Principle of Semantic Compositionality". In: *Topoi* 13.1 (1994), pp. 11–24. ISSN: 0167-7411. DOI: 10.1007/bf00763644.

[8] Jeff Mitchell and Mirella Lapata. "Composition in distributional models of semantics". In: *Cognitive science* 34.8 (2010), pp. 1388–1429.

[9] Woodford A Beach and Samuel E Fox. *Papers from the 13th Regional Meeting Chicago Linguistic Society, April 14-16, 1977*. University of Chicago, 1977.

[10] Sebastian Pado and Mirella Lapata. "Dependency-based construction of semantic space models". In: *Computational Linguistics* 33.2 (2007), pp. 161–199.

[11] Jeff Mitchell and Mirella Lapata. "Vector-based models of semantic composition". In: *proceedings of ACL-08: HLT*. 2008, pp. 236–244.

[12] Tal Linzen and Brian Leonard. "Distinct patterns of syntactic agreement errors in recurrent networks and humans". In: *arXiv* (2018). eprint: 1807.06882.

[13] Ethan Wilcox et al. "What do RNN Language Models Learn about Filler–Gap Dependencies?" In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 211–221. DOI: 10.18653/v1/W18-5423. URL: https://www.aclweb.org/anthology/W18-5423.

[14] Roberto Dessì and Marco Baroni. "CNNs found to jump around more skillfully than RNNs: Compositional Generalization in Seq2seq Convolutional Networks". In: (2019), pp. 3919–3923. DOI: 10.18653/v1/p19-1381.

[15] Jonathan R Brennan and Andrea E Martin. "Phase synchronization varies systematically with linguistic structure composition". In: *Philosophical Transactions of the Royal Society B* 375.1791 (2020), p. 20190305.

[16] Adina Williams, Andrew Drozdov, and Samuel R Bowman. "Do latent tree learning models identify meaningful structure in sentences?" In: *arXiv* (2017). eprint: 1709.01121.