Learning Syntactic Properties with Deep Neural Networks

Deep Learning and the Nature of Linguistic Representation

Shalom Lappin

University of Gothenburg, Queen Mary University of London, and King's College London

WeSSLLI 2020, Brandeis University

July 14, 2020

Outline

Machine Learning of Syntactic Properties

- Both symbolic and statistical machine learning methods have been applied to syntactic learning tasks for many years, with varying levels of success.
- These include, *inter alia*, part of speech tagging, phrasal chunking, and sentential parsing.
- DL has made significant progress across these, and other syntactic applications.
- Identifying subject-verb agreement is a particularly interesting domain, because it involves long distance relations, and hierarchical structure.

Machine Learning of Syntactic Properties

- Both symbolic and statistical machine learning methods have been applied to syntactic learning tasks for many years, with varying levels of success.
- These include, *inter alia*, part of speech tagging, phrasal chunking, and sentential parsing.
- DL has made significant progress across these, and other syntactic applications.
- Identifying subject-verb agreement is a particularly interesting domain, because it involves long distance relations, and hierarchical structure.

Machine Learning of Syntactic Properties

- Both symbolic and statistical machine learning methods have been applied to syntactic learning tasks for many years, with varying levels of success.
- These include, *inter alia*, part of speech tagging, phrasal chunking, and sentential parsing.
- DL has made significant progress across these, and other syntactic applications.
- Identifying subject-verb agreement is a particularly interesting domain, because it involves long distance relations, and hierarchical structure.

Machine Learning of Syntactic Properties

- Both symbolic and statistical machine learning methods have been applied to syntactic learning tasks for many years, with varying levels of success.
- These include, *inter alia*, part of speech tagging, phrasal chunking, and sentential parsing.
- DL has made significant progress across these, and other syntactic applications.
- Identifying subject-verb agreement is a particularly interesting domain, because it involves long distance relations, and hierarchical structure.

Linzen et al.'s (2016) LSTM Agreement Model

- Linzen et al. (2016) train an LSTM on a subset of a Wikipedia corpus to predict the number of a verb.
- As they observe, the task increases in difficulty in relation to the length of the sequence of NPs with the wrong number feature that occur between a subject and the verb that it controls.
- They refer to such intervening NPs as *attractors*.

Linzen et al.'s (2016) LSTM Agreement Model

- Linzen et al. (2016) train an LSTM on a subset of a Wikipedia corpus to predict the number of a verb.
- As they observe, the task increases in difficulty in relation to the length of the sequence of NPs with the wrong number feature that occur between a subject and the verb that it controls.
- They refer to such intervening NPs as *attractors*.

Linzen et al.'s (2016) LSTM Agreement Model

- Linzen et al. (2016) train an LSTM on a subset of a Wikipedia corpus to predict the number of a verb.
- As they observe, the task increases in difficulty in relation to the length of the sequence of NPs with the wrong number feature that occur between a subject and the verb that it controls.
- They refer to such intervening NPs as *attractors*.

Agreement and Intervening Attractors

The subject-verb pairs are in italics, and the attractors are indicated in boldface.

1(a) *The students submit* a final project to complete the course.

(b) *The students* enrolled in **the program** *submit* a final project to complete the course.

(c) *The students* enrolled in **the program** in **the Department** *submit* a final project to complete the course.

(d) *The students* enrolled in **the program** in **the Department** where **my colleague** teaches *submit* a final project to complete the course.

Supervised Learning of Agreement

- Linzen et al. use a dependency parser to identify the controlling subject of each verb in their corpus of examples.
- This identification is necessary to compute the number of attractors, but it is not used in the training for the number prediction task.
- The number of the verb is morphologically manifest in the raw data.
- They train their LSTM on $\sim$121,500 examples (9% of the total corpus) by showing it the correct number feature of the verb.
- They test the LSTM's number predictions on $\sim$1.21 million examples (90% of their corpus).

Supervised Learning of Agreement

- Linzen et al. use a dependency parser to identify the controlling subject of each verb in their corpus of examples.
- This identification is necessary to compute the number of attractors, but it is not used in the training for the number prediction task.
- The number of the verb is morphologically manifest in the raw data.
- They train their LSTM on ~121,500 examples (9% of the total corpus) by showing it the correct number feature of the verb.
- They test the LSTM's number predictions on ~1.21 million examples (90% of their corpus).

Supervised Learning of Agreement

- Linzen et al. use a dependency parser to identify the controlling subject of each verb in their corpus of examples.
- This identification is necessary to compute the number of attractors, but it is not used in the training for the number prediction task.
- The number of the verb is morphologically manifest in the raw data.
- They train their LSTM on ∼121,500 examples (9% of the total corpus) by showing it the correct number feature of the verb.
- They test the LSTM's number predictions on ∼1.21 million examples (90% of their corpus).

Supervised Learning of Agreement

- Linzen et al. use a dependency parser to identify the controlling subject of each verb in their corpus of examples.
- This identification is necessary to compute the number of attractors, but it is not used in the training for the number prediction task.
- The number of the verb is morphologically manifest in the raw data.
- They train their LSTM on ∼121,500 examples (9% of the total corpus) by showing it the correct number feature of the verb.
- They test the LSTM's number predictions on ∼1.21 million examples (90% of their corpus).

Supervised Learning of Agreement

- Linzen et al. use a dependency parser to identify the controlling subject of each verb in their corpus of examples.
- This identification is necessary to compute the number of attractors, but it is not used in the training for the number prediction task.
- The number of the verb is morphologically manifest in the raw data.
- They train their LSTM on $\sim$121,500 examples (9% of the total corpus) by showing it the correct number feature of the verb.
- They test the LSTM's number predictions on $\sim$1.21 million examples (90% of their corpus).

Accuracy of The Model

- Linzen et al. encode input words as vectors in 50 dimensions, and their LSTM has 50 hidden units.
- They report that their system achieves 99% accuracy in the number prediction task for cases with 0 attractors between the subject and its verb.
- It declines to 83% accuracy for examples with 4 attractors.
- They do not report scores for examples with more than 4 attractors .

Accuracy of The Model

- Linzen et al. encode input words as vectors in 50 dimensions, and their LSTM has 50 hidden units.
- They report that their system achieves 99% accuracy in the number prediction task for cases with 0 attractors between the subject and its verb.
- It declines to 83% accuracy for examples with 4 attractors.
- They do not report scores for examples with more than 4 attractors .

Accuracy of The Model

- Linzen et al. encode input words as vectors in 50 dimensions, and their LSTM has 50 hidden units.
- They report that their system achieves 99% accuracy in the number prediction task for cases with 0 attractors between the subject and its verb.
- It declines to 83% accuracy for examples with 4 attractors.
- They do not report scores for examples with more than 4 attractors .

Accuracy of The Model

- Linzen et al. encode input words as vectors in 50 dimensions, and their LSTM has 50 hidden units.
- They report that their system achieves 99% accuracy in the number prediction task for cases with 0 attractors between the subject and its verb.
- It declines to 83% accuracy for examples with 4 attractors.
- They do not report scores for examples with more than 4 attractors .

Unsupervised Learning with a Language Model

- Linzen et al. also train a generative language model (predicting the next word) to predict the number of the verb in an unsupervised manner.
- In contrast to their supervised LSTM, their language model scores below chance in its predictions for 4 attractor cases.
- The much larger Google LM (Jozefowicz et al., 2016) does better, at a $\sim$45% error rate for 4 attractors, but it is still well below their supervised RNN.
- Linzen et al. conclude that a DNN can learn a considerable amount of syntactic structure, if it is properly supervised.

Unsupervised Learning with a Language Model

- Linzen et al. also train a generative language model (predicting the next word) to predict the number of the verb in an unsupervised manner.
- In contrast to their supervised LSTM, their language model scores below chance in its predictions for 4 attractor cases.
- The much larger Google LM (Jozefowicz et al., 2016) does better, at a ∼45% error rate for 4 attractors, but it is still well below their supervised RNN.
- Linzen et al. conclude that a DNN can learn a considerable amount of syntactic structure, if it is properly supervised.

Unsupervised Learning with a Language Model

- Linzen et al. also train a generative language model (predicting the next word) to predict the number of the verb in an unsupervised manner.
- In contrast to their supervised LSTM, their language model scores below chance in its predictions for 4 attractor cases.
- The much larger Google LM (Jozefowicz et al., 2016) does better, at a $\sim$45% error rate for 4 attractors, but it is still well below their supervised RNN.
- Linzen et al. conclude that a DNN can learn a considerable amount of syntactic structure, if it is properly supervised.

Unsupervised Learning with a Language Model

- Linzen et al. also train a generative language model (predicting the next word) to predict the number of the verb in an unsupervised manner.
- In contrast to their supervised LSTM, their language model scores below chance in its predictions for 4 attractor cases.
- The much larger Google LM (Jozefowicz et al., 2016) does better, at a $\sim$45% error rate for 4 attractors, but it is still well below their supervised RNN.
- Linzen et al. conclude that a DNN can learn a considerable amount of syntactic structure, if it is properly supervised.

Bernardy and Lappin's (2017) Agreement Experiments

- Bernardy and Lappin (2017) (BL) experiment with several DNN architectures, and alternative values for a variety of parameters, for the subject-verb agreement task.

- The architectures include an LSTM, a CNN, and a Gated Recurrent Unit (GRU, Cho et al., 2014).

- The parameters are
  - Ratio of training to testing as a partition of the corpus
  - Number of hidden units (memory size)
  - Vocabulary size
  - Number of layers
  - Dropout rate
  - Lexical embedding dimension size

Bernardy and Lappin's (2017) Agreement Experiments

- Bernardy and Lappin (2017) (BL) experiment with several DNN architectures, and alternative values for a variety of parameters, for the subject-verb agreement task.

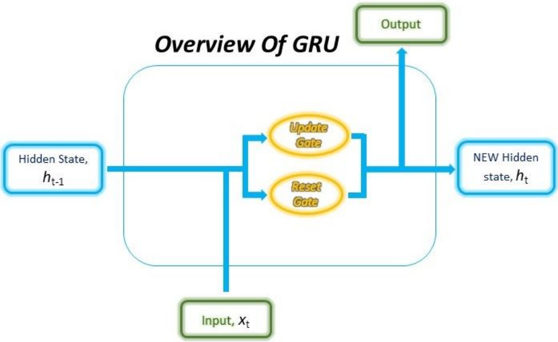
- The architectures include an LSTM, a CNN, and a Gated Recurrent Unit (GRU, Cho et al., 2014).

- The parameters are
  - Ratio of training to testing as a partition of the corpus
  - Number of hidden units (memory size)
  - Vocabulary size
  - Number of layers
  - Dropout rate
  - Lexical embedding dimension size

Bernardy and Lappin's (2017) Agreement Experiments

- Bernardy and Lappin (2017) (BL) experiment with several DNN architectures, and alternative values for a variety of parameters, for the subject-verb agreement task.
- The architectures include an LSTM, a CNN, and a Gated Recurrent Unit (GRU, Cho et al., 2014).
- The parameters are
  - Ratio of training to testing as a partition of the corpus
  - Number of hidden units (memory size)
  - Vocabulary size
  - Number of layers
  - Dropout rate
  - Lexical embedding dimension size

GRU Architecture



From Gabriel Loye, "Gated Recurrent Unit (GRU) With PyTorch", *FloydHub*, July 22, 2019

BL's CNN Design

- BL's CNN has 6 levels, with filtering successively compressing vector dimensions from 15 through 20, 15, 10 to 5.
- Convolution at these levels yields 7, 5, 5, and 3 features, respectively.
- Every convolution layer uses a ReLU activation function.
- The last layer is a dense layer with sigmoid activation.

BL's CNN Design

- BL's CNN has 6 levels, with filtering successively compressing vector dimensions from 15 through 20, 15, 10 to 5.
- Convolution at these levels yields 7, 5, 5, and 3 features, respectively.
- Every convolution layer uses a ReLU activation function.
- The last layer is a dense layer with sigmoid activation.

BL's CNN Design

- BL's CNN has 6 levels, with filtering successively compressing vector dimensions from 15 through 20, 15, 10 to 5.
- Convolution at these levels yields 7, 5, 5, and 3 features, respectively.
- Every convolution layer uses a ReLU activation function.
- The last layer is a dense layer with sigmoid activation.

BL's CNN Design

- BL's CNN has 6 levels, with filtering successively compressing vector dimensions from 15 through 20, 15, 10 to 5.
- Convolution at these levels yields 7, 5, 5, and 3 features, respectively.
- Every convolution layer uses a ReLU activation function.
- The last layer is a dense layer with sigmoid activation.

BL's CNN

$$
\begin{tikzcd}
\text{class} \\
\mathbb{R}^5 \arrow[u, "\text{dense}"'] \\
\mathbb{R}^{10} \arrow[u, "\text{6-dilated convolution, filter size = 3}"'] \\
\mathbb{R}^{15} \arrow[u, "\text{4-dilated convolution, filter size = 5}"'] \\
\mathbb{R}^{20} \arrow[u, "\text{2-dilated convolution, filter size = 5}"'] \\
\mathbb{R}^{50} \arrow[u, "\text{convolution, filter size = 7}"'] \\
\{0,1\}^{\text{voc. size}} \arrow[u, "\text{embedding}"']
\end{tikzcd}
$$

Training and Test Corpus

- BL use the WaCkypedia English corpus (Baroni et al., 2009), which contains ∼24 million example cases of present tense subject-verb agreement.
- The corpus is annotated with POS tags by TreeTagger (Schmid, 1995), and with dependency relations by the MaltParser (Nivre et al., 2007).
- Linzen et al. restrict training and testing to one agreement case per sentence in their corpus.
- BL use the full set of number agreement relations in the sentences of their corpus.

Training and Test Corpus

- BL use the WaCkypedia English corpus (Baroni et al., 2009), which contains ∼24 million example cases of present tense subject-verb agreement.
- The corpus is annotated with POS tags by TreeTagger (Schmid, 1995), and with dependency relations by the MaltParser (Nivre et al., 2007).
- Linzen et al. restrict training and testing to one agreement case per sentence in their corpus.
- BL use the full set of number agreement relations in the sentences of their corpus.

Training and Test Corpus

- BL use the WaCkypedia English corpus (Baroni et al., 2009), which contains ∼24 million example cases of present tense subject-verb agreement.
- The corpus is annotated with POS tags by TreeTagger (Schmid, 1995), and with dependency relations by the MaltParser (Nivre et al., 2007).
- Linzen et al. restrict training and testing to one agreement case per sentence in their corpus.
- BL use the full set of number agreement relations in the sentences of their corpus.

Training and Test Corpus

- BL use the WaCkypedia English corpus (Baroni et al., 2009), which contains $\sim$24 million example cases of present tense subject-verb agreement.
- The corpus is annotated with POS tags by TreeTagger (Schmid, 1995), and with dependency relations by the MaltParser (Nivre et al., 2007).
- Linzen et al. restrict training and testing to one agreement case per sentence in their corpus.
- BL use the full set of number agreement relations in the sentences of their corpus.

Training and Test Corpus

- Linzen et al. limit their test, but not their training examples to cases in which all NPs intervening between the subject and the verb that it controls are attractors.
- BL include the cases in which agreeing, as well as non-agreeing NPs intervene between the subject and its verb.
- Their motivation for departing from Linzen et al.'s experimental design is a concern to measure the accuracy with which a DNN predicts verb number in complex, and possibly confusing syntactic sequences.

Training and Test Corpus

- Linzen et al. limit their test, but not their training examples to cases in which all NPs intervening between the subject and the verb that it controls are attractors.
- BL include the cases in which agreeing, as well as non-agreeing NPs intervene between the subject and its verb.
- Their motivation for departing from Linzen et al.'s experimental design is a concern to measure the accuracy with which a DNN predicts verb number in complex, and possibly confusing syntactic sequences.

Training and Test Corpus

- Linzen et al. limit their test, but not their training examples to cases in which all NPs intervening between the subject and the verb that it controls are attractors.
- BL include the cases in which agreeing, as well as non-agreeing NPs intervene between the subject and its verb.
- Their motivation for departing from Linzen et al.'s experimental design is a concern to measure the accuracy with which a DNN predicts verb number in complex, and possibly confusing syntactic sequences.

BL's Supervised Learning Experiments

- BL first identify a benchmark of reasonable performance for the supervised DNN configuration and training.
- The benchmark is an LSTM with one layer of 150 units and no dropout, a data-set constructed with 10,000 words, lexical embeddings of dimension 50, and a training regimen of 90% of the corpus.
- They then ran experiments varying each of these parameters independently, holding the others constant.

BL's Supervised Learning Experiments

- BL first identify a benchmark of reasonable performance for the supervised DNN configuration and training.
- The benchmark is an LSTM with one layer of 150 units and no dropout, a data-set constructed with 10,000 words, lexical embeddings of dimension 50, and a training regimen of 90% of the corpus.
- They then ran experiments varying each of these parameters independently, holding the others constant.

BL's Supervised Learning Experiments

- BL first identify a benchmark of reasonable performance for the supervised DNN configuration and training.
- The benchmark is an LSTM with one layer of 150 units and no dropout, a data-set constructed with 10,000 words, lexical embeddings of dimension 50, and a training regimen of 90% of the corpus.
- They then ran experiments varying each of these parameters independently, holding the others constant.

Different Parameters Values

BL experiment with the following parameter values for their DNNs:

- Training on 10%, 50%, and 90% of the corpus, testing on the remainder for each split,
- 50, 150, 450 and 1350 units for the LSTM layers,
- Embedding vocabulary sizes of 100, 10k and 100k words, substituting corresponding POS tags for the rest,
- 1, 2 and 4 layers for the LSTM,
- Dropout rates of 0, 0.1, 0.2 and 0.5, applied to the weights within the LSTM layers, but not at the final dense layer, and
- Lexical embedding dimension sizes of 17, 50, 150, and 450.

Different Parameters Values

BL experiment with the following parameter values for their DNNs:

- Training on 10%, 50%, and 90% of the corpus, testing on the remainder for each split,
- 50, 150, 450 and 1350 units for the LSTM layers,
- Embedding vocabulary sizes of 100, 10k and 100k words, substituting corresponding POS tags for the rest,
- 1, 2 and 4 layers for the LSTM,
- Dropout rates of 0, 0.1, 0.2 and 0.5, applied to the weights within the LSTM layers, but not at the final dense layer, and
- Lexical embedding dimension sizes of 17, 50, 150, and 450.

Different Parameters Values

BL experiment with the following parameter values for their DNNs:

- Training on 10%, 50%, and 90% of the corpus, testing on the remainder for each split,
- 50, 150, 450 and 1350 units for the LSTM layers,
- Embedding vocabulary sizes of 100, 10k and 100k words, substituting corresponding POS tags for the rest,
- 1, 2 and 4 layers for the LSTM,
- Dropout rates of 0, 0.1, 0.2 and 0.5, applied to the weights within the LSTM layers, but not at the final dense layer, and
- Lexical embedding dimension sizes of 17, 50, 150, and 450.

Different Parameters Values

BL experiment with the following parameter values for their DNNs:

- Training on 10%, 50%, and 90% of the corpus, testing on the remainder for each split,
- 50, 150, 450 and 1350 units for the LSTM layers,
- Embedding vocabulary sizes of 100, 10k and 100k words, substituting corresponding POS tags for the rest,
- 1, 2 and 4 layers for the LSTM,
- Dropout rates of 0, 0.1, 0.2 and 0.5, applied to the weights within the LSTM layers, but not at the final dense layer, and
- Lexical embedding dimension sizes of 17, 50, 150, and 450.

Different Parameters Values

BL experiment with the following parameter values for their DNNs:

- Training on 10%, 50%, and 90% of the corpus, testing on the remainder for each split,
- 50, 150, 450 and 1350 units for the LSTM layers,
- Embedding vocabulary sizes of 100, 10k and 100k words, substituting corresponding POS tags for the rest,
- 1, 2 and 4 layers for the LSTM,
- Dropout rates of 0, 0.1, 0.2 and 0.5, applied to the weights within the LSTM layers, but not at the final dense layer, and
- Lexical embedding dimension sizes of 17, 50, 150, and 450.

Different Parameters Values

BL experiment with the following parameter values for their DNNs:

- Training on 10%, 50%, and 90% of the corpus, testing on the remainder for each split,
- 50, 150, 450 and 1350 units for the LSTM layers,
- Embedding vocabulary sizes of 100, 10k and 100k words, substituting corresponding POS tags for the rest,
- 1, 2 and 4 layers for the LSTM,
- Dropout rates of 0, 0.1, 0.2 and 0.5, applied to the weights within the LSTM layers, but not at the final dense layer, and
- Lexical embedding dimension sizes of 17, 50, 150, and 450.

Hypothesis Concerning Reduced Vocabulary

- BL hypothesise that a DNN will learn the target syntactic dependency pattern more efficiently if it is exposed to input consisting largely of POS sequences in which number features are marked on noun and verb tags.
- They conjecture that such input would highlight the dependency relations more clearly by abstracting away from possibly confounding distributional lexical information contained in richer embeddings.
- On this view impoverished lexical sequences would facilitate DNN learning of agreement patterns through highlighting the relevant number feature.

Hypothesis Concerning Reduced Vocabulary

- BL hypothesise that a DNN will learn the target syntactic dependency pattern more efficiently if it is exposed to input consisting largely of POS sequences in which number features are marked on noun and verb tags.
- They conjecture that such input would highlight the dependency relations more clearly by abstracting away from possibly confounding distributional lexical information contained in richer embeddings.
- On this view impoverished lexical sequences would facilitate DNN learning of agreement patterns through highlighting the relevant number feature.

Hypothesis Concerning Reduced Vocabulary

- BL hypothesise that a DNN will learn the target syntactic dependency pattern more efficiently if it is exposed to input consisting largely of POS sequences in which number features are marked on noun and verb tags.
- They conjecture that such input would highlight the dependency relations more clearly by abstracting away from possibly confounding distributional lexical information contained in richer embeddings.
- On this view impoverished lexical sequences would facilitate DNN learning of agreement patterns through highlighting the relevant number feature.

An LSTM Language Model

- BL also train an LSTM as a generative language model.
- The LSTM has two layers of 1200 units per layer, and a dropout rate of 0.5.
- It is trained on the WaCky corpus of sentences, with the 100 most common words, and corresponding POS tags substituted for the others in the sentences.
- This design is intended to test the reduced vocabulary hypothesis.

An LSTM Language Model

- BL also train an LSTM as a generative language model.
- The LSTM has two layers of 1200 units per layer, and a dropout rate of 0.5.
- It is trained on the WaCky corpus of sentences, with the 100 most common words, and corresponding POS tags substituted for the others in the sentences.
- This design is intended to test the reduced vocabulary hypothesis.

An LSTM Language Model

- BL also train an LSTM as a generative language model.
- The LSTM has two layers of 1200 units per layer, and a dropout rate of 0.5.
- It is trained on the WaCky corpus of sentences, with the 100 most common words, and corresponding POS tags substituted for the others in the sentences.
- This design is intended to test the reduced vocabulary hypothesis.

An LSTM Language Model

- BL also train an LSTM as a generative language model.
- The LSTM has two layers of 1200 units per layer, and a dropout rate of 0.5.
- It is trained on the WaCky corpus of sentences, with the 100 most common words, and corresponding POS tags substituted for the others in the sentences.
- This design is intended to test the reduced vocabulary hypothesis.

Unsupervised Prediction of Agreement

- BL use their neural LM for unsupervised prediction of agreement in two ways.
- Let $p(w_i|w_{i-1},...,w_{i-k})$ be the predicted probability of a word w in a string, given the prefix of $w_{i-1},...,w_{i-k}$ of preceding words in that string.
- On the first approach, they determine, for each sentence in the test set, whether the following condition holds: $\sum_{V^n} p(V_i^n|w_{i-1},...,w_{i-k}) > \sum_{V^{\neg n}} p(V_i^{\neg n}|w_{i-1},...,w_{i-k})$, where V^n ranges over verbs with the correct number feature (n) in a particular string, and $V^{\neg n}$ ranges over verbs with the wrong number feature in that string.
- On the second method BL test to see if the following condition holds: $p(Verb_i^n|w_{i-1},...,w_{i-k}) > p(Verb_i^{\neg n}|w_{i-1},...,w_{i-k})$.

Unsupervised Prediction of Agreement

- BL use their neural LM for unsupervised prediction of agreement in two ways.
- Let $p(w_i|w_{i-1}, ..., w_{i-k})$ be the predicted probability of a word w in a string, given the prefix of $w_{i-1}, ..., w_{i-k}$ of preceding words in that string.
- On the first approach, they determine, for each sentence in the test set, whether the following condition holds: $\sum_{V^n} p(V_i^n|w_{i-1}, ..., w_{i-k}) > \sum_{V^{\neg n}} p(V_i^{\neg n}|w_{i-1}, ..., w_{i-k})$, where V^n ranges over verbs with the correct number feature (n) in a particular string, and $V^{\neg n}$ ranges over verbs with the wrong number feature in that string.
- On the second method BL test to see if the following condition holds: $p(Verb_i^n|w_{i-1}, ..., w_{i-k}) > p(Verb_i^{\neg n}|w_{i-1}, ..., w_{i-k})$.

Unsupervised Prediction of Agreement

- BL use their neural LM for unsupervised prediction of agreement in two ways.
- Let $p(w_i|w_{i-1},...,w_{i-k})$ be the predicted probability of a word w in a string, given the prefix of $w_{i-1},...,w_{i-k}$ of preceding words in that string.
- On the first approach, they determine, for each sentence in the test set, whether the following condition holds: $\sum_{V^n} p(V_i^n|w_{i-1},...,w_{i-k}) > \sum_{V^{\neg n}} p(V_i^{\neg n}|w_{i-1},...,w_{i-k})$, where V^n ranges over verbs with the correct number feature (n) in a particular string, and $V^{\neg n}$ ranges over verbs with the wrong number feature in that string.
- On the second method BL test to see if the following condition holds: $p(Verb_i^n|w_{i-1},...,w_{i-k}) > p(Verb_i^{\neg n}|w_{i-1},...,w_{i-k})$.

Unsupervised Prediction of Agreement

- BL use their neural LM for unsupervised prediction of agreement in two ways.
- Let $p(w_i|w_{i-1},...,w_{i-k})$ be the predicted probability of a word w in a string, given the prefix of $w_{i-1},...,w_{i-k}$ of preceding words in that string.
- On the first approach, they determine, for each sentence in the test set, whether the following condition holds:
  $\sum_{V^n} p(V_i^n|w_{i-1},...,w_{i-k}) > \sum_{V^{\neg n}} p(V_i^{\neg n}|w_{i-1},...,w_{i-k})$,
  where V^n ranges over verbs with the correct number feature (n) in a particular string, and $V^{\neg n}$ ranges over verbs with the wrong number feature in that string.
- On the second method BL test to see if the following condition holds:
  $p(Verb_i^n|w_{i-1},...,w_{i-k}) > p(Verb_i^{\neg n}|w_{i-1},...,w_{i-k})$.

Unsupervised Prediction of Agreement

- The first method measures the summed conditional probabilities of all correctly number inflected verbs after the prefix in a string against the summed predicted probabilities of all incorrectly number inflected verbs appearing in this position (*the summing method*).
- The second procedure compares the predicted probability of the correctly number-marked form of the actual verb in this position with that of its incorrectly marked form (*the verb targeted method*).
- While in the summing method the LM is not given any semantic cue, in the verb targeted method it is priimed to expect a specific verb.
- This bias contributes minimal semantic information, given that the BL LM's 100-word vocabulary contains only the inflectional verb forms "to be", "to have", and "to state", with other verbs represented by the "VV" part of speech code.

Unsupervised Prediction of Agreement

- The first method measures the summed conditional probabilities of all correctly number inflected verbs after the prefix in a string against the summed predicted probabilities of all incorrectly number inflected verbs appearing in this position (*the summing method*).
- The second procedure compares the predicted probability of the correctly number-marked form of the actual verb in this position with that of its incorrectly marked form (*the verb targeted method*).
- While in the summing method the LM is not given any semantic cue, in the verb targeted method it is priimed to expect a specific verb.
- This bias contributes minimal semantic information, given that the BL LM's 100-word vocabulary contains only the inflectional verb forms "to be", "to have", and "to state", with other verbs represented by the "VV" part of speech code.

Unsupervised Prediction of Agreement

- The first method measures the summed conditional probabilities of all correctly number inflected verbs after the prefix in a string against the summed predicted probabilities of all incorrectly number inflected verbs appearing in this position (*the summing method*).
- The second procedure compares the predicted probability of the correctly number-marked form of the actual verb in this position with that of its incorrectly marked form (*the verb targeted method*).
- While in the summing method the LM is not given any semantic cue, in the verb targeted method it is priimed to expect a specific verb.
- This bias contributes minimal semantic information, given that the BL LM's 100-word vocabulary contains only the inflectional verb forms "to be", "to have", and "to state", with other verbs represented by the "VV" part of speech code.

Unsupervised Prediction of Agreement

- The first method measures the summed conditional probabilities of all correctly number inflected verbs after the prefix in a string against the summed predicted probabilities of all incorrectly number inflected verbs appearing in this position (*the summing method*).
- The second procedure compares the predicted probability of the correctly number-marked form of the actual verb in this position with that of its incorrectly marked form (*the verb targeted method*).
- While in the summing method the LM is not given any semantic cue, in the verb targeted method it is priimed to expect a specific verb.
- This bias contributes minimal semantic information, given that the BL LM's 100-word vocabulary contains only the inflectional verb forms "to be", "to have", and "to state", with other verbs represented by the "VV" part of speech code.

Results of BL Experiments

- In all the following graphs, except for the last one (number of test examples per number of attractors) the *y*-axis gives the accuracy rate, and the *x*-axis the number of NP attractors.
- 73% of the verbs in the test sets are singular.
- This provides a majority choice baseline, which is indicated by a straight horizontal line in the graphs.

Results of BL Experiments

- In all the following graphs, except for the last one (number of test examples per number of attractors) the *y*-axis gives the accuracy rate, and the *x*-axis the number of NP attractors.
- 73% of the verbs in the test sets are singular.
- This provides a majority choice baseline, which is indicated by a straight horizontal line in the graphs.

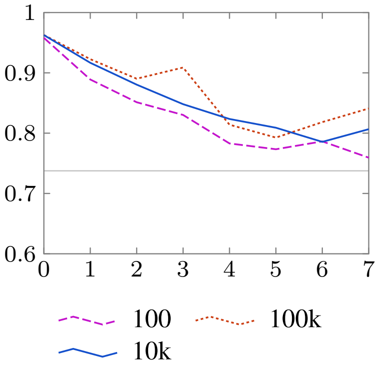
Results of BL Experiments

- In all the following graphs, except for the last one (number of test examples per number of attractors) the y-axis gives the accuracy rate, and the x-axis the number of NP attractors.
- 73% of the verbs in the test sets are singular.
- This provides a majority choice baseline, which is indicated by a straight horizontal line in the graphs.

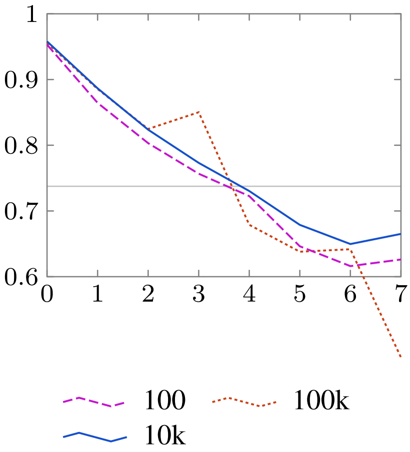
LSTM Vocabulary Size

A reduced vocabulary of the 100 most common words, with POS tags for the remainder, reduces accuracy across DNN architectures, for supervised learning.

Increasing the vocabulary to 100k words yields a significant improvement for the LSTM, but gives mixed results for the CNN.

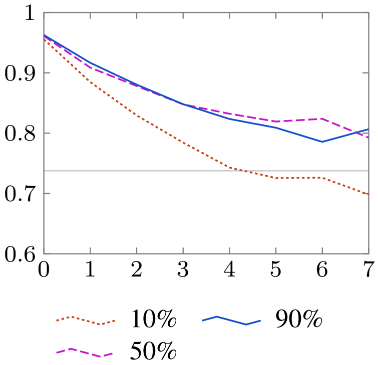


CNN Vocabulary Size



Training Size

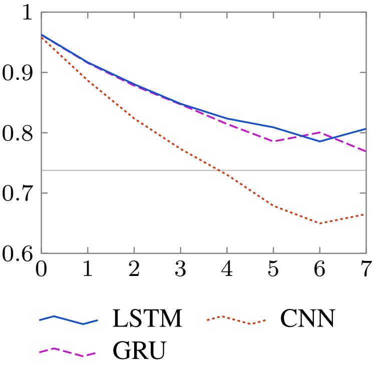
Increasing the ratio of training to testing examples from 10% to 50% significantly improves the performance of the LSTM (with 150 units and a vocabulary of 10,000 word embeddings).
Further increasing it to 90% does not make much of a difference, even degrading accuracy slightly at 6 attractors.



Architecture

The LSTM and GRU perform at a comparable level.

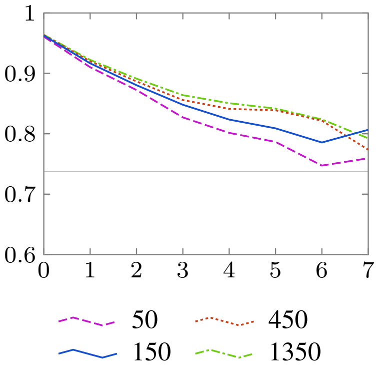
Both achieve significantly better accuracy than the CNN.



Memory Size

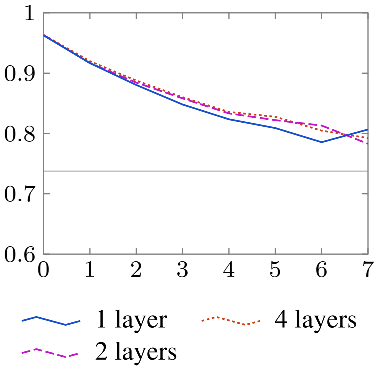
Increasing the number of units in an LSTM improves accuracy relative to the number of attractors.

Each three-fold increase in units achieves a similar improvement in percentage points for a higher number of attractors, up to 450 units.



Number of Layers

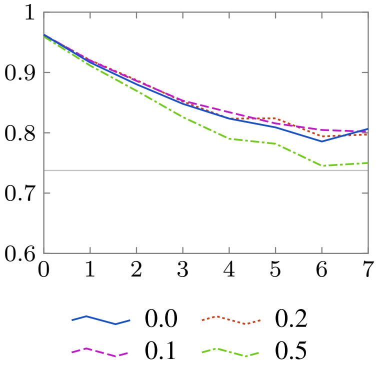
Increasing the number of layers for an LSTM from 1 150-unit layer to 2 such layers marginally improves its performance.
A further increase to 4 150-unit layers makes no clear difference.



Dropout Rates

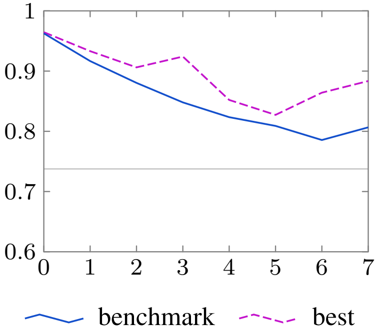
A dropout rate of 0.1 for the LSTM RNN (1 layer with 150 units) improves LSTM performance slightly.

An increase to 0.2 provides no clear benefit, while increasing the rate to 0.5 degrades performance.



Model with the Best Parameters

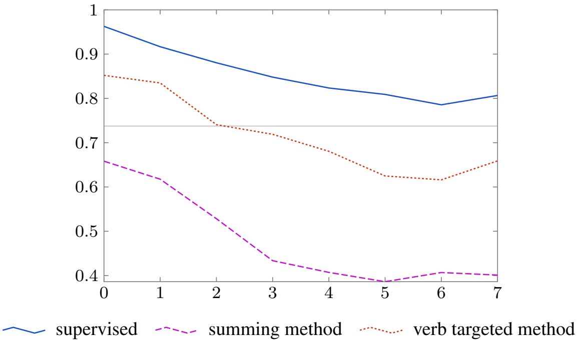
The DNN configured with the best observed parameter values is an LSTM with 2 layers, 1350 units, a dropout rate of 0.1, a vocabulary size of 100k, trained on 90% of the corpus, and lexical embedding size of 150 dimensions.



Unsupervised Language Model

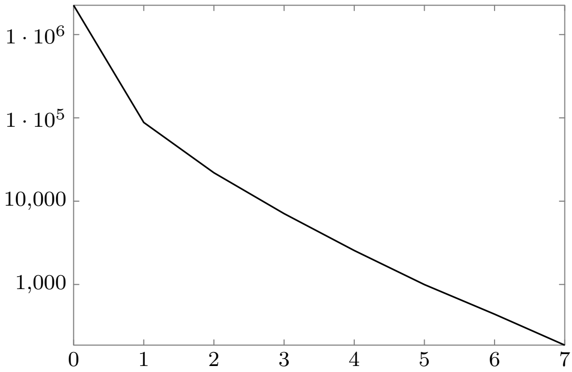
The BL LM prediction of verb number with the summing method is comparable to Linzen et al.'s LM.

The verb targeted method achieves a far higher level of accuracy than their model, and than the much larger Google LM, but it is still below BL's best supervised results.



Training Examples and Attractors

There is an inverse relation between the number of examples in the corpus and the number of attractor NPs in these sentences.



Significance of BL's Experiments

- BL's results support Linzen et al.'s finding that RNNs (both LSTMs and GRUs) learn long distance syntactic dependencies within extended, complex sequences.
- Their success in learning subject-verb agreement scales with the size of the data set on which they train.
- Training DNNs on data that is lexically impoverished, but highlights the syntactic elements of a relation, does not (for this task) facilitate learning, but degrades it, contrary to their initial hypothesis.
- This suggests that DNNs extract syntactic patterns incrementally from lexical embeddings, through recognition of their distributional regularities.
- An unsupervised language model can achieve reasonable results on the agreement prediction task.

Significance of BL's Experiments

- BL's results support Linzen et al.'s finding that RNNs (both LSTMs and GRUs) learn long distance syntactic dependencies within extended, complex sequences.
- Their success in learning subject-verb agreement scales with the size of the data set on which they train.
- Training DNNs on data that is lexically impoverished, but highlights the syntactic elements of a relation, does not (for this task) facilitate learning, but degrades it, contrary to their initial hypothesis.
- This suggests that DNNs extract syntactic patterns incrementally from lexical embeddings, through recognition of their distributional regularities.
- An unsupervised language model can achieve reasonable results on the agreement prediction task.

Significance of BL's Experiments

- BL's results support Linzen et al.'s finding that RNNs (both LSTMs and GRUs) learn long distance syntactic dependencies within extended, complex sequences.
- Their success in learning subject-verb agreement scales with the size of the data set on which they train.
- Training DNNs on data that is lexically impoverished, but highlights the syntactic elements of a relation, does not (for this task) facilitate learning, but degrades it, contrary to their initial hypothesis.
- This suggests that DNNs extract syntactic patterns incrementally from lexical embeddings, through recognition of their distributional regularities.
- An unsupervised language model can achieve reasonable results on the agreement prediction task.

Significance of BL's Experiments

- BL's results support Linzen et al.'s finding that RNNs (both LSTMs and GRUs) learn long distance syntactic dependencies within extended, complex sequences.
- Their success in learning subject-verb agreement scales with the size of the data set on which they train.
- Training DNNs on data that is lexically impoverished, but highlights the syntactic elements of a relation, does not (for this task) facilitate learning, but degrades it, contrary to their initial hypothesis.
- This suggests that DNNs extract syntactic patterns incrementally from lexical embeddings, through recognition of their distributional regularities.
- An unsupervised language model can achieve reasonable results on the agreement prediction task.

Significance of BL's Experiments

- BL's results support Linzen et al.'s finding that RNNs (both LSTMs and GRUs) learn long distance syntactic dependencies within extended, complex sequences.
- Their success in learning subject-verb agreement scales with the size of the data set on which they train.
- Training DNNs on data that is lexically impoverished, but highlights the syntactic elements of a relation, does not (for this task) facilitate learning, but degrades it, contrary to their initial hypothesis.
- This suggests that DNNs extract syntactic patterns incrementally from lexical embeddings, through recognition of their distributional regularities.
- An unsupervised language model can achieve reasonable results on the agreement prediction task.

The Nature of Syntactic Representation

- It is an open question as to how DNN learning resembles and diverges from human learning.
- BL make no cognitive claims concerning the relevance of their experiments to human syntactic representation.
- It is interesting to note that some recent work in neurolinguistics indicates that syntactic knowledge is distributed through different language centres in the brain, and closely integrated with lexical-semantic representations (Blank et al., 2016).
- This lexically encoded and distributed way of representing syntactic information is consistent with the role of rich lexical embeddings in DNN syntactic learning.

The Nature of Syntactic Representation

- It is an open question as to how DNN learning resembles and diverges from human learning.
- BL make no cognitive claims concerning the relevance of their experiments to human syntactic representation.
- It is interesting to note that some recent work in neurolinguistics indicates that syntactic knowledge is distributed through different language centres in the brain, and closely integrated with lexical-semantic representations (Blank et al., 2016).
- This lexically encoded and distributed way of representing syntactic information is consistent with the role of rich lexical embeddings in DNN syntactic learning.

The Nature of Syntactic Representation

- It is an open question as to how DNN learning resembles and diverges from human learning.
- BL make no cognitive claims concerning the relevance of their experiments to human syntactic representation.
- It is interesting to note that some recent work in neurolinguistics indicates that syntactic knowledge is distributed through different language centres in the brain, and closely integrated with lexical-semantic representations (Blank et al., 2016).
- This lexically encoded and distributed way of representing syntactic information is consistent with the role of rich lexical embeddings in DNN syntactic learning.

The Nature of Syntactic Representation

- It is an open question as to how DNN learning resembles and diverges from human learning.
- BL make no cognitive claims concerning the relevance of their experiments to human syntactic representation.
- It is interesting to note that some recent work in neurolinguistics indicates that syntactic knowledge is distributed through different language centres in the brain, and closely integrated with lexical-semantic representations (Blank et al., 2016).
- This lexically encoded and distributed way of representing syntactic information is consistent with the role of rich lexical embeddings in DNN syntactic learning.

Neural LMs and Hierarchical Syntactic Structure

- Gulordava et al. (2018) use an LSTM LM for unsupervised learning of subject-verb agreement, with tests sets in Italian, English, Hebrew, and Russian.
- They extract test sentences from a dependency tree bank, and convert them into nonsense (nonce) sentences by substituting arbitrary lexical items with corresponding POS, from the LM's vocabulary, for the the nouns and verbs in the originals.
- The test set is comprised of both the original and the nonsense sentences.
- They train the LSTM LM on Wikipedia text from the four languages, with 50m tokens in each training corpora.

Neural LMs and Hierarchical Syntactic Structure

- Gulordava et al. (2018) use an LSTM LM for unsupervised learning of subject-verb agreement, with tests sets in Italian, English, Hebrew, and Russian.
- They extract test sentences from a dependency tree bank, and convert them into nonsense (nonce) sentences by substituting arbitrary lexical items with corresponding POS, from the LM's vocabulary, for the the nouns and verbs in the originals.
- The test set is comprised of both the original and the nonsense sentences.
- They train the LSTM LM on Wikipedia text from the four languages, with 50m tokens in each training corpora.

Neural LMs and Hierarchical Syntactic Structure

- Gulordava et al. (2018) use an LSTM LM for unsupervised learning of subject-verb agreement, with tests sets in Italian, English, Hebrew, and Russian.
- They extract test sentences from a dependency tree bank, and convert them into nonsense (nonce) sentences by substituting arbitrary lexical items with corresponding POS, from the LM's vocabulary, for the the nouns and verbs in the originals.
- The test set is comprised of both the original and the nonsense sentences.
- They train the LSTM LM on Wikipedia text from the four languages, with 50m tokens in each training corpora.

Neural LMs and Hierarchical Syntactic Structure

- Gulordava et al. (2018) use an LSTM LM for unsupervised learning of subject-verb agreement, with tests sets in Italian, English, Hebrew, and Russian.
- They extract test sentences from a dependency tree bank, and convert them into nonsense (nonce) sentences by substituting arbitrary lexical items with corresponding POS, from the LM's vocabulary, for the the nouns and verbs in the originals.
- The test set is comprised of both the original and the nonsense sentences.
- They train the LSTM LM on Wikipedia text from the four languages, with 50m tokens in each training corpora.

Gulordava et al.'s Experimental Results

	Italian	English	Hebrew	Russian
Unigram Majority Baseline				
Original	54.6	65.9	67.8	60.2
Nonce	54.1	42.5	63.1	54.0
5-ngram Kneser-Ney smoothing				
Original	63.9	63.4	72.1	73.5
Nounce	52.8	43.4	61.7	56.8
Perplexity	147.8	168.9	122.0	166.6
5-gram LSTM				
Original	81.8	70.2	90.9	91.5
Nounce	78.0	58.2	77.5	85.7
Perplexity	62.6	71.6	59.9	61.1
LSTM				
Original	92.1	81.0	94.7	96.1
Nounce	85.5	74.1	80.8	88.8
Perplexity	45.2	52.1	42.5	48.9

Gulordava et al.'s Results

- Gulordava et al.'s LM significantly outperforms the three baseline systems.
- The fact that it yields reasonable accuracy for nonce sentences indicates that it learns hierarchical syntactic structure independently of semantic cues.
- But its predictive accuracy is still significantly higher for the original sentences, suggesting that these cues facilitate syntactic learning, as BL's work suggests.
- The model performs better with morphologically richer languages, like Hebrew, and Russian, than with English.
- There is a correlation between the quality of the LM (perplexity value) and its accuracy on the agreement task.

Gulordava et al.'s Results

- Gulordava et al.'s LM significantly outperforms the three baseline systems.
- The fact that it yields reasonable accuracy for nonce sentences indicates that it learns hierarchical syntactic structure independently of semantic cues.
- But its predictive accuracy is still significantly higher for the original sentences, suggesting that these cues facilitate syntactic learning, as BL's work suggests.
- The model performs better with morphologically richer languages, like Hebrew, and Russian, than with English.
- There is a correlation between the quality of the LM (perplexity value) and its accuracy on the agreement task.

Gulordava et al.'s Results

- Gulordava et al.'s LM significantly outperforms the three baseline systems.
- The fact that it yields reasonable accuracy for nonce sentences indicates that it learns hierarchical syntactic structure independently of semantic cues.
- But its predictive accuracy is still significantly higher for the original sentences, suggesting that these cues facilitate syntactic learning, as BL's work suggests.
- The model performs better with morphologically richer languages, like Hebrew, and Russian, than with English.
- There is a correlation between the quality of the LM (perplexity value) and its accuracy on the agreement task.

Gulordava et al.'s Results

- Gulordava et al.'s LM significantly outperforms the three baseline systems.
- The fact that it yields reasonable accuracy for nonce sentences indicates that it learns hierarchical syntactic structure independently of semantic cues.
- But its predictive accuracy is still significantly higher for the original sentences, suggesting that these cues facilitate syntactic learning, as BL's work suggests.
- The model performs better with morphologically richer languages, like Hebrew, and Russian, than with English.
- There is a correlation between the quality of the LM (perplexity value) and its accuracy on the agreement task.

Gulordava et al.'s Results

- Gulordava et al.'s LM significantly outperforms the three baseline systems.
- The fact that it yields reasonable accuracy for nonce sentences indicates that it learns hierarchical syntactic structure independently of semantic cues.
- But its predictive accuracy is still significantly higher for the original sentences, suggesting that these cues facilitate syntactic learning, as BL's work suggests.
- The model performs better with morphologically richer languages, like Hebrew, and Russian, than with English.
- There is a correlation between the quality of the LM (perplexity value) and its accuracy on the agreement task.

The LM Compared to Human Predictions

- Gulordava et al. compare the performance of their LSTM LM to AMT human predictions for their Italian data set.
- They report that the LSTM model approaches the human level of performance.
- For the original sentences, average human accuracy is 94.5, and the LSTM LM is 92.1.
- For the nonce sentences, average human accuracy is 88.4, and the LSTM LM is 85.5.
- These results support the conjecture that there are parallels in the ways that humans and DNNs learn to perform certain linguistic tasks.

The LM Compared to Human Predictions

- Gulordava et al. compare the performance of their LSTM LM to AMT human predictions for their Italian data set.
- They report that the LSTM model approaches the human level of performance.
- For the original sentences, average human accuracy is 94.5, and the LSTM LM is 92.1.
- For the nonce sentences, average human accuracy is 88.4, and the LSTM LM is 85.5.
- These results support the conjecture that there are parallels in the ways that humans and DNNs learn to perform certain linguistic tasks.

The LM Compared to Human Predictions

- Gulordava et al. compare the performance of their LSTM LM to AMT human predictions for their Italian data set.
- They report that the LSTM model approaches the human level of performance.
- For the original sentences, average human accuracy is 94.5, and the LSTM LM is 92.1.
- For the nonce sentences, average human accuracy is 88.4, and the LSTM LM is 85.5.
- These results support the conjecture that there are parallels in the ways that humans and DNNs learn to perform certain linguistic tasks.

The LM Compared to Human Predictions

- Gulordava et al. compare the performance of their LSTM LM to AMT human predictions for their Italian data set.
- They report that the LSTM model approaches the human level of performance.
- For the original sentences, average human accuracy is 94.5, and the LSTM LM is 92.1.
- For the nonce sentences, average human accuracy is 88.4, and the LSTM LM is 85.5.
- These results support the conjecture that there are parallels in the ways that humans and DNNs learn to perform certain linguistic tasks.

The LM Compared to Human Predictions

- Gulordava et al. compare the performance of their LSTM LM to AMT human predictions for their Italian data set.
- They report that the LSTM model approaches the human level of performance.
- For the original sentences, average human accuracy is 94.5, and the LSTM LM is 92.1.
- For the nonce sentences, average human accuracy is 88.4, and the LSTM LM is 85.5.
- These results support the conjecture that there are parallels in the ways that humans and DNNs learn to perform certain linguistic tasks.

Agreement Across Different Structures

- Marvin and Linzen (2018) test an LSTM LM on subject-verb agreement in ten different syntactic structures.
- They present the model with minimal pairs of correct and ill-formed instances of each structure, rating the LM's prediction as accurate if it assigns higher probability to the well-formed sentence.
- They also use AMT crowdsourcing to obtain binary human judgments on these pairs.
- In 5 of the 10 constructions the LSTM LM approaches or surpasses human performance, but in the other 5 it scores 21-35 points below the human judgment standard.

Agreement Across Different Structures

- Marvin and Linzen (2018) test an LSTM LM on subject-verb agreement in ten different syntactic structures.
- They present the model with minimal pairs of correct and ill-formed instances of each structure, rating the LM's prediction as accurate if it assigns higher probability to the well-formed sentence.
- They also use AMT crowdsourcing to obtain binary human judgments on these pairs.
- In 5 of the 10 constructions the LSTM LM approaches or surpasses human performance, but in the other 5 it scores 21-35 points below the human judgment standard.

Agreement Across Different Structures

- Marvin and Linzen (2018) test an LSTM LM on subject-verb agreement in ten different syntactic structures.
- They present the model with minimal pairs of correct and ill-formed instances of each structure, rating the LM's prediction as accurate if it assigns higher probability to the well-formed sentence.
- They also use AMT crowdsourcing to obtain binary human judgments on these pairs.
- In 5 of the 10 constructions the LSTM LM approaches or surpasses human performance, but in the other 5 it scores 21-35 points below the human judgment standard.

Agreement Across Different Structures

- Marvin and Linzen (2018) test an LSTM LM on subject-verb agreement in ten different syntactic structures.
- They present the model with minimal pairs of correct and ill-formed instances of each structure, rating the LM's prediction as accurate if it assigns higher probability to the well-formed sentence.
- They also use AMT crowdsourcing to obtain binary human judgments on these pairs.
- In 5 of the 10 constructions the LSTM LM approaches or surpasses human performance, but in the other 5 it scores 21-35 points below the human judgment standard.

Applying BERT to the Marvin-Linzen Test Set

- Goldberg (2019) uses BERT (Base and Large, both untuned) as a LM on Marvin and Linzen's test set.
- He masks the contrasting verbs in each minimal pair, and uses BERT's pre-softmax logit scores as quasi-probability values to test its prediction for each of these verbs.
- By virtue of its bidirectional processing of input, BERT conditions its predicted scores on both the left and right contexts of the masked verb.
- In 9 of the 10 syntactic constructions BERT approaches or surpasses Marvin and Linzen's reported human judgment accuracy, in some cases by a significant margin.
- In the one exception, sentential complements, BERT Base scores 0.83, and BERT Large 0.86, while human accuracy is 0.93.

Applying BERT to the Marvin-Linzen Test Set

- Goldberg (2019) uses BERT (Base and Large, both untuned) as a LM on Marvin and Linzen's test set.
- He masks the contrasting verbs in each minimal pair, and uses BERT's pre-softmax logit scores as quasi-probability values to test its prediction for each of these verbs.
- By virtue of its bidirectional processing of input, BERT conditions its predicted scores on both the left and right contexts of the masked verb.
- In 9 of the 10 syntactic constructions BERT approaches or surpasses Marvin and Linzen's reported human judgment accuracy, in some cases by a significant margin.
- In the one exception, sentential complements, BERT Base scores 0.83, and BERT Large 0.86, while human accuracy is 0.93.

Applying BERT to the Marvin-Linzen Test Set

- Goldberg (2019) uses BERT (Base and Large, both untuned) as a LM on Marvin and Linzen's test set.
- He masks the contrasting verbs in each minimal pair, and uses BERT's pre-softmax logit scores as quasi-probability values to test its prediction for each of these verbs.
- By virtue of its bidirectional processing of input, BERT conditions its predicted scores on both the left and right contexts of the masked verb.
- In 9 of the 10 syntactic constructions BERT approaches or surpasses Marvin and Linzen's reported human judgment accuracy, in some cases by a significant margin.
- In the one exception, sentential complements, BERT Base scores 0.83, and BERT Large 0.86, while human accuracy is 0.93.

Applying BERT to the Marvin-Linzen Test Set

- Goldberg (2019) uses BERT (Base and Large, both untuned) as a LM on Marvin and Linzen's test set.
- He masks the contrasting verbs in each minimal pair, and uses BERT's pre-softmax logit scores as quasi-probability values to test its prediction for each of these verbs.
- By virtue of its bidirectional processing of input, BERT conditions its predicted scores on both the left and right contexts of the masked verb.
- In 9 of the 10 syntactic constructions BERT approaches or surpasses Marvin and Linzen's reported human judgment accuracy, in some cases by a significant margin.
- In the one exception, sentential complements, BERT Base scores 0.83, and BERT Large 0.86, while human accuracy is 0.93.

Applying BERT to the Marvin-Linzen Test Set

- Goldberg (2019) uses BERT (Base and Large, both untuned) as a LM on Marvin and Linzen's test set.
- He masks the contrasting verbs in each minimal pair, and uses BERT's pre-softmax logit scores as quasi-probability values to test its prediction for each of these verbs.
- By virtue of its bidirectional processing of input, BERT conditions its predicted scores on both the left and right contexts of the masked verb.
- In 9 of the 10 syntactic constructions BERT approaches or surpasses Marvin and Linzen's reported human judgment accuracy, in some cases by a significant margin.
- In the one exception, sentential complements, BERT Base scores 0.83, and BERT Large 0.86, while human accuracy is 0.93.

Recurrent Neural Network Grammar LMs

- Kuncoro et al. (2018) apply Recurrent Neural Network Grammars (RNNGs, Dyer et al., 2016) to Linzen et al.'s (2016) test set.
- Their RNNG LM contains a stack and a composition operator for predicting the constituency structure of a string.
- It assigns joint probability to a string and a phrase structure.
- This RNNG and their best sequential LSTM LM achieve approximately the same level of accuracy in predicting agreement for sentences with 0-2 attractors (98%, 96.5%, and 95%, respectively).
- However, the RNNG outperforms the LSTM LM for cases with higher numbers of attractors, with 93% vs 87% at 4 attractors, and 88% vs 82% at 5 attractors).

Recurrent Neural Network Grammar LMs

- Kuncoro et al. (2018) apply Recurrent Neural Network Grammars (RNNGs, Dyer et al., 2016) to Linzen et al.'s (2016) test set.
- Their RNNG LM contains a stack and a composition operator for predicting the constituency structure of a string.
- It assigns joint probability to a string and a phrase structure.
- This RNNG and their best sequential LSTM LM achieve approximately the same level of accuracy in predicting agreement for sentences with 0-2 attractors (98%, 96.5%, and 95%, respectively).
- However, the RNNG outperforms the LSTM LM for cases with higher numbers of attractors, with 93% vs 87% at 4 attractors, and 88% vs 82% at 5 attractors).

Recurrent Neural Network Grammar LMs

- Kuncoro et al. (2018) apply Recurrent Neural Network Grammars (RNNGs, Dyer et al., 2016) to Linzen et al.'s (2016) test set.
- Their RNNG LM contains a stack and a composition operator for predicting the constituency structure of a string.
- It assigns joint probability to a string and a phrase structure.
- This RNNG and their best sequential LSTM LM achieve approximately the same level of accuracy in predicting agreement for sentences with 0-2 attractors (98%, 96.5%, and 95%, respectively).
- However, the RNNG outperforms the LSTM LM for cases with higher numbers of attractors, with 93% vs 87% at 4 attractors, and 88% vs 82% at 5 attractors).

Recurrent Neural Network Grammar LMs

- Kuncoro et al. (2018) apply Recurrent Neural Network Grammars (RNNGs, Dyer et al., 2016) to Linzen et al.'s (2016) test set.
- Their RNNG LM contains a stack and a composition operator for predicting the constituency structure of a string.
- It assigns joint probability to a string and a phrase structure.
- This RNNG and their best sequential LSTM LM achieve approximately the same level of accuracy in predicting agreement for sentences with 0-2 attractors (98%, 96.5%, and 95%, respectively).
- However, the RNNG outperforms the LSTM LM for cases with higher numbers of attractors, with 93% vs 87% at 4 attractors, and 88% vs 82% at 5 attractors).

Recurrent Neural Network Grammar LMs

- Kuncoro et al. (2018) apply Recurrent Neural Network Grammars (RNNGs, Dyer et al., 2016) to Linzen et al.'s (2016) test set.
- Their RNNG LM contains a stack and a composition operator for predicting the constituency structure of a string.
- It assigns joint probability to a string and a phrase structure.
- This RNNG and their best sequential LSTM LM achieve approximately the same level of accuracy in predicting agreement for sentences with 0-2 attractors (98%, 96.5%, and 95%, respectively).
- However, the RNNG outperforms the LSTM LM for cases with higher numbers of attractors, with 93% vs 87% at 4 attractors, and 88% vs 82% at 5 attractors).

Syntactic Structure as an Architectural Bias

- An RNNG is designed to induce hierarchical structure on a string.
- The fact that it outperforms an LSTM lacking this architectural feature, on the agreement task, for more complex cases, suggests that it is better able to learn long distance syntactic dependencies.
- However, Kuncoro et al. only test their RNNG LM on a limited set of English data.
- Gulordava et al.'s experiments indicate that simple LSTMs achieve very high accuracy on the agreement task, for morphologically richer languages, and they replicate human performance for Italian.

Syntactic Structure as an Architectural Bias

- An RNNG is designed to induce hierarchical structure on a string.
- The fact that it outperforms an LSTM lacking this architectural feature, on the agreement task, for more complex cases, suggests that it is better able to learn long distance syntactic dependencies.
- However, Kuncoro et al. only test their RNNG LM on a limited set of English data.
- Gulordava et al.'s experiments indicate that simple LSTMs achieve very high accuracy on the agreement task, for morphologically richer languages, and they replicate human performance for Italian.

Syntactic Structure as an Architectural Bias

- An RNNG is designed to induce hierarchical structure on a string.
- The fact that it outperforms an LSTM lacking this architectural feature, on the agreement task, for more complex cases, suggests that it is better able to learn long distance syntactic dependencies.
- However, Kuncoro et al. only test their RNNG LM on a limited set of English data.
- Gulordava et al.'s experiments indicate that simple LSTMs achieve very high accuracy on the agreement task, for morphologically richer languages, and they replicate human performance for Italian.

Syntactic Structure as an Architectural Bias

- An RNNG is designed to induce hierarchical structure on a string.
- The fact that it outperforms an LSTM lacking this architectural feature, on the agreement task, for more complex cases, suggests that it is better able to learn long distance syntactic dependencies.
- However, Kuncoro et al. only test their RNNG LM on a limited set of English data.
- Gulordava et al.'s experiments indicate that simple LSTMs achieve very high accuracy on the agreement task, for morphologically richer languages, and they replicate human performance for Italian.

Distilled Syntax-Aware LSTM LMs

- Kuncoro et al. (2019) observe that RNNGs do not scale up to large training sets, because of the high computational cost of training them to predict both strings and parse structures for large corpora.
- As an alternative they use an RNNG as a teacher model to supervise the training of a sequential LSTM over large data sets (knowledge distallation).
- This appears to induce the RNNG's hierarchical structural bias in the squential LSTM LM's predictions.

Distilled Syntax-Aware LSTM LMs

- Kuncoro et al. (2019) observe that RNNGs do not scale up to large training sets, because of the high computational cost of training them to predict both strings and parse structures for large corpora.

- As an alternative they use an RNNG as a teacher model to supervise the training of a sequential LSTM over large data sets (knowledge distallation).

- This appears to induce the RNNG's hierarchical structural bias in the squential LSTM LM's predictions.

Distilled Syntax-Aware LSTM LMs

- Kuncoro et al. (2019) observe that RNNGs do not scale up to large training sets, because of the high computational cost of training them to predict both strings and parse structures for large corpora.

- As an alternative they use an RNNG as a teacher model to supervise the training of a sequential LSTM over large data sets (knowledge distallation).

- This appears to induce the RNNG's hierarchical structural bias in the squential LSTM LM's predictions.

Distilled Syntax-Aware LSTM LMs

- Kuncoro et al. (2019) report that their Distilled Syntax-Aware LSTM (DSA-LSTM) scores, on average, 90% on Gulordava et al.'s English subject-verb agreement test set.
- Their non-DSA-LSTM obtains 87%, and BERT 89%.
- These scores are comparable, and they are all slightly above average human performance for this test set, which they give as 86%.
- This suggests that, with sufficient training data, both sequential LSTMs and bidirectional transformers can achieve high accuracy on the agreement task, even without prior hierarchical structure bias.

Distilled Syntax-Aware LSTM LMs

- Kuncoro et al. (2019) report that their Distilled Syntax-Aware LSTM (DSA-LSTM) scores, on average, 90% on Gulordava et al.'s English subject-verb agreement test set.
- Their non-DSA-LSTM obtains 87%, and BERT 89%.
- These scores are comparable, and they are all slightly above average human performance for this test set, which they give as 86%.
- This suggests that, with sufficient training data, both sequential LSTMs and bidirectional transformers can achieve high accuracy on the agreement task, even without prior hierarchical structure bias.

Distilled Syntax-Aware LSTM LMs

- Kuncoro et al. (2019) report that their Distilled Syntax-Aware LSTM (DSA-LSTM) scores, on average, 90% on Gulordava et al.'s English subject-verb agreement test set.
- Their non-DSA-LSTM obtains 87%, and BERT 89%.
- These scores are comparable, and they are all slightly above average human performance for this test set, which they give as 86%.
- This suggests that, with sufficient training data, both sequential LSTMs and bidirectional transformers can achieve high accuracy on the agreement task, even without prior hierarchical structure bias.

Distilled Syntax-Aware LSTM LMs

- Kuncoro et al. (2019) report that their Distilled Syntax-Aware LSTM (DSA-LSTM) scores, on average, 90% on Gulordava et al.'s English subject-verb agreement test set.
- Their non-DSA-LSTM obtains 87%, and BERT 89%.
- These scores are comparable, and they are all slightly above average human performance for this test set, which they give as 86%.
- This suggests that, with sufficient training data, both sequential LSTMs and bidirectional transformers can achieve high accuracy on the agreement task, even without prior hierarchical structure bias.

Tree RNNs

- TreeRNNs (Socher el al., 2011; Bowman et al., 2016) are trained to assign syntactic trees to input sentences by supervised learning on parse structure annotations.
- These models have achieved improved performance on tasks like natural language inference (NLI) and sentiment analysis.
- Latent tree RNNs (Yogatama et al., 2017; Maillard et al., 2017; Choi et al., 2018) learn to induce tree representations without supervised learning on parse annotations.
- Williams et al. (2018a) test several TreeRNNs, supervised and latent, as well as a baseline non-tree LSTM, on two NLI test sets: the Stanford NLI set (Bowman et al., 2015), and the MultiNLI corpora (Williams et al., 2018).

Tree RNNs

- TreeRNNs (Socher el al., 2011; Bowman et al., 2016) are trained to assign syntactic trees to input sentences by supervised learning on parse structure annotations.
- These models have achieved improved performance on tasks like natural language inference (NLI) and sentiment analysis.
- Latent tree RNNs (Yogatama et al., 2017; Maillard et al., 2017; Choi et al., 2018) learn to induce tree representations without supervised learning on parse annotations.
- Williams et al. (2018a) test several TreeRNNs, supervised and latent, as well as a baseline non-tree LSTM, on two NLI test sets: the Stanford NLI set (Bowman et al., 2015), and the MultiNLI corpora (Williams et al., 2018).

Tree RNNs

- TreeRNNs (Socher el al., 2011; Bowman et al., 2016) are trained to assign syntactic trees to input sentences by supervised learning on parse structure annotations.
- These models have achieved improved performance on tasks like natural language inference (NLI) and sentiment analysis.
- Latent tree RNNs (Yogatama et al., 2017; Maillard et al., 2017; Choi et al., 2018) learn to induce tree representations without supervised learning on parse annotations.
- Williams et al. (2018a) test several TreeRNNs, supervised and latent, as well as a baseline non-tree LSTM, on two NLI test sets: the Stanford NLI set (Bowman et al., 2015), and the MultiNLI corpora (Williams et al., 2018).

Tree RNNs

- TreeRNNs (Socher el al., 2011; Bowman et al., 2016) are trained to assign syntactic trees to input sentences by supervised learning on parse structure annotations.
- These models have achieved improved performance on tasks like natural language inference (NLI) and sentiment analysis.
- Latent tree RNNs (Yogatama et al., 2017; Maillard et al., 2017; Choi et al., 2018) learn to induce tree representations without supervised learning on parse annotations.
- Williams et al. (2018a) test several TreeRNNs, supervised and latent, as well as a baseline non-tree LSTM, on two NLI test sets: the Stanford NLI set (Bowman et al., 2015), and the MultiNLI corpora (Williams et al., 2018).

Williams et al.'s (2018a) Results

- Choi et al.'s (2018) latent tree RNN (ST-Gumbel) outperformed all other systems, scoring accuracy rates of 83.7 on SNLI, and 69.5 on MNLI.
- The other systems scored between 81.3 and 82.6 on SNLI, and between 66.2 and 69.1 on MNLI.
- The non-tree LSTM achieved the second highest rate of accuracy, with 82.6 on SNLI, and 69.1 on MNLI.
- Williams et al. (2018a) observe that Choi et al.'s ST-Gumbel model, and the other latent tree RNNs that they test, yield shallow parse structures, which are not consistent across test sentences.

Williams et al.'s (2018a) Results

- Choi et al.'s (2018) latent tree RNN (ST-Gumbel) outperformed all other systems, scoring accuracy rates of 83.7 on SNLI, and 69.5 on MNLI.
- The other systems scored between 81.3 and 82.6 on SNLI, and between 66.2 and 69.1 on MNLI.
- The non-tree LSTM achieved the second highest rate of accuracy, with 82.6 on SNLI, and 69.1 on MNLI.
- Williams et al. (2018a) observe that Choi et al.'s ST-Gumbel model, and the other latent tree RNNs that they test, yield shallow parse structures, which are not consistent across test sentences.

Williams et al.'s (2018a) Results

- Choi et al.'s (2018) latent tree RNN (ST-Gumbel) outperformed all other systems, scoring accuracy rates of 83.7 on SNLI, and 69.5 on MNLI.
- The other systems scored between 81.3 and 82.6 on SNLI, and between 66.2 and 69.1 on MNLI.
- The non-tree LSTM achieved the second highest rate of accuracy, with 82.6 on SNLI, and 69.1 on MNLI.
- Williams et al. (2018a) observe that Choi et al.'s ST-Gumbel model, and the other latent tree RNNs that they test, yield shallow parse structures, which are not consistent across test sentences.

Williams et al.'s (2018a) Results

- Choi et al.'s (2018) latent tree RNN (ST-Gumbel) outperformed all other systems, scoring accuracy rates of 83.7 on SNLI, and 69.5 on MNLI.
- The other systems scored between 81.3 and 82.6 on SNLI, and between 66.2 and 69.1 on MNLI.
- The non-tree LSTM achieved the second highest rate of accuracy, with 82.6 on SNLI, and 69.1 on MNLI.
- Williams et al. (2018a) observe that Choi et al.'s ST-Gumbel model, and the other latent tree RNNs that they test, yield shallow parse structures, which are not consistent across test sentences.

Significance of Tree Representations in the NLI Task

- Williams et al. (2018a) also point out that the elements of these parses do not correspond to the constituents of either the Penn Tree Bank (Marcus et al., 1999), or those of formal syntactic theories.
- Given these results, and the fact that the non-tree LSTM baseline achieves results comparable to Choi et al.'s model, it is unclear to what extent tree representations, latent or supervised, contribute to the NLI task.
- It is necessary to test other DNN architectures, both with and without parse representations, across a variety of NLP tasks, to determine whether tree structures enhance performance on these tasks.

Significance of Tree Representations in the NLI Task

- Williams et al. (2018a) also point out that the elements of these parses do not correspond to the constituents of either the Penn Tree Bank (Marcus et al., 1999), or those of formal syntactic theories.
- Given these results, and the fact that the non-tree LSTM baseline achieves results comparable to Choi et al.'s model, it is unclear to what extent tree representations, latent or supervised, contribute to the NLI task.
- It is necessary to test other DNN architectures, both with and without parse representations, across a variety of NLP tasks, to determine whether tree structures enhance performance on these tasks.

Significance of Tree Representations in the NLI Task

- Williams et al. (2018a) also point out that the elements of these parses do not correspond to the constituents of either the Penn Tree Bank (Marcus et al., 1999), or those of formal syntactic theories.
- Given these results, and the fact that the non-tree LSTM baseline achieves results comparable to Choi et al.'s model, it is unclear to what extent tree representations, latent or supervised, contribute to the NLI task.
- It is necessary to test other DNN architectures, both with and without parse representations, across a variety of NLP tasks, to determine whether tree structures enhance performance on these tasks.

Implicit Representation of Trees in Transformers

- Hewitt and Manning (2019) (HM) use a supervised probe to test for the presence of syntactic tree structures in the word vectors of DNNs.
- The probe consists of a squared L2 distance measure, which determines the distance (number of nodes) between word pairs in a tree, and a squared L2 norm, which identifies the relative depth (chain of nodes from the root) of each word in a tree.
- This metric defines a mapping from the word vectors in the sentence representation of a DNN into a parse tree structure.
- HM test their probe on the word representations of the unlabelled Stanford Dependency parses (de Marneffe et al., 2006) for the parsing train/dev/test splits of the Penn Treebank (PTB).

Implicit Representation of Trees in Transformers

- Hewitt and Manning (2019) (HM) use a supervised probe to test for the presence of syntactic tree structures in the word vectors of DNNs.
- The probe consists of a squared L2 distance measure, which determines the distance (number of nodes) between word pairs in a tree, and a squared L2 norm, which identifies the relative depth (chain of nodes from the root) of each word in a tree.
- This metric defines a mapping from the word vectors in the sentence representation of a DNN into a parse tree structure.
- HM test their probe on the word representations of the unlabelled Stanford Dependency parses (de Marneffe et al., 2006) for the parsing train/dev/test splits of the Penn Treebank (PTB).

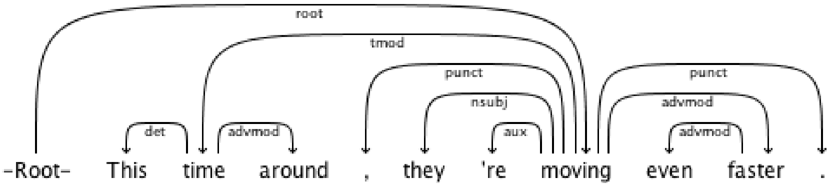
Implicit Representation of Trees in Transformers

- Hewitt and Manning (2019) (HM) use a supervised probe to test for the presence of syntactic tree structures in the word vectors of DNNs.
- The probe consists of a squared L2 distance measure, which determines the distance (number of nodes) between word pairs in a tree, and a squared L2 norm, which identifies the relative depth (chain of nodes from the root) of each word in a tree.
- This metric defines a mapping from the word vectors in the sentence representation of a DNN into a parse tree structure.
- HM test their probe on the word representations of the unlabelled Stanford Dependency parses (de Marneffe et al., 2006) for the parsing train/dev/test splits of the Penn Treebank (PTB).

Implicit Representation of Trees in Transformers

- Hewitt and Manning (2019) (HM) use a supervised probe to test for the presence of syntactic tree structures in the word vectors of DNNs.
- The probe consists of a squared L2 distance measure, which determines the distance (number of nodes) between word pairs in a tree, and a squared L2 norm, which identifies the relative depth (chain of nodes from the root) of each word in a tree.
- This metric defines a mapping from the word vectors in the sentence representation of a DNN into a parse tree structure.
- HM test their probe on the word representations of the unlabelled Stanford Dependency parses (de Marneffe et al., 2006) for the parsing train/dev/test splits of the Penn Treebank (PTB).

Stanford Dependency Parser



Labelled dependency parse tree from

`https://nlp.stanford.edu/software/nndep.html`

Transformer Accuracy on Parse Prediction

- HM find that ELMO and BERT significantly outperform baseline systems in predicting unlabelled dependency parse trees for the PTB training set.
- ELM01 scores an average Spearman correlation of 0.87 on word distances, and 0.87 on word depth.
- $\text{BERT}_{LARGE15}$ achieves an average Spearman correlation of 0.87 on distance, and 0.89 on depth.
- These results suggest that the lexical embeddings of large pre-trained transformers incorporate consistent hierarchical syntactic information corresponding to dependency parse trees.

Transformer Accuracy on Parse Prediction

- HM find that ELMO and BERT significantly outperform baseline systems in predicting unlabelled dependency parse trees for the PTB training set.
- ELM01 scores an average Spearman correlation of 0.87 on word distances, and 0.87 on word depth.
- $\text{BERT}_{LARGE15}$ achieves an average Spearman correlation of 0.87 on distance, and 0.89 on depth.
- These results suggest that the lexical embeddings of large pre-trained transformers incorporate consistent hierarchical syntactic information corresponding to dependency parse trees.

Transformer Accuracy on Parse Prediction

- HM find that ELMO and BERT significantly outperform baseline systems in predicting unlabelled dependency parse trees for the PTB training set.
- ELM01 scores an average Spearman correlation of 0.87 on word distances, and 0.87 on word depth.
- $\text{BERT}_{LARGE15}$ achieves an average Spearman correlation of 0.87 on distance, and 0.89 on depth.
- These results suggest that the lexical embeddings of large pre-trained transformers incorporate consistent hierarchical syntactic information corresponding to dependency parse trees.

Transformer Accuracy on Parse Prediction

- HM find that ELMO and BERT significantly outperform baseline systems in predicting unlabelled dependency parse trees for the PTB training set.
- ELM01 scores an average Spearman correlation of 0.87 on word distances, and 0.87 on word depth.
- $\text{BERT}_{LARGE15}$ achieves an average Spearman correlation of 0.87 on distance, and 0.89 on depth.
- These results suggest that the lexical embeddings of large pre-trained transformers incorporate consistent hierarchical syntactic information corresponding to dependency parse trees.

Conclusions

- The experimental work considered here strongly indicates that DNNs learn a considerable amount of hierarchical syntactic structure.
- This information is implicit in the distributed lexical embeddings of a DNN, and it is encoded in the vector representations of the words, phrases, and sentences that the DNN produces.
- Adding explicit syntactic annotations to the training data of deep learning models does not seem to enhance their performance on tasks requiring syntactic knowledge.
- The extent of similarity between deep learning of syntactic structure, and the way in which humans acquire and represent this structure, remains an open question.

Conclusions

- The experimental work considered here strongly indicates that DNNs learn a considerable amount of hierarchical syntactic structure.
- This information is implicit in the distributed lexical embeddings of a DNN, and it is encoded in the vector representations of the words, phrases, and sentences that the DNN produces.
- Adding explicit syntactic annotations to the training data of deep learning models does not seem to enhance their performance on tasks requiring syntactic knowledge.
- The extent of similarity between deep learning of syntactic structure, and the way in which humans acquire and represent this structure, remains an open question.

Conclusions

- The experimental work considered here strongly indicates that DNNs learn a considerable amount of hierarchical syntactic structure.
- This information is implicit in the distributed lexical embeddings of a DNN, and it is encoded in the vector representations of the words, phrases, and sentences that the DNN produces.
- Adding explicit syntactic annotations to the training data of deep learning models does not seem to enhance their performance on tasks requiring syntactic knowledge.
- The extent of similarity between deep learning of syntactic structure, and the way in which humans acquire and represent this structure, remains an open question.

Conclusions

- The experimental work considered here strongly indicates that DNNs learn a considerable amount of hierarchical syntactic structure.
- This information is implicit in the distributed lexical embeddings of a DNN, and it is encoded in the vector representations of the words, phrases, and sentences that the DNN produces.
- Adding explicit syntactic annotations to the training data of deep learning models does not seem to enhance their performance on tasks requiring syntactic knowledge.
- The extent of similarity between deep learning of syntactic structure, and the way in which humans acquire and represent this structure, remains an open question.