Machine Learning and The Sentence Acceptability Task

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 15, 2020

Outline

Grammaticality and Probability

- Grammaticality cannot be directly reduced to probability (Clark and Lappin, 2011).
- The probability of a sentence is the likelihood of its occurrence.
- Long, complex grammatical sentences with infrequent lexical items are less probable than some short ill-formed sentences containing only frequent words.
- There is, however, a connection between grammaticality and probability, as grammatical sentences are generally more likely to occur than ill-formed ones.

Grammaticality and Probability

- Grammaticality cannot be directly reduced to probability (Clark and Lappin, 2011).
- The probability of a sentence is the likelihood of its occurrence.
- Long, complex grammatical sentences with infrequent lexical items are less probable than some short ill-formed sentences containing only frequent words.
- There is, however, a connection between grammaticality and probability, as grammatical sentences are generally more likely to occur than ill-formed ones.

Grammaticality and Probability

- Grammaticality cannot be directly reduced to probability (Clark and Lappin, 2011).
- The probability of a sentence is the likelihood of its occurrence.
- Long, complex grammatical sentences with infrequent lexical items are less probable than some short ill-formed sentences containing only frequent words.
- There is, however, a connection between grammaticality and probability, as grammatical sentences are generally more likely to occur than ill-formed ones.

Grammaticality and Probability

- Grammaticality cannot be directly reduced to probability (Clark and Lappin, 2011).
- The probability of a sentence is the likelihood of its occurrence.
- Long, complex grammatical sentences with infrequent lexical items are less probable than some short ill-formed sentences containing only frequent words.
- There is, however, a connection between grammaticality and probability, as grammatical sentences are generally more likely to occur than ill-formed ones.

Grammaticality and Acceptability

- Grammaticality is a theoretical property which is not directly accessible to observation.
- Speakers' acceptability judgements can be observed and measured.
- These judgements provide the primary data for most linguistic theories.
- An adequate theory of syntactic knowledge must be able to account for the observed data of acceptability judgements.

Grammaticality and Acceptability

- Grammaticality is a theoretical property which is not directly accessible to observation.
- Speakers' acceptability judgements can be observed and measured.
- These judgements provide the primary data for most linguistic theories.
- An adequate theory of syntactic knowledge must be able to account for the observed data of acceptability judgements.

Grammaticality and Acceptability

- Grammaticality is a theoretical property which is not directly accessible to observation.
- Speakers' acceptability judgements can be observed and measured.
- These judgements provide the primary data for most linguistic theories.
- An adequate theory of syntactic knowledge must be able to account for the observed data of acceptability judgements.

Grammaticality and Acceptability

- Grammaticality is a theoretical property which is not directly accessible to observation.
- Speakers' acceptability judgements can be observed and measured.
- These judgements provide the primary data for most linguistic theories.
- An adequate theory of syntactic knowledge must be able to account for the observed data of acceptability judgements.

Evidence for Gradience in Sentence Acceptability

- Lau, Clark, and Lappin (2014, 2015, 2017) (LCL) present extensive experimental evidence for gradience in human sentence acceptability judgements.
- They show that crowd sourced judgements on round trip machine translated sentences from the British National Corpus (BNC) exhibit both aggregate (mean) and individual gradience.
- They also demonstrate that crowd sourced judgements on linguists' examples from Adger (2003), in which semantic/pragmatic anomaly has been filtered out, display the same sort of gradience.
- LCL find a high Pearson correlation (≥ 0.92) between the three modes of presentation (binary, 4 categories of naturalness, sliding scale with 100 underlying points) that they use.

Evidence for Gradience in Sentence Acceptability

- Lau, Clark, and Lappin (2014, 2015, 2017) (LCL) present extensive experimental evidence for gradience in human sentence acceptability judgements.
- They show that crowd sourced judgements on round trip machine translated sentences from the British National Corpus (BNC) exhibit both aggregate (mean) and individual gradience.
- They also demonstrate that crowd sourced judgements on linguists' examples from Adger (2003), in which semantic/pragmatic anomaly has been filtered out, display the same sort of gradience.
- LCL find a high Pearson correlation (≥ 0.92) between the three modes of presentation (binary, 4 categories of naturalness, sliding scale with 100 underlying points) that they use.

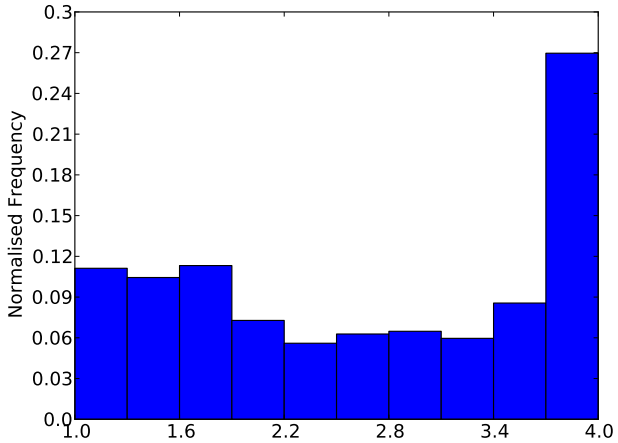
Evidence for Gradience in Sentence Acceptability

- Lau, Clark, and Lappin (2014, 2015, 2017) (LCL) present extensive experimental evidence for gradience in human sentence acceptability judgements.
- They show that crowd sourced judgements on round trip machine translated sentences from the British National Corpus (BNC) exhibit both aggregate (mean) and individual gradience.
- They also demonstrate that crowd sourced judgements on linguists' examples from Adger (2003), in which semantic/pragmatic anomaly has been filtered out, display the same sort of gradience.
- LCL find a high Pearson correlation (≥ 0.92) between the three modes of presentation (binary, 4 categories of naturalness, sliding scale with 100 underlying points) that they use.

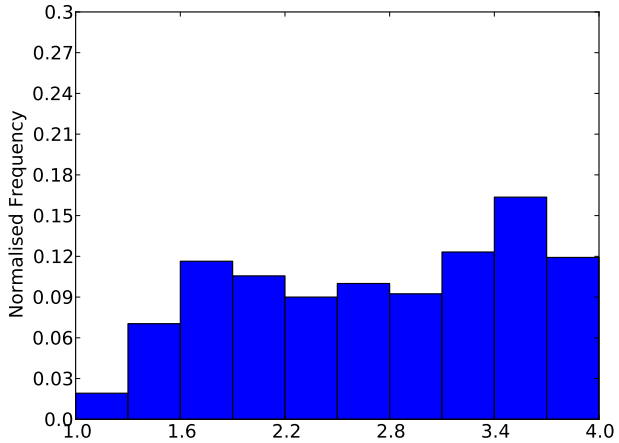
Evidence for Gradience in Sentence Acceptability

- Lau, Clark, and Lappin (2014, 2015, 2017) (LCL) present extensive experimental evidence for gradience in human sentence acceptability judgements.
- They show that crowd sourced judgements on round trip machine translated sentences from the British National Corpus (BNC) exhibit both aggregate (mean) and individual gradience.
- They also demonstrate that crowd sourced judgements on linguists' examples from Adger (2003), in which semantic/pragmatic anomaly has been filtered out, display the same sort of gradience.
- LCL find a high Pearson correlation (≥ 0.92) between the three modes of presentation (binary, 4 categories of naturalness, sliding scale with 100 underlying points) that they use.

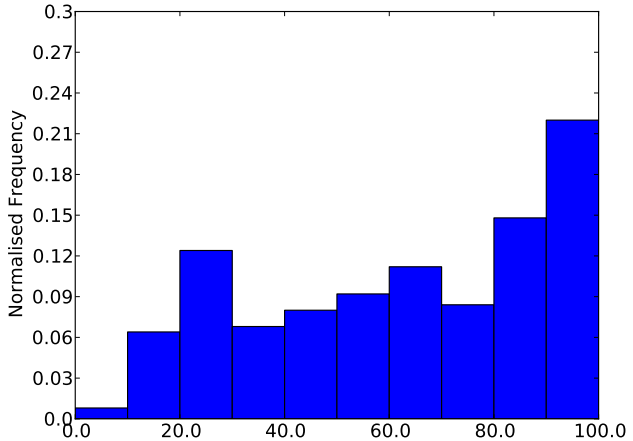
Histograms for Mean Sentence Ratings in the Three Modes of Presentation



(a) binary

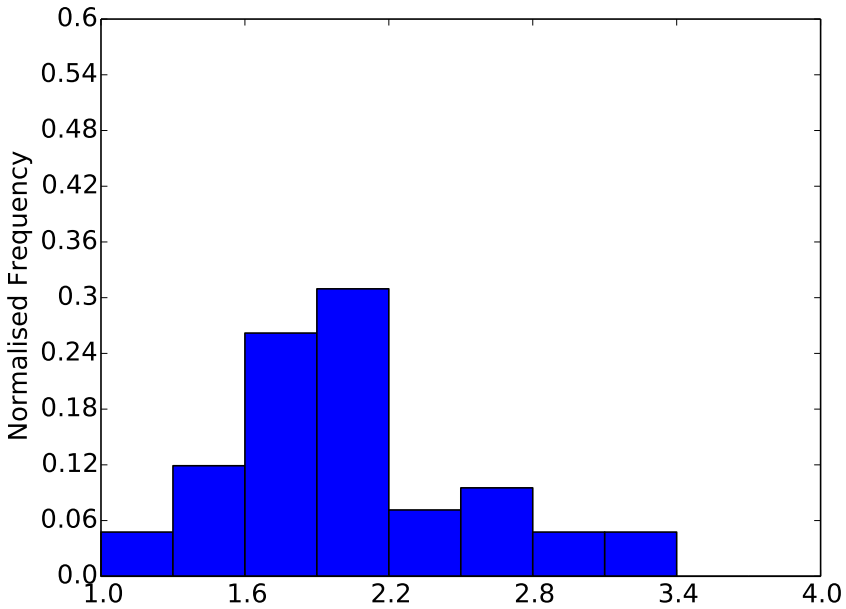


(b) 4 category

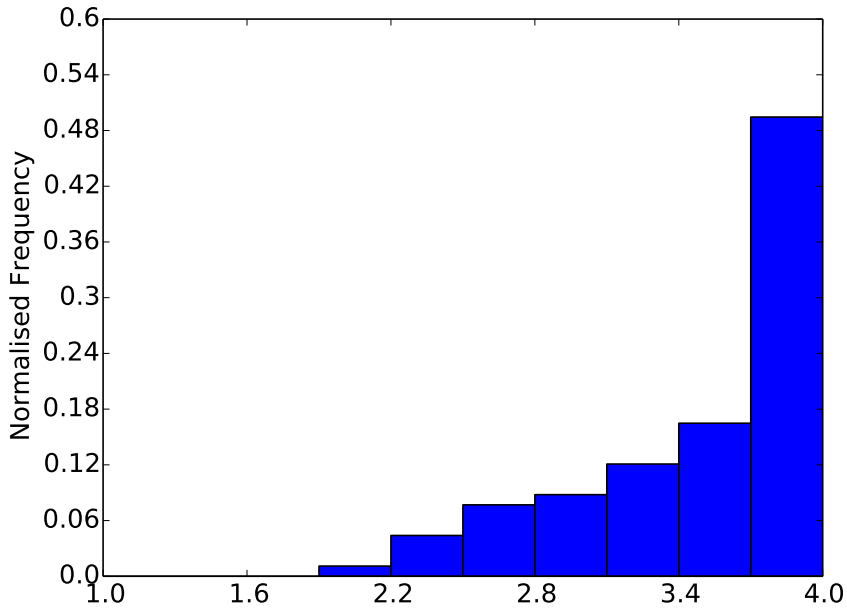


(c) slider

Semantically/Pragmatically Filtered Adger Sentence Mean Ratings

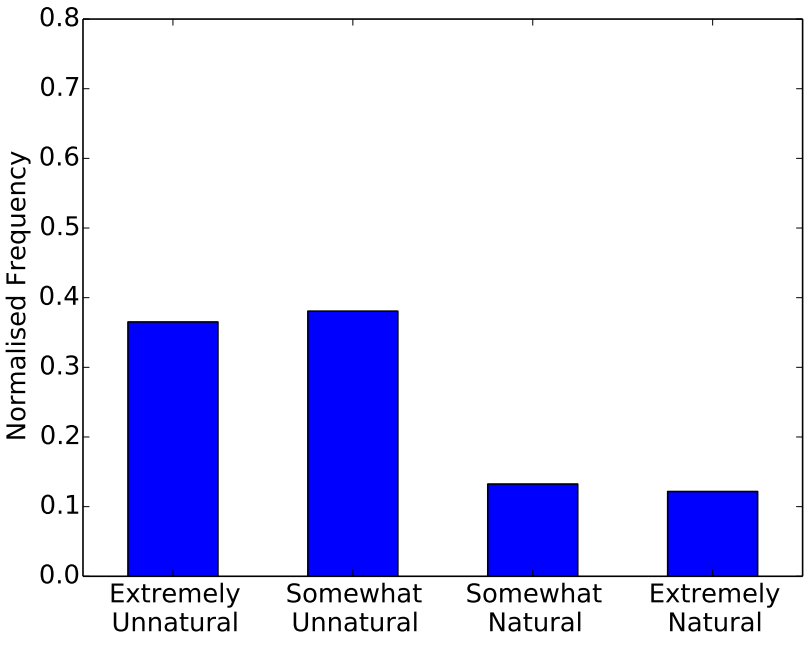


(a) Bad Sentences

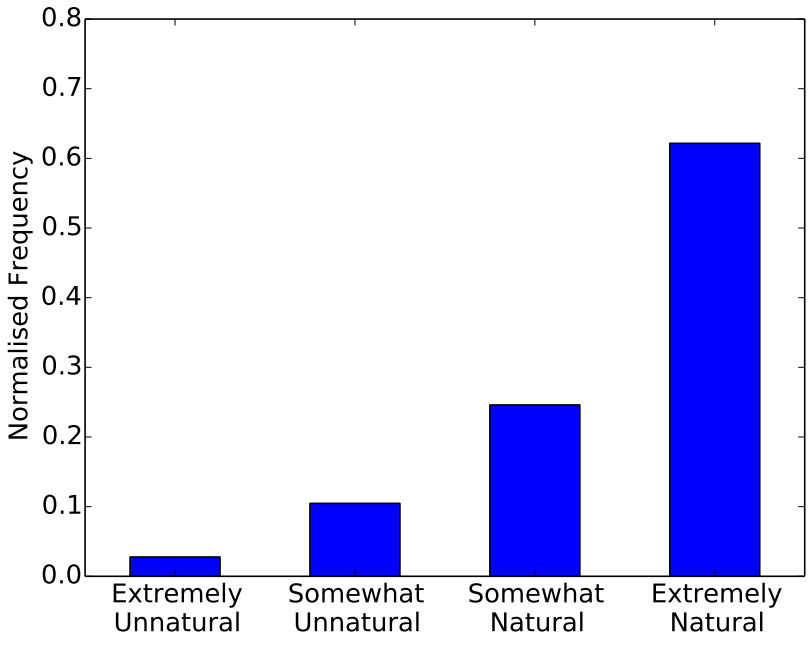


(b) Good Sentences

Semantically/Pragmatically Filtered Adger Sentence Individual Ratings



(a) Bad Sentences



(b) Good Sentences

Non-Linguistic Classifiers

- LCL also use AMT to crowdsource test two non-linguistic classifiers:
  1. Body weight (HITS with graphic representations of people of different sizes)
  2. Even and odd numbers (a HIT consisting of Armstrong et al.'s (1983) 21 natural numbers)

- The body weight judgements exhibit the same sort of gradience as the sentence acceptability results.

- By contrast, the even-odd judgements are sharply binary.

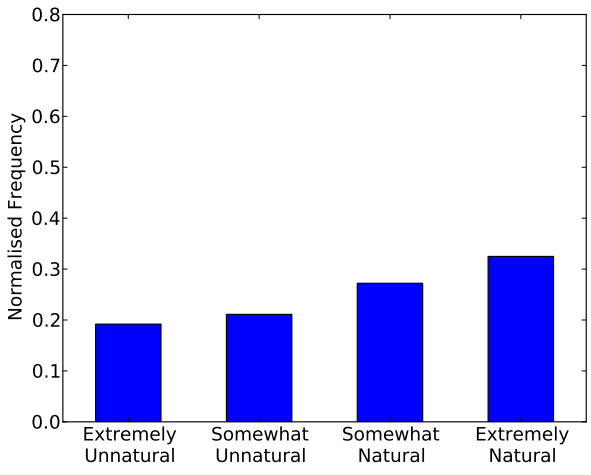
Non-Linguistic Classifiers

- LCL also use AMT to crowdsource test two non-linguistic classifiers:
  1. Body weight (HITS with graphic representations of people of different sizes)
  2. Even and odd numbers (a HIT consisting of Armstrong et al.'s (1983) 21 natural numbers)
- The body weight judgements exhibit the same sort of gradience as the sentence acceptability results.
- By contrast, the even-odd judgements are sharply binary.

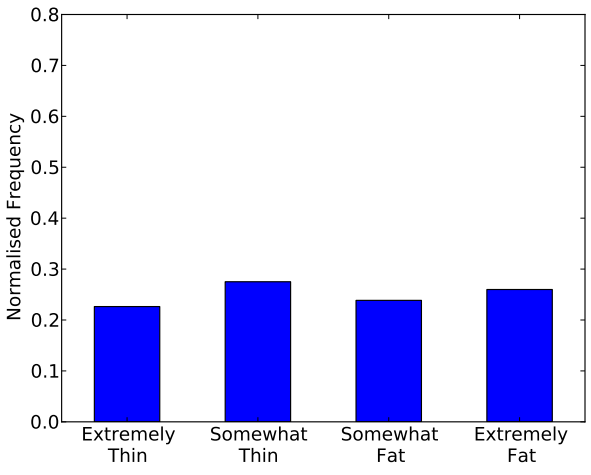
Non-Linguistic Classifiers

- LCL also use AMT to crowdsource test two non-linguistic classifiers:
  1. Body weight (HITS with graphic representations of people of different sizes)
  2. Even and odd numbers (a HIT consisting of Armstrong et al.'s (1983) 21 natural numbers)
- The body weight judgements exhibit the same sort of gradience as the sentence acceptability results.
- By contrast, the even-odd judgements are sharply binary.

Histograms of Individual Classifier Ratings: 4 Category Presentation

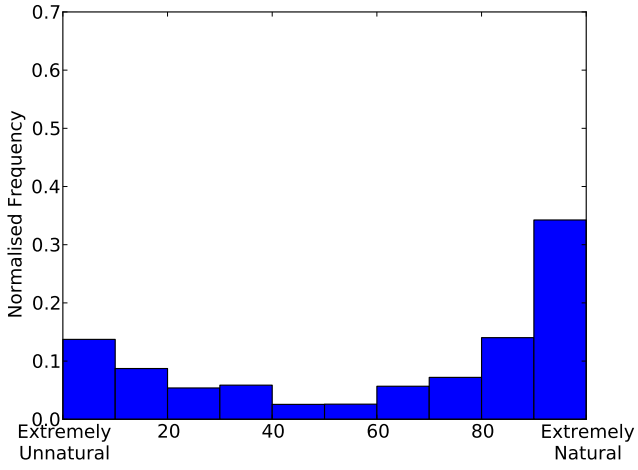


(a) Sentence Ratings

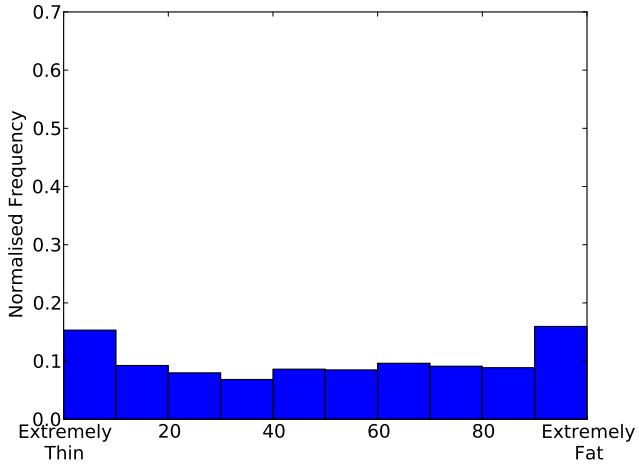


(b) Body Weight Ratings

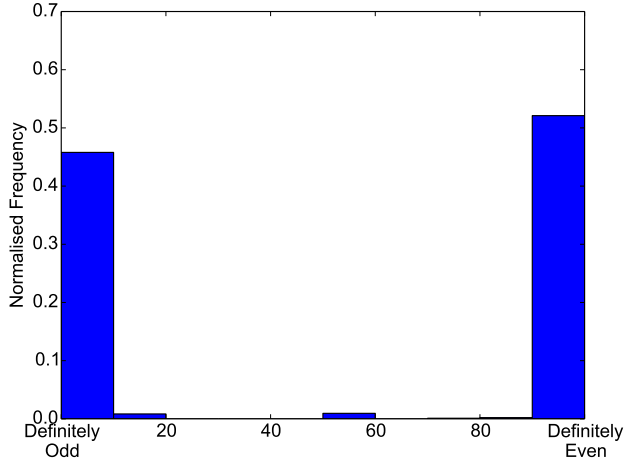
Histograms of Individual Classifier Ratings: Slider Presentation



(a) Sentence Ratings



(b) Body Weight Ratings



(c) Even-Odd Ratings

Testing Unsupervised Language Models

- LCL (2015, 2017) train a series of unsupervised language models on the full British National Corpus (BNC, 100m words).
- They apply these models to a crowd source (Amazon Mechanical Turk) annotated 2500 sentence test set obtained by round trip machine translation, through four languages, from BNC original sentences.
- The purpose of round-trip MT is to introduce a wide variety of infelicities into some of the sentences, and so insure variation in acceptability judgments across the examples of the set.
- They use a set of scoring functions to map the distributions that the models generate for the test set, into acceptability scores.
- LCL evaluate the accuracy of these scores through Pearson coefficient correlation with the mean speakers' judgements for the test set.

Testing Unsupervised Language Models

- LCL (2015, 2017) train a series of unsupervised language models on the full British National Corpus (BNC, 100m words).
- They apply these models to a crowd source (Amazon Mechanical Turk) annotated 2500 sentence test set obtained by round trip machine translation, through four languages, from BNC original sentences.
- The purpose of round-trip MT is to introduce a wide variety of infelicities into some of the sentences, and so insure variation in acceptability judgments across the examples of the set.
- They use a set of scoring functions to map the distributions that the models generate for the test set, into acceptability scores.
- LCL evaluate the accuracy of these scores through Pearson coefficient correlation with the mean speakers' judgements for the test set.

Testing Unsupervised Language Models

- LCL (2015, 2017) train a series of unsupervised language models on the full British National Corpus (BNC, 100m words).
- They apply these models to a crowd source (Amazon Mechanical Turk) annotated 2500 sentence test set obtained by round trip machine translation, through four languages, from BNC original sentences.
- The purpose of round-trip MT is to introduce a wide variety of infelicities into some of the sentences, and so insure variation in acceptability judgments across the examples of the set.
- They use a set of scoring functions to map the distributions that the models generate for the test set, into acceptability scores.
- LCL evaluate the accuracy of these scores through Pearson coefficient correlation with the mean speakers' judgements for the test set.

Testing Unsupervised Language Models

- LCL (2015, 2017) train a series of unsupervised language models on the full British National Corpus (BNC, 100m words).
- They apply these models to a crowd source (Amazon Mechanical Turk) annotated 2500 sentence test set obtained by round trip machine translation, through four languages, from BNC original sentences.
- The purpose of round-trip MT is to introduce a wide variety of infelicities into some of the sentences, and so insure variation in acceptability judgments across the examples of the set.
- They use a set of scoring functions to map the distributions that the models generate for the test set, into acceptability scores.
- LCL evaluate the accuracy of these scores through Pearson coefficient correlation with the mean speakers' judgements for the test set.

Testing Unsupervised Language Models

- LCL (2015, 2017) train a series of unsupervised language models on the full British National Corpus (BNC, 100m words).
- They apply these models to a crowd source (Amazon Mechanical Turk) annotated 2500 sentence test set obtained by round trip machine translation, through four languages, from BNC original sentences.
- The purpose of round-trip MT is to introduce a wide variety of infelicities into some of the sentences, and so insure variation in acceptability judgments across the examples of the set.
- They use a set of scoring functions to map the distributions that the models generate for the test set, into acceptability scores.
- LCL evaluate the accuracy of these scores through Pearson coefficient correlation with the mean speakers' judgements for the test set.

The Language Models

The primary classes of model that LCL experiment with are

- Lexical *n*-gram models (bigram, trigram, and 4-gram),
- A second-order Bayesian Hidden Markov Model (BHMM),
- A two-tier BHMM, and
- A recurrent neural network language model (RNNLM)
- For purposes of comparison they also test the Stanford PCFG parser.

The Language Models

The primary classes of model that LCL experiment with are

- Lexical *n*-gram models (bigram, trigram, and 4-gram),
- A second-order Bayesian Hidden Markov Model (BHMM),
- A two-tier BHMM, and
- A recurrent neural network language model (RNNLM)
- For purposes of comparison they also test the Stanford PCFG parser.

The Language Models

The primary classes of model that LCL experiment with are

- Lexical *n*-gram models (bigram, trigram, and 4-gram),
- A second-order Bayesian Hidden Markov Model (BHMM),
- A two-tier BHMM, and
- A recurrent neural network language model (RNNLM)
- For purposes of comparison they also test the Stanford PCFG parser.

The Language Models

The primary classes of model that LCL experiment with are

- Lexical *n*-gram models (bigram, trigram, and 4-gram),
- A second-order Bayesian Hidden Markov Model (BHMM),
- A two-tier BHMM, and
- A recurrent neural network language model (RNNLM)
- For purposes of comparison they also test the Stanford PCFG parser.

The Language Models

The primary classes of model that LCL experiment with are

- Lexical *n*-gram models (bigram, trigram, and 4-gram),
- A second-order Bayesian Hidden Markov Model (BHMM),
- A two-tier BHMM, and
- A recurrent neural network language model (RNNLM)
- For purposes of comparison they also test the Stanford PCFG parser.

Lexical *N*-gram Models and BHMMs

- In a lexical *n*-gram model, the probability of generating a word w_i depends on the preceding n words.
- A BHMM first generates the (latent) word class given its preceding word classes, and then it generates a word on the basis of the selected word class.
- A BHMM has two sets of multinomials: the state transition multinomials and the word emission multinomials.

(a) Lexical Trigram

(b) Bayesian HMM (2nd Order)

Lexical *N*-gram Models and BHMMs

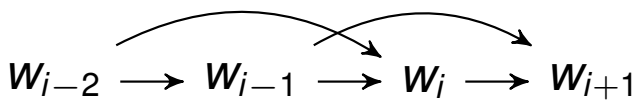
- In a lexical *n*-gram model, the probability of generating a word w_i depends on the preceding n words.
- A BHMM first generates the (latent) word class given its preceding word classes, and then it generates a word on the basis of the selected word class.
- A BHMM has two sets of multinomials: the state transition multinomials and the word emission multinomials.

(a) Lexical Trigram

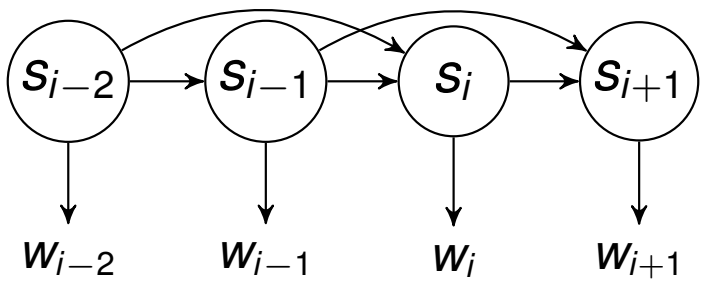
(b) Bayesian HMM (2nd Order)

Lexical *N*-gram Models and BHMMs

- In a lexical *n*-gram model, the probability of generating a word w_i depends on the preceding n words.
- A BHMM first generates the (latent) word class given its preceding word classes, and then it generates a word on the basis of the selected word class.
- A BHMM has two sets of multinomials: the state transition multinomials and the word emission multinomials.



(a) Lexical Trigram



(b) Bayesian HMM (2nd Order)

The RNNLM

- LCL use Mikolov et al.'s (2011) implementation of an RNNLM.
- Mikilov (2012) specifies an RNNLM that combines neural network learning with a Maximum Entropy (ME) component that learns direct connections among *N*-gram features.
- LCL find that the efficacy of the ME component declines as the number of neural units increases, with neural network performance rendering ME insignificant after 500 units.

The RNNLM

- LCL use Mikolov et al.'s (2011) implementation of an RNNLM.
- Mikilov (2012) specifies an RNNLM that combines neural network learning with a Maximum Entropy (ME) component that learns direct connections among *N*-gram features.
- LCL find that the efficacy of the ME component declines as the number of neural units increases, with neural network performance rendering ME insignificant after 500 units.

The RNNLM

- LCL use Mikolov et al.'s (2011) implementation of an RNNLM.
- Mikilov (2012) specifies an RNNLM that combines neural network learning with a Maximum Entropy (ME) component that learns direct connections among *N*-gram features.
- LCL find that the efficacy of the ME component declines as the number of neural units increases, with neural network performance rendering ME insignificant after 500 units.

The Stanford PCFG

- LCL focus on unsupervised models because of what they can show us about the limits of human learning.
- For purposes of comparison they also experiment with both the lexicalised and the unlexicalised Stanford PCFG parser (Klein and Manning, 2003a, 2003b), which is a supervised system.
- To compute the log probability of their test sentences, they use both the top-1 and the top-1K parses.
- The unlexicalised PCFG parser gives better performance, but LCL saw little difference between using the top-1 and the top-1K parses for computing log probability.

The Stanford PCFG

- LCL focus on unsupervised models because of what they can show us about the limits of human learning.
- For purposes of comparison they also experiment with both the lexicalised and the unlexicalised Stanford PCFG parser (Klein and Manning, 2003a, 2003b), which is a supervised system.
- To compute the log probability of their test sentences, they use both the top-1 and the top-1K parses.
- The unlexicalised PCFG parser gives better performance, but LCL saw little difference between using the top-1 and the top-1K parses for computing log probability.

The Stanford PCFG

- LCL focus on unsupervised models because of what they can show us about the limits of human learning.
- For purposes of comparison they also experiment with both the lexicalised and the unlexicalised Stanford PCFG parser (Klein and Manning, 2003a, 2003b), which is a supervised system.
- To compute the log probability of their test sentences, they use both the top-1 and the top-1K parses.
- The unlexicalised PCFG parser gives better performance, but LCL saw little difference between using the top-1 and the top-1K parses for computing log probability.

The Stanford PCFG

- LCL focus on unsupervised models because of what they can show us about the limits of human learning.
- For purposes of comparison they also experiment with both the lexicalised and the unlexicalised Stanford PCFG parser (Klein and Manning, 2003a, 2003b), which is a supervised system.
- To compute the log probability of their test sentences, they use both the top-1 and the top-1K parses.
- The unlexicalised PCFG parser gives better performance, but LCL saw little difference between using the top-1 and the top-1K parses for computing log probability.

Sentence Length, Lexical Frequency, and Grammatical Acceptability

- LCL use scoring functions to map sentence logprob values into acceptability scores.
- These functions are designed to eliminate the effect of sentence length and lexical frequency.
- LCL measure the effect of the scoring functions by computing their correlations with both properties.
- They also compute the correlations of these properties with human acceptability judgements.
  1. Corr. of human ratings and sentence length $= +0.13$.
  2. Corr. of human ratings and min word frequency $= +0.07$.
- These findings support the view that human grammatical acceptability judgements are not determined by either factor.

Sentence Length, Lexical Frequency, and Grammatical Acceptability

- LCL use scoring functions to map sentence logprob values into acceptability scores.
- These functions are designed to eliminate the effect of sentence length and lexical frequency.
- LCL measure the effect of the scoring functions by computing their correlations with both properties.
- They also compute the correlations of these properties with human acceptability judgements.
  1. Corr. of human ratings and sentence length $= +0.13$.
  2. Corr. of human ratings and min word frequency $= +0.07$.
- These findings support the view that human grammatical acceptability judgements are not determined by either factor.

Sentence Length, Lexical Frequency, and Grammatical Acceptability

- LCL use scoring functions to map sentence logprob values into acceptability scores.
- These functions are designed to eliminate the effect of sentence length and lexical frequency.
- LCL measure the effect of the scoring functions by computing their correlations with both properties.
- They also compute the correlations of these properties with human acceptability judgements.
  1. Corr. of human ratings and sentence length $= +0.13$.
  2. Corr. of human ratings and min word frequency $= +0.07$.
- These findings support the view that human grammatical acceptability judgements are not determined by either factor.

Sentence Length, Lexical Frequency, and Grammatical Acceptability

- LCL use scoring functions to map sentence logprob values into acceptability scores.
- These functions are designed to eliminate the effect of sentence length and lexical frequency.
- LCL measure the effect of the scoring functions by computing their correlations with both properties.
- They also compute the correlations of these properties with human acceptability judgements.
  1. Corr. of human ratings and sentence length $= +0.13$.
  2. Corr. of human ratings and min word frequency $= +0.07$.
- These findings support the view that human grammatical acceptability judgements are not determined by either factor.

Sentence Length, Lexical Frequency, and Grammatical Acceptability

- LCL use scoring functions to map sentence logprob values into acceptability scores.
- These functions are designed to eliminate the effect of sentence length and lexical frequency.
- LCL measure the effect of the scoring functions by computing their correlations with both properties.
- They also compute the correlations of these properties with human acceptability judgements.
  1. Corr. of human ratings and sentence length $= +0.13$.
  2. Corr. of human ratings and min word frequency $= +0.07$.
- These findings support the view that human grammatical acceptability judgements are not determined by either factor.

Acceptability Scoring Functions

Scoring Function	Equation
LogProb	$\log P_m(\xi)$
Mean LP	$\frac{\log P_m(\xi)}{\lvert\xi\rvert}$
Norm LP (Div)	$-\frac{\log P_m(\xi)}{\log P_u(\xi)}$
SLOR	$\frac{\log P_m(\xi) - \log P_u(\xi)}{\lvert\xi\rvert}$

$\xi =$ sentence;
$P_m(\xi) =$ the probability of the sentence given by the model;
$P_u(\xi) =$ is the unigram probability of sentence;
SLOR is proposed by Pauls and Klein (2012)

Word Log Prob Scoring Functions

- In addition to the sentence log probability scoring functions, LCL also experiment with *individual word log probability*.
- For each test sentence they extract the 5 words that yield the lowest normalised log probability for the sentence (normalised using the word's log unigram probability).
- They take each of these values in turn as the score of the sentence.
- LCL denote these measures as *Word LP Min-N* (*Word LP Min-1* is the log probability given by the word with the lowest normalised log probability, *Word LP Min-2* the log probability given by the word with the second lowest normalised log probability, etc.).
- This class of scoring functions seeks to identify the lexical locus of syntactic anomaly.

Word Log Prob Scoring Functions

- In addition to the sentence log probability scoring functions, LCL also experiment with *individual word log probability*.
- For each test sentence they extract the 5 words that yield the lowest normalised log probability for the sentence (normalised using the word's log unigram probability).
- They take each of these values in turn as the score of the sentence.
- LCL denote these measures as *Word LP Min-N* (*Word LP Min-1* is the log probability given by the word with the lowest normalised log probability, *Word LP Min-2* the log probability given by the word with the second lowest normalised log probability, etc.).
- This class of scoring functions seeks to identify the lexical locus of syntactic anomaly.

Word Log Prob Scoring Functions

- In addition to the sentence log probability scoring functions, LCL also experiment with *individual word log probability*.
- For each test sentence they extract the 5 words that yield the lowest normalised log probability for the sentence (normalised using the word's log unigram probability).
- They take each of these values in turn as the score of the sentence.
- LCL denote these measures as *Word LP Min-N* (*Word LP Min-1* is the log probability given by the word with the lowest normalised log probability, *Word LP Min-2* the log probability given by the word with the second lowest normalised log probability, etc.).
- This class of scoring functions seeks to identify the lexical locus of syntactic anomaly.

Word Log Prob Scoring Functions

- In addition to the sentence log probability scoring functions, LCL also experiment with *individual word log probability*.
- For each test sentence they extract the 5 words that yield the lowest normalised log probability for the sentence (normalised using the word's log unigram probability).
- They take each of these values in turn as the score of the sentence.
- LCL denote these measures as *Word LP Min-N* (*Word LP Min-1* is the log probability given by the word with the lowest normalised log probability, *Word LP Min-2* the log probability given by the word with the second lowest normalised log probability, etc.).
- This class of scoring functions seeks to identify the lexical locus of syntactic anomaly.

Word Log Prob Scoring Functions

- In addition to the sentence log probability scoring functions, LCL also experiment with *individual word log probability*.
- For each test sentence they extract the 5 words that yield the lowest normalised log probability for the sentence (normalised using the word's log unigram probability).
- They take each of these values in turn as the score of the sentence.
- LCL denote these measures as *Word LP Min-N* (*Word LP Min-1* is the log probability given by the word with the lowest normalised log probability, *Word LP Min-2* the log probability given by the word with the second lowest normalised log probability, etc.).
- This class of scoring functions seeks to identify the lexical locus of syntactic anomaly.

Results: 3-Gram, BHMM, and 2T, BNC Trained and Tested

Measure	3-gram	BHMM	2T
LogProb	0.30	0.25	0.26
Mean LP	0.35	0.26	0.31
Norm LP (Div)	**0.42**	0.44	**0.50**
SLOR	0.41	**0.45**	**0.50**
Word LP Min-1	0.35	0.26	0.35
Word LP Min-2	0.41	0.38	0.43
Word LP Min-3	0.41	0.42	0.44
Word LP Min-4	0.40	0.43	0.43
Word LP Min-5	0.39	0.41	0.41

RNNLM (600 Neurons), BNC Trained and Tested

Measure	RNNLM
LogProb	0.32
Mean LP	0.39
Norm LP (Div)	**0.53**
SLOR	**0.53**
Word LP Min-1	0.38
Word LP Min-2	0.48
Word LP Min-3	0.50
Word LP Min-4	0.51
Word LP Min-5	0.50

Stanford PCFG BNC Tested

Measure	Stanford PCFG (Unlexicalised)
LogProb	0.21
Mean LP	0.18
Norm LP (Div)	**0.26**
Norm LP (Sub)	0.22
SLOR	0.25

3-Gram, BHMM, and 2T, BNC Trained and Tested on Filtered Adger Sentences

Measure	3-gram	BHMM	2T
LogProb	0.33	0.26	−0.21
Mean LP	0.27	0.11	0.18
Norm LP (Div)	0.36	0.30	0.17
SLOR	0.37	0.31	0.37
Word LP Min-1	**0.45**	0.10	0.32
Word LP Min-2	0.34	**0.34**	**0.40**
Word LP Min-3	0.34	0.41	**0.40**
Word LP Min-4	0.25	0.34	0.35
Word LP Min-5	0.33	0.26	0.30

RNNLM (600 Neurons), BNC Trained and Tested on Filtered Adger Sentences

Measure	RNNLM
LogProb	0.32
Mean LP	0.17
Norm LP (Div)	0.23
SLOR	0.23
Word LP Min-1	0.02
Word LP Min-2	0.27
Word LP Min-3	**0.38**
Word LP Min-4	0.28
Word LP Min-5	0.29

3-Gram, BHMM, and 2T, English Wikipedia Trained and Tested on Filtered Adger Sentences

Measure	3-gram	BHMM	2T
LogProb	0.34	0.33	0.35
Mean LP	0.28	0.23	0.26
Norm LP (Div)	0.36	**0.41**	0.41
SLOR	0.36	0.40	0.39
Word LP Min-1	**0.49**	0.25	0.46
Word LP Min-2	0.35	0.37	**0.49**
Word LP Min-3	0.34	**0.41**	0.35
Word LP Min-4	0.29	0.39	0.29
Word LP Min-5	0.32	**0.41**	0.29

RNNLM (600 Neurons), English Wikipedia Trained and Tested on Filtered Adger Sentences

Measure	RNNLM
LogProb	0.35
Mean LP	0.23
Norm LP (Div)	0.27
SLOR	0.25
Word LP Min-1	0.04
Word LP Min-2	0.30
Word LP Min-3	**0.38**
Word LP Min-4	0.34
Word LP Min-5	0.28

Experimenting with English Wikipedia Corpora

- LCL train and test their models on sentences from the English Wikipedia (ENWIKI).
- They follow the same protocol that they applied for their BNC experiment.
- They use AMT crowd sourcing (filtered for language fluency) on round trip Google translated sentences to obtain a test set of 2500 sentences.
- LCL train their models on 100m words of randomly selected English Wikipedia text.

Experimenting with English Wikipedia Corpora

- LCL train and test their models on sentences from the English Wikipedia (ENWIKI).
- They follow the same protocol that they applied for their BNC experiment.
- They use AMT crowd sourcing (filtered for language fluency) on round trip Google translated sentences to obtain a test set of 2500 sentences.
- LCL train their models on 100m words of randomly selected English Wikipedia text.

Experimenting with English Wikipedia Corpora

- LCL train and test their models on sentences from the English Wikipedia (ENWIKI).
- They follow the same protocol that they applied for their BNC experiment.
- They use AMT crowd sourcing (filtered for language fluency) on round trip Google translated sentences to obtain a test set of 2500 sentences.
- LCL train their models on 100m words of randomly selected English Wikipedia text.

Experimenting with English Wikipedia Corpora

- LCL train and test their models on sentences from the English Wikipedia (ENWIKI).
- They follow the same protocol that they applied for their BNC experiment.
- They use AMT crowd sourcing (filtered for language fluency) on round trip Google translated sentences to obtain a test set of 2500 sentences.
- LCL train their models on 100m words of randomly selected English Wikipedia text.

ENWIKI Experiment: 3-Gram, BHMM, and 2T

Measure	3-gram	BHMM	2T
LogProb	0.36	0.32	0.35
Mean LP	0.36	0.28	0.35
Norm LP (Div)	0.41	0.44	0.49
SLOR	0.41	0.46	**0.50**
Word LP Min-1	0.38	0.36	0.37
Word LP Min-2	0.43	0.46	0.49
Word LP Min-3	0.43	0.47	**0.50**
Word LP Min-4	**0.44**	0.47	**0.50**
Word LP Min-5	0.43	**0.48**	0.49

ENWIKI: RNNLM (600 Neurons)

Measure	RNNLM
LogProb	0.44
Mean LP	0.46
Norm LP (Div)	0.55
SLOR	0.57
Word LP Min-1	0.51
Word LP Min-2	0.60
Word LP Min-3	**0.62**
Word LP Min-4	0.60
Word LP Min-5	0.58

Wikipedia Corpora in Other Languages

- LCL train and test their models on Wikipedia texts in Spanish (ESWIKI), German (DEWIKI), and Russian (RUWIKI).
- They use the same protocol for round trip machine translation and crowd sourced AMT grammaticality judgements to annotate test sentences in these languages, that they employed for the BNC and English Wikipedia experiments.
- The RNNLM, with 600 neurons, combined with *SLOR*, gives the best performance for these three Wikipedia corpora.

Wikipedia Corpora in Other Languages

- LCL train and test their models on Wikipedia texts in Spanish (ESWIKI), German (DEWIKI), and Russian (RUWIKI).
- They use the same protocol for round trip machine translation and crowd sourced AMT grammaticality judgements to annotate test sentences in these languages, that they employed for the BNC and English Wikipedia experiments.
- The RNNLM, with 600 neurons, combined with *SLOR*, gives the best performance for these three Wikipedia corpora.

Wikipedia Corpora in Other Languages

- LCL train and test their models on Wikipedia texts in Spanish (ESWIKI), German (DEWIKI), and Russian (RUWIKI).
- They use the same protocol for round trip machine translation and crowd sourced AMT grammaticality judgements to annotate test sentences in these languages, that they employed for the BNC and English Wikipedia experiments.
- The RNNLM, with 600 neurons, combined with *SLOR*, gives the best performance for these three Wikipedia corpora.

RNNLM, with SLOR, for ESWIKI, DEWIKI, and RUWIKI

Corpora	RNNLM (600 Neurons)
ESWIKI	0.60
DEWIKI	0.69
RUWIKI	0.61

Estimating Individual Human Performance

- It is not reasonable to expect ML models to achieve a perfect correlation with mean judgements, given that individual human annotators could not do this.
- LCL estimate an arbitrary individual human annotator's performance relative to the set of mean judgements for a test set.
- They randomly select a single rating for each sentence, and they compute the Pearson correlation between these individual judgements and the mean ratings for the rest of the annotators (one vs the rest).
- LCL ran the experiment 50 times to reduce sample variation.
- The simulated individual human predictor specifies an upper bound on any model's expected performance.

Estimating Individual Human Performance

- It is not reasonable to expect ML models to achieve a perfect correlation with mean judgements, given that individual human annotators could not do this.
- LCL estimate an arbitrary individual human annotator's performance relative to the set of mean judgements for a test set.
- They randomly select a single rating for each sentence, and they compute the Pearson correlation between these individual judgements and the mean ratings for the rest of the annotators (one vs the rest).
- LCL ran the experiment 50 times to reduce sample variation.
- The simulated individual human predictor specifies an upper bound on any model's expected performance.

Estimating Individual Human Performance

- It is not reasonable to expect ML models to achieve a perfect correlation with mean judgements, given that individual human annotators could not do this.
- LCL estimate an arbitrary individual human annotator's performance relative to the set of mean judgements for a test set.
- They randomly select a single rating for each sentence, and they compute the Pearson correlation between these individual judgements and the mean ratings for the rest of the annotators (one vs the rest).
- LCL ran the experiment 50 times to reduce sample variation.
- The simulated individual human predictor specifies an upper bound on any model's expected performance.

Estimating Individual Human Performance

- It is not reasonable to expect ML models to achieve a perfect correlation with mean judgements, given that individual human annotators could not do this.
- LCL estimate an arbitrary individual human annotator's performance relative to the set of mean judgements for a test set.
- They randomly select a single rating for each sentence, and they compute the Pearson correlation between these individual judgements and the mean ratings for the rest of the annotators (one vs the rest).
- LCL ran the experiment 50 times to reduce sample variation.
- The simulated individual human predictor specifies an upper bound on any model's expected performance.

Estimating Individual Human Performance

- It is not reasonable to expect ML models to achieve a perfect correlation with mean judgements, given that individual human annotators could not do this.
- LCL estimate an arbitrary individual human annotator's performance relative to the set of mean judgements for a test set.
- They randomly select a single rating for each sentence, and they compute the Pearson correlation between these individual judgements and the mean ratings for the rest of the annotators (one vs the rest).
- LCL ran the experiment 50 times to reduce sample variation.
- The simulated individual human predictor specifies an upper bound on any model's expected performance.

One vs. Many Simulated Individual Annotator

Test Domain	Corr to Mean
Adger Filtered	0.726
BNC	0.667
ENWIKI	0.741
ESWIKI	0.701
DEWIKI	0.773
RUWIKI	0.655

Evaluating LCL's Models Against Estimated Human Performance

- If we use estimated human performance to evaluate our models, then LCL's best enriched models do quite well.
- 3-gram + *Word LP Min* – 1 and 2T BHMM + *Word LP Min* – 2, trained on ENWIKI, both scored 0.49 for the Adger filtered test set, against an estimated individual human annotator correlation of 0.726
- RNNLM + *Norm* (*Div*) and RNNLM + *SLOR*, trained and tested on the BNC, scored 0.53, against an estimated human correlation of 0.667
- RNNLM + *Word LP Min* – 3, trained and tested on an ENWIKI corpus, achieved 0.62, against estimated human performance of 0.741
- RNNLM + *SLOR* on DEWIKI is 0.69 against an estimated human performance of 0.773.
- RNNLM + *SLOR* on ESWIKI is 0.60 against an estimated human performance of 0.701.
- RNNLM + *SLOR* on RUWIKI is 0.61 against an estimated human performance of 0.655.

Evaluating LCL's Models Against Estimated Human Performance

- If we use estimated human performance to evaluate our models, then LCL's best enriched models do quite well.
- 3-gram + *Word LP Min* − 1 and 2T BHMM + *Word LP Min* − 2, trained on ENWIKI, both scored 0.49 for the Adger filtered test set, against an estimated individual human annotator correlation of 0.726
- RNNLM + *Norm* (*Div*) and RNNLM + *SLOR*, trained and tested on the BNC, scored 0.53, against an estimated human correlation of 0.667
- RNNLM + *Word LP Min* − 3, trained and tested on an ENWIKI corpus, achieved 0.62, against estimated human performance of 0.741
- RNNLM + *SLOR* on DEWIKI is 0.69 against an estimated human performance of 0.773.
- RNNLM + *SLOR* on ESWIKI is 0.60 against an estimated human performance of 0.701.
- RNNLM + *SLOR* on RUWIKI is 0.61 against an estimated human performance of 0.655.

Evaluating LCL's Models Against Estimated Human Performance

- If we use estimated human performance to evaluate our models, then LCL's best enriched models do quite well.
- 3-gram + *Word LP Min* − 1 and 2T BHMM + *Word LP Min* − 2, trained on ENWIKI, both scored 0.49 for the Adger filtered test set, against an estimated individual human annotator correlation of 0.726
- RNNLM + *Norm* (*Div*) and RNNLM + *SLOR*, trained and tested on the BNC, scored 0.53, against an estimated human correlation of 0.667
- RNNLM + *Word LP Min* − 3, trained and tested on an ENWIKI corpus, achieved 0.62, against estimated human performance of 0.741
- RNNLM + *SLOR* on DEWIKI is 0.69 against an estimated human performance of 0.773.
- RNNLM + *SLOR* on ESWIKI is 0.60 against an estimated human performance of 0.701.
- RNNLM + *SLOR* on RUWIKI is 0.61 against an estimated human performance of 0.655.

Evaluating LCL's Models Against Estimated Human Performance

- If we use estimated human performance to evaluate our models, then LCL's best enriched models do quite well.
- 3-gram + *Word LP Min* − 1 and 2T BHMM + *Word LP Min* − 2, trained on ENWIKI, both scored 0.49 for the Adger filtered test set, against an estimated individual human annotator correlation of 0.726
- RNNLM + *Norm* (*Div*) and RNNLM + *SLOR*, trained and tested on the BNC, scored 0.53, against an estimated human correlation of 0.667
- RNNLM + *Word LP Min* − 3, trained and tested on an ENWIKI corpus, achieved 0.62, against estimated human performance of 0.741
- RNNLM + *SLOR* on DEWIKI is 0.69 against an estimated human performance of 0.773.
- RNNLM + *SLOR* on ESWIKI is 0.60 against an estimated human performance of 0.701.
- RNNLM + *SLOR* on RUWIKI is 0.61 against an estimated human performance of 0.655.

Evaluating LCL's Models Against Estimated Human Performance

- If we use estimated human performance to evaluate our models, then LCL's best enriched models do quite well.
- 3-gram + *Word LP Min* − 1 and 2T BHMM + *Word LP Min* − 2, trained on ENWIKI, both scored 0.49 for the Adger filtered test set, against an estimated individual human annotator correlation of 0.726
- RNNLM + *Norm* (*Div*) and RNNLM + *SLOR*, trained and tested on the BNC, scored 0.53, against an estimated human correlation of 0.667
- RNNLM + *Word LP Min* − 3, trained and tested on an ENWIKI corpus, achieved 0.62, against estimated human performance of 0.741
- RNNLM + *SLOR* on DEWIKI is 0.69 against an estimated human performance of 0.773.
- RNNLM + *SLOR* on ESWIKI is 0.60 against an estimated human performance of 0.701.
- RNNLM + *SLOR* on RUWIKI is 0.61 against an estimated human performance of 0.655.

Evaluating LCL's Models Against Estimated Human Performance

- If we use estimated human performance to evaluate our models, then LCL's best enriched models do quite well.
- 3-gram + *Word LP Min* − 1 and 2T BHMM + *Word LP Min* − 2, trained on ENWIKI, both scored 0.49 for the Adger filtered test set, against an estimated individual human annotator correlation of 0.726
- RNNLM + *Norm* (*Div*) and RNNLM + *SLOR*, trained and tested on the BNC, scored 0.53, against an estimated human correlation of 0.667
- RNNLM + *Word LP Min* − 3, trained and tested on an ENWIKI corpus, achieved 0.62, against estimated human performance of 0.741
- RNNLM + *SLOR* on DEWIKI is 0.69 against an estimated human performance of 0.773.
- RNNLM + *SLOR* on ESWIKI is 0.60 against an estimated human performance of 0.701.
- RNNLM + *SLOR* on RUWIKI is 0.61 against an estimated human performance of 0.655.

Evaluating LCL's Models Against Estimated Human Performance

- If we use estimated human performance to evaluate our models, then LCL's best enriched models do quite well.
- 3-gram + *Word LP Min* − 1 and 2T BHMM + *Word LP Min* − 2, trained on ENWIKI, both scored 0.49 for the Adger filtered test set, against an estimated individual human annotator correlation of 0.726
- RNNLM + *Norm* (*Div*) and RNNLM + *SLOR*, trained and tested on the BNC, scored 0.53, against an estimated human correlation of 0.667
- RNNLM + *Word LP Min* − 3, trained and tested on an ENWIKI corpus, achieved 0.62, against estimated human performance of 0.741
- RNNLM + *SLOR* on DEWIKI is 0.69 against an estimated human performance of 0.773.
- RNNLM + *SLOR* on ESWIKI is 0.60 against an estimated human performance of 0.701.
- RNNLM + *SLOR* on RUWIKI is 0.61 against an estimated human performance of 0.655.

Enriching the Training Data with Syntactic and Semantic Annotation

- Ek, Bernardy, and Lappin (2019) (EBL) test the effect of enriching the training data with syntactic and semantic annotations, on the performance of an LSTM LM in the sentence acceptability task.
- For a simple LSTM LM trained on raw text, $P_M(w_i) = P(w_i|(w_{i-1}), ..., (w_{i-n}))$.
- An LSTM LM trained on text annotated with semantic or syntactic tags predicts the next word w_i in a sentence on the basis of the previous sequence of words and their tags.
- $P_M(w_i) = P(w_i|(w_{i-1}, t_{i-1}), ..., (w_{i-n}, t_{i-n}))$
- The current tag (t_i) is not given when the model predicts the current word (w_i).

Enriching the Training Data with Syntactic and Semantic Annotation

- Ek, Bernardy, and Lappin (2019) (EBL) test the effect of enriching the training data with syntactic and semantic annotations, on the performance of an LSTM LM in the sentence acceptability task.
- For a simple LSTM LM trained on raw text, $P_M(w_i) = P(w_i|(w_{i-1}), ..., (w_{i-n}))$.
- An LSTM LM trained on text annotated with semantic or syntactic tags predicts the next word w_i in a sentence on the basis of the previous sequence of words and their tags.
- $P_M(w_i) = P(w_i|(w_{i-1}, t_{i-1}), ..., (w_{i-n}, t_{i-n}))$
- The current tag (t_i) is not given when the model predicts the current word (w_i).

Enriching the Training Data with Syntactic and Semantic Annotation

- Ek, Bernardy, and Lappin (2019) (EBL) test the effect of enriching the training data with syntactic and semantic annotations, on the performance of an LSTM LM in the sentence acceptability task.
- For a simple LSTM LM trained on raw text, $P_M(w_i) = P(w_i|(w_{i-1}), ..., (w_{i-n}))$.
- An LSTM LM trained on text annotated with semantic or syntactic tags predicts the next word w_i in a sentence on the basis of the previous sequence of words and their tags.
- $P_M(w_i) = P(w_i|(w_{i-1}, t_{i-1}), ..., (w_{i-n}, t_{i-n}))$
- The current tag (t_i) is not given when the model predicts the current word (w_i).

Enriching the Training Data with Syntactic and Semantic Annotation

- Ek, Bernardy, and Lappin (2019) (EBL) test the effect of enriching the training data with syntactic and semantic annotations, on the performance of an LSTM LM in the sentence acceptability task.
- For a simple LSTM LM trained on raw text, $P_M(w_i) = P(w_i|(w_{i-1}), ..., (w_{i-n}))$.
- An LSTM LM trained on text annotated with semantic or syntactic tags predicts the next word w_i in a sentence on the basis of the previous sequence of words and their tags.
- $P_M(w_i) = P(w_i|(w_{i-1}, t_{i-1}), ..., (w_{i-n}, t_{i-n}))$
- The current tag (t_i) is not given when the model predicts the current word (w_i).

Enriching the Training Data with Syntactic and Semantic Annotation

- Ek, Bernardy, and Lappin (2019) (EBL) test the effect of enriching the training data with syntactic and semantic annotations, on the performance of an LSTM LM in the sentence acceptability task.
- For a simple LSTM LM trained on raw text, $P_M(w_i) = P(w_i|(w_{i-1}), ..., (w_{i-n}))$.
- An LSTM LM trained on text annotated with semantic or syntactic tags predicts the next word w_i in a sentence on the basis of the previous sequence of words and their tags.
- $P_M(w_i) = P(w_i|(w_{i-1}, t_{i-1}), ..., (w_{i-n}, t_{i-n}))$
- The current tag (t_i) is not given when the model predicts the current word (w_i).

The Language Models

EBL implement four variants of LSTM language models, each of which predicts the next word in a sequence, conditioned on

1. only the unannotated previous sequence of words,
2. the previous sequence of words and their semantic role tags,
3. the previous sequence of words and their syntactic dependency tags, and
4. the previous sequence of words and their dependency tree depth indicators.

Hyperparameters

- EBL's LSTMs are unidirectional, with one level of 600 units.
- The models are trained on a vocabulary of 100,000 words.
- They use word embeddings of 300 dimensions, and 30 dimensions for tags.
- EBL apply a drop out of 0.4 after the LSTM layer.
- Training runs for 10 epochs.

Hyperparameters

- EBL's LSTMs are unidirectional, with one level of 600 units.
- The models are trained on a vocabulary of 100,000 words.
- They use word embeddings of 300 dimensions, and 30 dimensions for tags.
- EBL apply a drop out of 0.4 after the LSTM layer.
- Training runs for 10 epochs.

Hyperparameters

- EBL's LSTMs are unidirectional, with one level of 600 units.
- The models are trained on a vocabulary of 100,000 words.
- They use word embeddings of 300 dimensions, and 30 dimensions for tags.
- EBL apply a drop out of 0.4 after the LSTM layer.
- Training runs for 10 epochs.

Hyperparameters

- EBL's LSTMs are unidirectional, with one level of 600 units.
- The models are trained on a vocabulary of 100,000 words.
- They use word embeddings of 300 dimensions, and 30 dimensions for tags.
- EBL apply a drop out of 0.4 after the LSTM layer.
- Training runs for 10 epochs.

Hyperparameters

- EBL's LSTMs are unidirectional, with one level of 600 units.
- The models are trained on a vocabulary of 100,000 words.
- They use word embeddings of 300 dimensions, and 30 dimensions for tags.
- EBL apply a drop out of 0.4 after the LSTM layer.
- Training runs for 10 epochs.

Training Corpus

- EBL train their LMs on a randomly selected subset of the CoNLL 2017 Wikipedia dataset (Nivre et al., 2017).
- The corpus is annotated with dependency parse trees.
- They remove sentences whose dependency root is not a verb to eliminate non-sentences.
- They also delete sentences longer than 30 words.
- The remaining corpus contains 87M tokens and 5.3M sentences.

Training Corpus

- EBL train their LMs on a randomly selected subset of the CoNLL 2017 Wikipedia dataset (Nivre et al., 2017).
- The corpus is annotated with dependency parse trees.
- They remove sentences whose dependency root is not a verb to eliminate non-sentences.
- They also delete sentences longer than 30 words.
- The remaining corpus contains 87M tokens and 5.3M sentences.

Training Corpus

- EBL train their LMs on a randomly selected subset of the CoNLL 2017 Wikipedia dataset (Nivre et al., 2017).
- The corpus is annotated with dependency parse trees.
- They remove sentences whose dependency root is not a verb to eliminate non-sentences.
- They also delete sentences longer than 30 words.
- The remaining corpus contains 87M tokens and 5.3M sentences.

Training Corpus

- EBL train their LMs on a randomly selected subset of the CoNLL 2017 Wikipedia dataset (Nivre et al., 2017).
- The corpus is annotated with dependency parse trees.
- They remove sentences whose dependency root is not a verb to eliminate non-sentences.
- They also delete sentences longer than 30 words.
- The remaining corpus contains 87M tokens and 5.3M sentences.

Training Corpus

- EBL train their LMs on a randomly selected subset of the CoNLL 2017 Wikipedia dataset (Nivre et al., 2017).
- The corpus is annotated with dependency parse trees.
- They remove sentences whose dependency root is not a verb to eliminate non-sentences.
- They also delete sentences longer than 30 words.
- The remaining corpus contains 87M tokens and 5.3M sentences.

Test set

- EBL use LCL's AMT annotated test set of 2500 BNC sentences as their test suite.
- This set contains 500 original English sentences and 2000 sentences translated through Norwegian, Spanish, Chinese or Japanese back to English.
- Annotation is on a scale of 1 to 4.
- On average, each sentence is rated by 14 annotators.

Test set

- EBL use LCL's AMT annotated test set of 2500 BNC sentences as their test suite.
- This set contains 500 original English sentences and 2000 sentences translated through Norwegian, Spanish, Chinese or Japanese back to English.
- Annotation is on a scale of 1 to 4.
- On average, each sentence is rated by 14 annotators.

Test set

- EBL use LCL's AMT annotated test set of 2500 BNC sentences as their test suite.
- This set contains 500 original English sentences and 2000 sentences translated through Norwegian, Spanish, Chinese or Japanese back to English.
- Annotation is on a scale of 1 to 4.
- On average, each sentence is rated by 14 annotators.

Test set

- EBL use LCL's AMT annotated test set of 2500 BNC sentences as their test suite.
- This set contains 500 original English sentences and 2000 sentences translated through Norwegian, Spanish, Chinese or Japanese back to English.
- Annotation is on a scale of 1 to 4.
- On average, each sentence is rated by 14 annotators.

Semantic Tags

- EBL automatically tag the training corpus and the test set with semantic role markers obtained from Abzianidze et al. (2017)'s Parallel Meaning Bank.
- These roles provide a fine-grained set of semantic types for expressions of major lexical categories.
- The semantic roles of *he* and *his* are distinguished in

He	*took*	*his*	*book*	.
PRO	EPS	HAS	CON	NIL

Semantic Tags

- EBL automatically tag the training corpus and the test set with semantic role markers obtained from Abzianidze et al. (2017)'s Parallel Meaning Bank.
- These roles provide a fine-grained set of semantic types for expressions of major lexical categories.
- The semantic roles of *he* and *his* are distinguished in

He	*took*	*his*	*book*	.
PRO	EPS	HAS	CON	NIL

Semantic Tags

- EBL automatically tag the training corpus and the test set with semantic role markers obtained from Abzianidze et al. (2017)'s Parallel Meaning Bank.
- These roles provide a fine-grained set of semantic types for expressions of major lexical categories.
- The semantic roles of *he* and *his* are distinguished in

He	*took*	*his*	*book*	.
PRO	EPS	HAS	CON	NIL

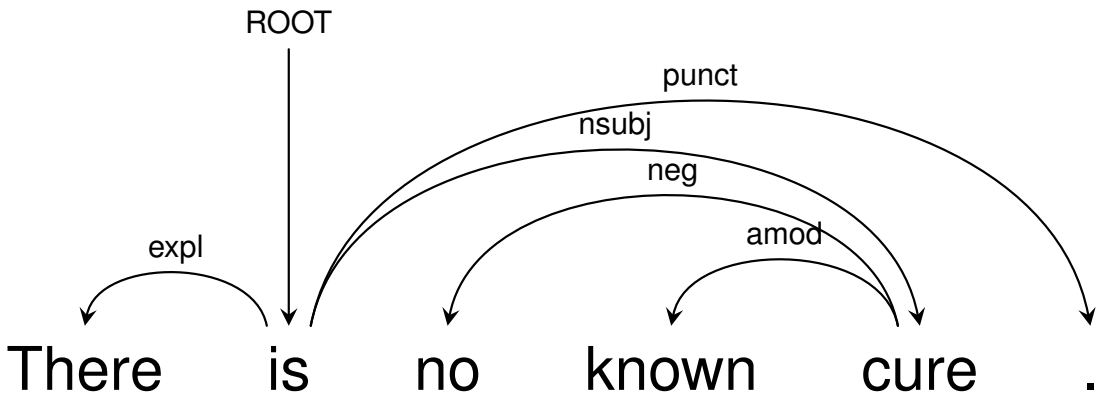
Syntactic Tags

Nivre's (2016) Universal Dependency grammar parses sentence's with trees that specify dependency relations among its constituents.

EBL use Chen and Manning's (2014) Stanford Dependency Parser to generate syntactic tags for the training and test sets.

Syntactic Tags

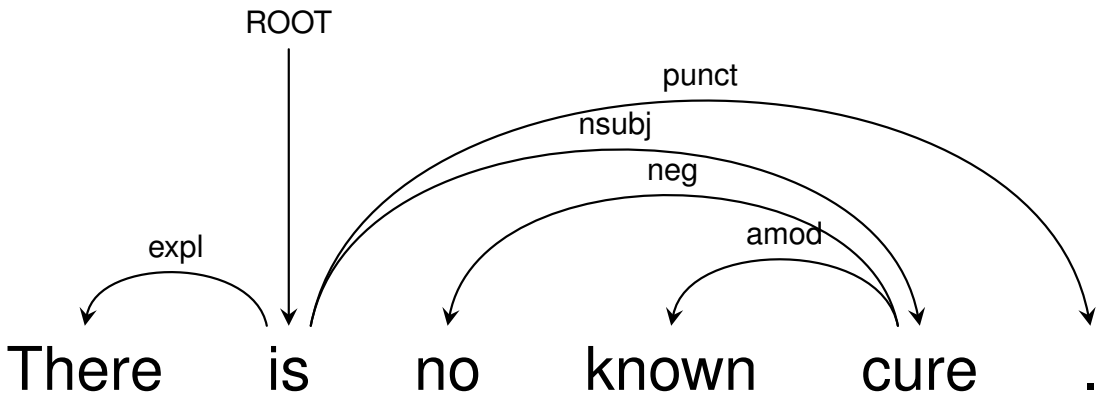
Nivre's (2016) Universal Dependency grammar parses sentence's with trees that specify dependency relations among its constituents.



EBL use Chen and Manning's (2014) Stanford Dependency Parser to generate syntactic tags for the training and test sets.

Syntactic Tags

Nivre's (2016) Universal Dependency grammar parses sentence's with trees that specify dependency relations among its constituents.



EBL use Chen and Manning's (2014) Stanford Dependency Parser to generate syntactic tags for the training and test sets.

Dependency Tree Depth Indicators

EBL follow Gòmez-Rodrìguez, and Vilares (2018) in encoding the tree depth of a word n by computing the number of common ancestors in the tree between word n and word $n + 1$.

Dependency tags combined with depth indicators provide a linearised encoding of a dependency tree.

Dependency Tree Depth Indicators

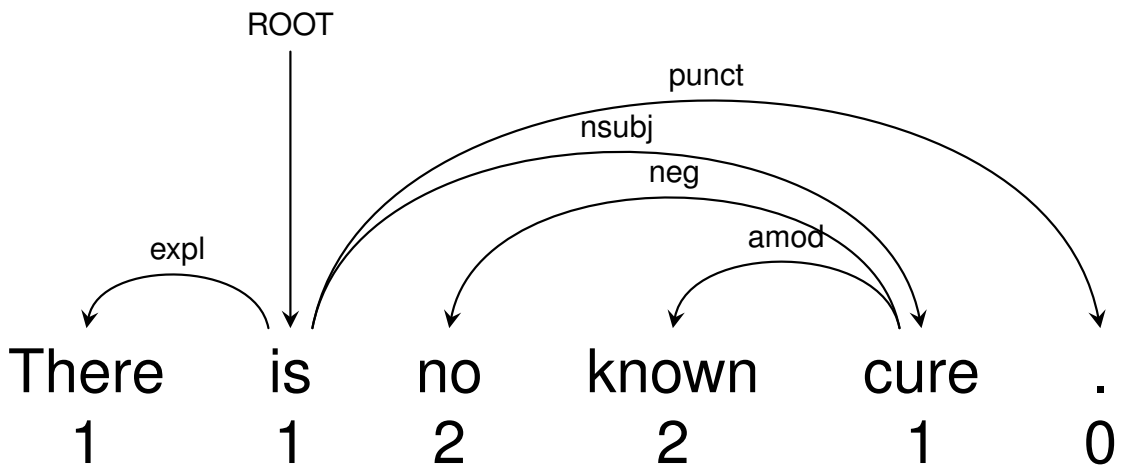
EBL follow Gòmez-Rodrìguez, and Vilares (2018) in encoding the tree depth of a word n by computing the number of common ancestors in the tree between word n and word $n + 1$.

Dependency tags combined with depth indicators provide a linearised encoding of a dependency tree.

Dependency Tree Depth Indicators

EBL follow Gòmez-Rodrìguez, and Vilares (2018) in encoding the tree depth of a word n by computing the number of common ancestors in the tree between word n and word $n + 1$.

Dependency tags combined with depth indicators provide a linearised encoding of a dependency tree.



Evaluating Model Accuracy and Perplexity

- EBL use SLOR as the acceptability scoring function for their models.
- Instead of the simple Pearson correlation coefficient, they apply a weighted Pearson correlation to assess their model's performance in predicting acceptability, relative to human ratings.
- The weighted Pearson metric takes account of the fact that the model's score is compared to the mean rating value of a sentence, derived from several annotators, by giving greater weight to ratings with lower standard deviation.
- EBL measure the perplexity of a model by its cross entropy training loss, which is the logarithm of its perplexity.

Evaluating Model Accuracy and Perplexity

- EBL use SLOR as the acceptability scoring function for their models.
- Instead of the simple Pearson correlation coefficient, they apply a weighted Pearson correlation to assess their model's performance in predicting acceptability, relative to human ratings.
- The weighted Pearson metric takes account of the fact that the model's score is compared to the mean rating value of a sentence, derived from several annotators, by giving greater weight to ratings with lower standard deviation.
- EBL measure the perplexity of a model by its cross entropy training loss, which is the logarithm of its perplexity.

Evaluating Model Accuracy and Perplexity

- EBL use SLOR as the acceptability scoring function for their models.
- Instead of the simple Pearson correlation coefficient, they apply a weighted Pearson correlation to assess their model's performance in predicting acceptability, relative to human ratings.
- The weighted Pearson metric takes account of the fact that the model's score is compared to the mean rating value of a sentence, derived from several annotators, by giving greater weight to ratings with lower standard deviation.
- EBL measure the perplexity of a model by its cross entropy training loss, which is the logarithm of its perplexity.

Evaluating Model Accuracy and Perplexity

- EBL use SLOR as the acceptability scoring function for their models.
- Instead of the simple Pearson correlation coefficient, they apply a weighted Pearson correlation to assess their model's performance in predicting acceptability, relative to human ratings.
- The weighted Pearson metric takes account of the fact that the model's score is compared to the mean rating value of a sentence, derived from several annotators, by giving greater weight to ratings with lower standard deviation.
- EBL measure the perplexity of a model by its cross entropy training loss, which is the logarithm of its perplexity.

Model Performance Results

*M** corresponds to the the model *M* with the tags randomly shuffled in the test set.

	HUMAN	LSTM	+SYN	+SYN*	+SEM	+SEM*	+DEPTH
HUMAN	1.00						
LSTM	**0.58**	1.00					
+SYN	0.55	0.96	1.00				
+SYN*	0.39	0.76	0.75	1.00			
+SEM	0.54	0.81	0.78	0.61	1.00		
+SEM*	0.52	0.81	0.78	0.63	0.96	1.00	
+DEPTH	0.56	0.97	0.97	0.74	0.79	0.79	1.00
+DEPTH*	0.46	0.87	0.85	0.73	0.72	0.72	0.86
+SYN+DEPTH	0.54						

The Effect of Annotation on Predictive Performance

- The simple LSTM LM outperforms all models trained on text with syntactic and semantic tags, achieving 0.58 correlation to mean human judgments (comparable to the LCL results).
- The depth indicator model does best of all annotated models (0.56), the syntactic tag model is just below it (0.55), while the semantic role model scores lowest (0.54).
- Shuffling the tags causes a drop of 0.16 in correlation for syntactic tags, 0.1 for tree depth, but only 0.02, for semantic tags, indicating that syntactic tags and depth markers contribute more information to their respective models' predictions.
- The combined syntactic tag + tree depth marker model (0.54) performs below each of its component models.

The Effect of Annotation on Predictive Performance

- The simple LSTM LM outperforms all models trained on text with syntactic and semantic tags, achieving 0.58 correlation to mean human judgments (comparable to the LCL results).
- The depth indicator model does best of all annotated models (0.56), the syntactic tag model is just below it (0.55), while the semantic role model scores lowest (0.54).
- Shuffling the tags causes a drop of 0.16 in correlation for syntactic tags, 0.1 for tree depth, but only 0.02, for semantic tags, indicating that syntactic tags and depth markers contribute more information to their respective models' predictions.
- The combined syntactic tag + tree depth marker model (0.54) performs below each of its component models.

The Effect of Annotation on Predictive Performance

- The simple LSTM LM outperforms all models trained on text with syntactic and semantic tags, achieving 0.58 correlation to mean human judgments (comparable to the LCL results).
- The depth indicator model does best of all annotated models (0.56), the syntactic tag model is just below it (0.55), while the semantic role model scores lowest (0.54).
- Shuffling the tags causes a drop of 0.16 in correlation for syntactic tags, 0.1 for tree depth, but only 0.02, for semantic tags, indicating that syntactic tags and depth markers contribute more information to their respective models' predictions.
- The combined syntactic tag + tree depth marker model (0.54) performs below each of its component models.

The Effect of Annotation on Predictive Performance

- The simple LSTM LM outperforms all models trained on text with syntactic and semantic tags, achieving 0.58 correlation to mean human judgments (comparable to the LCL results).
- The depth indicator model does best of all annotated models (0.56), the syntactic tag model is just below it (0.55), while the semantic role model scores lowest (0.54).
- Shuffling the tags causes a drop of 0.16 in correlation for syntactic tags, 0.1 for tree depth, but only 0.02, for semantic tags, indicating that syntactic tags and depth markers contribute more information to their respective models' predictions.
- The combined syntactic tag + tree depth marker model (0.54) performs below each of its component models.

Annotation and Model Perplexity

Model	Loss	Accuracy
LSTM	5.04	0.24
+Syn	**4.79**	0.26
+Sem	5.23	0.21
+Depth	4.88	0.27

- There does not appear to be a direct correlation between an LSTM's quality as a language model, as indicated by its perplexity, and its performance on the sentence acceptability task.
- Syntactic tags and depth indicators decrease perplexity, but semantic markers increase it.
- The simple non-annotated LSTM outperforms all of them.
- It might be that SLOR masks the underlying perplexity of these models.

Annotation and Model Perplexity

Model	Loss	Accuracy
LSTM	5.04	0.24
+Syn	**4.79**	0.26
+Sem	5.23	0.21
+Depth	4.88	0.27

- There does not appear to be a direct correlation between an LSTM's quality as a language model, as indicated by its perplexity, and its performance on the sentence acceptability task.
- Syntactic tags and depth indicators decrease perplexity, but semantic markers increase it.
- The simple non-annotated LSTM outperforms all of them.
- It might be that SLOR masks the underlying perplexity of these models.

Annotation and Model Perplexity

Model	Loss	Accuracy
LSTM	5.04	0.24
+Syn	**4.79**	0.26
+Sem	5.23	0.21
+Depth	4.88	0.27

- There does not appear to be a direct correlation between an LSTM's quality as a language model, as indicated by its perplexity, and its performance on the sentence acceptability task.
- Syntactic tags and depth indicators decrease perplexity, but semantic markers increase it.
- The simple non-annotated LSTM outperforms all of them.
- It might be that SLOR masks the underlying perplexity of these models.

Annotation and Model Perplexity

Model	Loss	Accuracy
LSTM	5.04	0.24
+Syn	**4.79**	0.26
+Sem	5.23	0.21
+Depth	4.88	0.27

- There does not appear to be a direct correlation between an LSTM's quality as a language model, as indicated by its perplexity, and its performance on the sentence acceptability task.
- Syntactic tags and depth indicators decrease perplexity, but semantic markers increase it.
- The simple non-annotated LSTM outperforms all of them.
- It might be that SLOR masks the underlying perplexity of these models.

Transformer Models Applied to a Related Task

- Warstadt et al. (2019) discuss several pre-trained transformer models appied to classifying sentences in their Corpus of Linguistic Acceptability (CoLA) as acceptable or not.
- These models exhibit levels of accuracy that vary widely relative to the types of syntactic and morphological patterns that appear in CoLA.
- CoLA is a very different sort of test set from the one that LCL and EBL use in their experiments.
- It is drawn from linguists' examples intended to illustrate particular sorts of syntactic construction, and annotated with linguists' binary judgments.
- By contrast, the BNC test set consists of naturally occurring text, where a wide range of infelicities are introduced into many of the sentences through round trip machine translation, and it is annotated through AMT crowd sourcing with gradient acceptability judgments.

Transformer Models Applied to a Related Task

- Warstadt et al. (2019) discuss several pre-trained transformer models appied to classifying sentences in their Corpus of Linguistic Acceptability (CoLA) as acceptable or not.
- These models exhibit levels of accuracy that vary widely relative to the types of syntactic and morphological patterns that appear in CoLA.
- CoLA is a very different sort of test set from the one that LCL and EBL use in their experiments.
- It is drawn from linguists' examples intended to illustrate particular sorts of syntactic construction, and annotated with linguists' binary judgments.
- By contrast, the BNC test set consists of naturally occurring text, where a wide range of infelicities are introduced into many of the sentences through round trip machine translation, and it is annotated through AMT crowd sourcing with gradient acceptability judgments.

Transformer Models Applied to a Related Task

- Warstadt et al. (2019) discuss several pre-trained transformer models appied to classifying sentences in their Corpus of Linguistic Acceptability (CoLA) as acceptable or not.
- These models exhibit levels of accuracy that vary widely relative to the types of syntactic and morphological patterns that appear in CoLA.
- CoLA is a very different sort of test set from the one that LCL and EBL use in their experiments.
- It is drawn from linguists' examples intended to illustrate particular sorts of syntactic construction, and annotated with linguists' binary judgments.
- By contrast, the BNC test set consists of naturally occurring text, where a wide range of infelicities are introduced into many of the sentences through round trip machine translation, and it is annotated through AMT crowd sourcing with gradient acceptability judgments.

Transformer Models Applied to a Related Task

- Warstadt et al. (2019) discuss several pre-trained transformer models appied to classifying sentences in their Corpus of Linguistic Acceptability (CoLA) as acceptable or not.
- These models exhibit levels of accuracy that vary widely relative to the types of syntactic and morphological patterns that appear in CoLA.
- CoLA is a very different sort of test set from the one that LCL and EBL use in their experiments.
- It is drawn from linguists' examples intended to illustrate particular sorts of syntactic construction, and annotated with linguists' binary judgments.
- By contrast, the BNC test set consists of naturally occurring text, where a wide range of infelicities are introduced into many of the sentences through round trip machine translation, and it is annotated through AMT crowd sourcing with gradient acceptability judgments.

Transformer Models Applied to a Related Task

- Warstadt et al. (2019) discuss several pre-trained transformer models appied to classifying sentences in their Corpus of Linguistic Acceptability (CoLA) as acceptable or not.
- These models exhibit levels of accuracy that vary widely relative to the types of syntactic and morphological patterns that appear in CoLA.
- CoLA is a very different sort of test set from the one that LCL and EBL use in their experiments.
- It is drawn from linguists' examples intended to illustrate particular sorts of syntactic construction, and annotated with linguists' binary judgments.
- By contrast, the BNC test set consists of naturally occurring text, where a wide range of infelicities are introduced into many of the sentences through round trip machine translation, and it is annotated through AMT crowd sourcing with gradient acceptability judgments.

Conclusions

- A simple LSTM performs surprisingly well on the sentence acceptability task.
- It is robust, given that, in the EBL experiments, it is trained on one domain (a Wikipedia corpus), and tested on another (BNC text).
- Enhancing the training input of an LSTM LM with syntactic or semantic markers, or full tree structures, does not improve its predictive power for the sentence acceptability task.
- These results provide additional support for BL's finding that using abstract syntactic markers to highlight structural relations may degrade an LSTM's performance on certain tasks.

Conclusions

- A simple LSTM performs surprisingly well on the sentence acceptability task.
- It is robust, given that, in the EBL experiments, it is trained on one domain (a Wikipedia corpus), and tested on another (BNC text).
- Enhancing the training input of an LSTM LM with syntactic or semantic markers, or full tree structures, does not improve its predictive power for the sentence acceptability task.
- These results provide additional support for BL's finding that using abstract syntactic markers to highlight structural relations may degrade an LSTM's performance on certain tasks.

Conclusions

- A simple LSTM performs surprisingly well on the sentence acceptability task.
- It is robust, given that, in the EBL experiments, it is trained on one domain (a Wikipedia corpus), and tested on another (BNC text).
- Enhancing the training input of an LSTM LM with syntactic or semantic markers, or full tree structures, does not improve its predictive power for the sentence acceptability task.
- These results provide additional support for BL's finding that using abstract syntactic markers to highlight structural relations may degrade an LSTM's performance on certain tasks.

Conclusions

- A simple LSTM performs surprisingly well on the sentence acceptability task.
- It is robust, given that, in the EBL experiments, it is trained on one domain (a Wikipedia corpus), and tested on another (BNC text).
- Enhancing the training input of an LSTM LM with syntactic or semantic markers, or full tree structures, does not improve its predictive power for the sentence acceptability task.
- These results provide additional support for BL's finding that using abstract syntactic markers to highlight structural relations may degrade an LSTM's performance on certain tasks.