Language Modeling with Syntactic and Semantic Representation for Sentence Acceptability Predictions

Adam Ek **Jean-Phillipe Bernardy** **Shalom Lappin**
Centre for Linguistic Theory and Studies in Probability
Department of Philosophy, Linguistics and Theory of Science
University of Gothenburg
{adam.ek, jean-philippe.bernardy, shalom.lappin}@gu.se

Abstract

In this paper, we investigate the effect of enhancing lexical embeddings in LSTM language models (LM) with syntactic and semantic representations. We evaluate the language models using perplexity, and we evaluate the performance of the models on the task of predicting human sentence acceptability judgments. We train LSTM language models on sentences automatically annotated with universal syntactic dependency roles (Nivre et al., 2016), dependency tree depth features, and universal semantic tags (Abzianidze et al., 2017) to predict sentence acceptability judgments. Our experiments indicate that syntactic depth and tags lower the perplexity compared to a plain LSTM language model, while semantic tags increase the perplexity. Our experiments also show that neither syntactic nor semantic tags improve the performance of LSTM language models on the task of predicting sentence acceptability judgments.

1 Introduction

Lau et al. (2014) show that human acceptability judgments are graded rather than binary. It is not entirely obvious what determines sentence acceptability for speakers and listeners. However, syntactic structure and semantic content are clearly central to acceptability judgments. In fact, as Lau et al. (2015, 2017) show, it is possible to use a language model, augmented with a scoring function, to predict acceptability. Standard RNN language models perform fairly well on the sentence acceptability prediction task.

By experimenting with different sorts of enrichments of the training data, one can explore their effect on both the perplexity and the predictive accuracy of the LM. For example, Bernardy et al. (2018) report that including contextual information in training and testing improves the performance of an LSTM LM on the acceptability task, when contextual information is contributed by preceding and following sentences in a document.

Here we report several experiments on the possible contribution of symbolic representations of semantic and syntactic features to the accuracy of LSTM LMs in predicting human sentence acceptability judgments. [1]

For semantic tags, we use the Universal Semantic Tagging scheme, which provides language independent and fine-grained semantic categories for individual words (Abzianidze et al., 2017). We take our syntactic roles from the Universal Dependency Grammar scheme (Nivre et al., 2016). This allows us to assign to each word in a sentence a semantic and a syntactic role, respectively.

Our working hypothesis is that for a language model the syntactic and semantic annotations will highlight semantic and syntactic patterns observed in the data. Therefore sentences that exhibit these patterns should be more acceptable than sentences which diverge from them. One would expect that if we get lower perplexity for one of the tagging scheme LMs, then its performance would improve on the acceptability prediction task. Clearly, better performance on this task indicates that tagging supplies useful information for predicting acceptability.

2 Experimental Setup

First, we train a set of language models, some of them on tag annotated corpora, and some on plain text. While we are interested in the effect of the tags on model perplexity, our main concern is to measure the influence of the tags on an LSTM

[1] Our training and test sets, and the code for generating our LSTM LM models are available at https://github.com/GU-CLASP/predicting-acceptability.

Proceedings of the 22nd Nordic Conference on Computational Linguistics (NoDaLiDa), pages 76–85
Turku, Finland, 30 September – 2 October,

LM's predictive power in the sentence acceptability task.

We implement four variants of LSTM language models. The first model is a plain LSTM that predicts the next word based on the previous sequence of words. The second, third and fourth models predict the next word w_i conditioned on the previous sequence of words and tags, for which we write $P_M(w_i)$. For a model M that uses syntactic or semantic information:

$$P_M(w_i) = P(w_i|(w_{i-1}, t_{i-1}), ..., (w_{i-n}, t_{i-n})) \tag{1}$$

We stress that the current tag (t_i) is not given when the model predicts the current word (w_i).

Using the main hyperparameters from a previous similar experiment (Bernardy et al., 2018), all language models use a unidirectional LSTM of size 600. We apply a drop-out of 0.4 after the LSTM layer. The models are trained on a vocabulary of 100,000 words. We randomly initialise word embeddings of size 300 dimensions, and tag embeddings of size 30 dimensions. Each model is trained for 10 epochs.

Following the literature on acceptability (Lau et al., 2015, 2017; Bernardy et al., 2018), we predict a judgment by applying a variant of the scoring function SLOR (Pauls and Klein, 2012) to a model's predictions.

2.1 SLOR

To estimate sentence acceptability, we use a length-normalized *syntactic log-odds ratio* (hereafter simply referred to as SLOR). We use SLOR rather than any other measurements since it was shown to have the best results in a previous study (Lau et al., 2015). It is calculated by taking the logarithm of the ratio to the probability of the sentence s predicted by a model M (P_M) with the probability predicted by the unigram model (P_U), divided by the length of the sequence $|s|$.

$$SLOR_M(s) = \frac{\log(P_M(s)) - \log(P_U(s))}{|s|} \tag{2}$$

where $P_M(s) = \prod_{i=1}^{|s|} P_M(w_i)$, and $P_U(s) = \prod_{i=1}^{|s|}(P_U(w_i))$. This formula discounts the effect of both word frequency and sentence length on the acceptability score that it assigns to the sentence. SLOR has been found to be a robustly effective scoring function for the acceptability prediction task (Lau et al., 2015).

2.2 Model evaluation

We evaluate the model by calculating the Weighted Pearson correlation coefficient between the SLOR score assigned by the model and the judgments assigned by the annotators.

Even though we show only the mean judgment in Figure 3, each data point comes also with a variance (there is heteroscedasticity). Thus we have chosen to weight the data points with the inverse of the variance when computing the Pearson correlation, as is standard when computing least square regression on heteroscedastic data.

We report the weighted correlation point wise between all models, and between each model and the human judgments. Additionally, we perform three experiments where we shuffle the syntactic and semantic representations in the test sentences. This is done to determine if the tags provide useful information for the task.

2.3 Language Model Training Data

For training the LMs we selected the English part of the CoNLL 2017 dataset (Nivre et al., 2017). The input sentences were taken from a subset of this corpus. We used only 1/10 of the total CoNNL 2017 Wikipedia corpus, randomly selected. We took out all sentences whose dependency root is not a verb, thus eliminating titles and other non-sentences. We also removed all sentences longer than 30 words. After filtering, the training data contained 87M tokens and 5.3M sentences.

3 Semantic Tags

We train a LSTM model for predicting semantic tags. We use this model to tag both the training set extracted from the CoNLL 2017 corpus, and the crowdsource annotated test set (described in Section 6).

The Universal semantic tagging scheme provides fine-grained semantic tags for tokens. It includes 80 different semantic labels. The semantic tags are similar to Part-of-Speech (POS) tags, but they are intended to generalise and to semantically disambiguate POS tags. For many purposes, POS tags do not provide enough information for semantic processing, and this is where semantic tags come into play. A significant element of POS disambiguation consists in assigning proper nouns to semantic classes (named entities). In this way, the scheme also provides a form of named entity recognition. The scheme is designed to be language independent. Annotations currently exist for English, German, Dutch and Italian, but we only use the English labels in our model.

The corpus of semantically tagged sentences that we use comes from the Parallel Meaning Bank (PMB) (Abzianidze et al., 2017). It contains 1.4M tagged tokens divided into 68,177 sentences[2]. The dataset is extracted from a variety of sources: Tatoeba, News Commentary, Recognizing Textual Entailment (RTE), Sherlock Holmes stories, and the Bible. The sentences are split into gold and silver annotations, where the gold has been manually annotated, and the silver has been annotated by a parser with manual corrections. The silver annotations are mostly correct, but may contain some errors.

Example (1) below is a semantically tagged sentence, taken from the PMB corpus. It includes two pronouns 'he' and 'his'. Both of these instantiate the same POS, but their semantic classes are distinct. The first is a simple third person pronoun, while the second is a possessive pronoun. Semantic tags are able to handle this distinction, by assigning PRO (pronoun) to the third person pronoun, and HAS (possessive) to the possessive pronoun.

(1)	He	took	his	book	.
	PRO	EPS	HAS	CON	NIL

3.1 Semantic Tagging Model

To assign semantic tags to the CoNNL 2017 training corpus and our training set we use a bidirectional LSTM of size 256, with a standard configuration. The model is trained with a batch size of 512 sentences. The word embeddings are of size 256 and are randomly initialized. The model is implemented with `keras` (Chollet et al., 2015). We stress that this model is separate from the language models used to predict sentence acceptability.

The semantic tagging model is trained for a maximum of 1024 epochs, with early stopping if the validation loss does not improve after 32 epochs. For each epoch, we feed the model 64 batches of 512 randomly selected sentences. The model observes 32,768 sentences (e.g. roughly half of the corpus) per epoch. To select the best model we left out 1024 gold annotated sentences, randomly selected, and we used them for validation.

Performance The model was validated on 1.5% of the sentences with gold annotations. The remaining data were used for training. This split was chosen because the primary goal of this model is a downstream task, namely tagging data for language modeling. We wish to maximise the number of sentences in the training data.

The model finished after 33 epochs, with a final validation loss of 0.317 and a validation accuracy of 91.1%. The performance of our model is similar to that of (Bjerva et al., 2016).

4 Syntactic Tags

To introduce syntactic information into our model in an explicit way, we provide it with Universal Dependency Grammar (UD) roles. The UD annotation scheme seeks to develop a unified syntactic annotation system that is language independent (Nivre et al., 2016). UD implements syntactic annotation through labelled directed graphs, where each edge represents a *dependency relation*. In total, UD contains 40 different dependency relations (or tags). For example, the sentence *'There is no known cure'* (taken from the CoNLL2017 Wikipedia corpus) is annotated as the dependency graph shown in Figure 1.

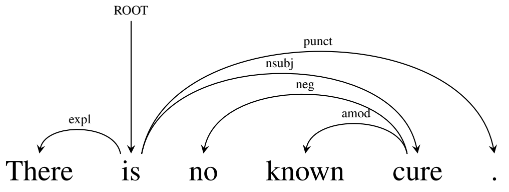


Figure 1: Dependency Graph

The model gives the label of the dependency originating from each word, which we call the *syntactic role* of the word. This label is provided as an additional feature for each word *in the input* to our language model. The model does not attempt to predict these roles. For the above sentence, the information given to our syntactic tag trained models would be:

There	is	no	known	cure
expl	root	neg	amod	nsubj

We use the Stanford Dependency Parser (Chen and Manning, 2014) to generate syntactic tags for

[2] Available for download at `https://pmb.let.rug.nl/releases/sem-0.1.0.zip`

the training and test sets.

5 Syntactic Depth

In addition to using syntactic and semantic tags, we also experiment with syntactic depth. To assign a depth to word n, we compute the number of common ancestors in the tree between word n and word $n + 1$. The last word is arbitrarily assigned depth 0. This method was proposed by Gómez-Rodríguez and Vilares (2018) for constituent trees, but the method works just as well for dependency trees. An example tree is shown below:

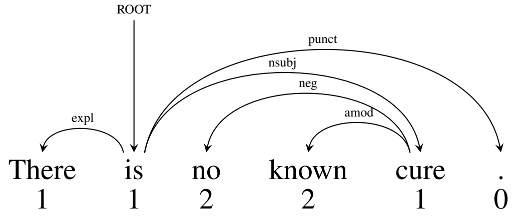


Figure 2: Linearized dependency graph

6 Test Set

The test set for evaluating our LMs comes from the work of Lau et al. (2015, 2017). 600 sentences were extracted from the BNC corpus (BNC Consortium, 2007) and filtered for length ($8 < |s| < 25$). After this filtering 500 sentences remained and were put through a round-trip machine translation process, from English to Norwegian, Spanish, Chinese or Japanese, and then back to English. In total, the test set contains 2500 sentences: 500 original sentences, and 500 from each language used for round-trip translation (i.e. Norwegian, Spanish, Chinese and Japanese). The purpose of using round-trip MT is to introduce a wide variety of infelicities into some of the sentence in our test set. This insures variation in acceptability judgements across the examples of the set.

We used Amazon Mechanical Turk (AMT) crowdsourcing to obtain acceptability judgments. The annotators were asked to rate the sentences based on their naturalness (as opposed to the theoretically committed notion of *well-formedness*) on a scale of 1 to 4. On average, each sentence had 14 annotators after filtering (for a more detailed description see (Lau et al., 2017)).

The results are shown in Table 1. The original sentences, and the sentences that were round-trip translated through Norwegian and Spanish have a higher mean rating than the sentences translated through Japanese and Chinese. The standard deviation is slightly higher for all the sentences which underwent round-trip translation, which is to be expected.

Table 1: Mean judgments and standard deviation for the test set.

SENTENCES	MEAN	ST-DEV
en	3.51	0.46
en-no-en	3.13	0.70
en-es-en	3.12	0.69
en-zh-en	2.42	0.72
en-ja-en	2.14	0.74

7 Results

Below we denote the plain LSTM LM by LSTM, the LM with syntactic tags as +SYN, the LM with semantic tags as +SEM, and the LM with syntactic tree depth as +DEPTH. We denote the models with shuffled tags by using the star (*) as a modifier.

7.1 Language Model Perplexity

We report in Table 3 the training loss for the plain-LSTM language model, and for the LSTM language models enhanced with syntactic and semantic tags. At the end of the training, the language model conditioned on syntactic tags shows the lowest loss. By definition loss is the logarithm of the perplexity. The semantic tag LM exhibits the highest degree of loss. It seems that the syntactic tags reduce LM perplexity, while the semantic tags increase it.

7.2 Acceptability Predictions

The matrix in Table 2 gives the results for the sentence acceptability prediction task. Each entry r^i_j indicates the weighted Pearson correlation r between $SLOR_i$ and $SLOR_j$. Scatter plots showing the correlation between human and model predictions are given in Figure 3

The plain LSTM performs close to the level that Bernardy et al. (2018) report for the same type of LM, trained and tested on English Wikipedia data. This indicates the robustness of this model for the sentence acceptability prediction task, given that, unlike the LSTM of Bernardy et al. (2018), it is trained on Wikipedia text, but tested on a BNC test set. Therefore, it sustains a relatively high level of performance on an out of domain test set.

Table 2: Weighted Pearson correlation between prediction from different models on the SMOG1 dataset. * indicates that the tags have been shuffled.

	HUMAN	LSTM	+SYN	+SYN*	+SEM	+SEM*	+DEPTH	+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	1.00

Table 3: Training loss and accuracy for the language modeling task.

MODEL	LOSS	ACCURACY
LSTM	5.04	0.24
+SYN	**4.79**	0.26
+SEM	5.23	0.21
+DEPTH	4.88	0.27

We also tested a model that combined depth markers and syntactic tags, which is, in effect, a full implicit labelled dependency tree model. Interestingly, its Pearson correlation of 0.54 was lower than the ones achieved by the syntactic tag and depth LSTM LMs individually.

None of the enhanced language models increases correlation with human judgments compared to the plain LSTM. Neither does the additional information significantly reduce correlation.

Shuffling the tags causes a drop of 0.16 in correlation for syntactic tags, and a drop of 0.1 for tree depth. Shuffling the semantic tags also lowers the correlation, but only by a small amount (-0.02).

8 Discussion

8.1 Semantic Tags

As can be observed in Table 3, the semantic tags show the highest loss during training. This indicates that semantic tags increase the perplexity of the model, and do not help to predict the next word in a sentence. Despite this, +SEM correlates fairly well with human judgments ($r = 0.54$).

The results obtained with shuffled semantic tags (+Sem*) are revealing. They yield a correlation factor nearly as high as the non-shuffled tags ($r = 0.53$). This suggests that the semantic tags *do not provide any useful information* for the prediction task. This hypothesis is further confirmed by the high correlation between the non-shuffled and the shuffled semantic tag LMs ($r = 0.96$).

The question of why semantic tags do not reduce perplexity, or why randomly assigned semantic tags are almost as good as non-shuffled tags at predicting acceptability requires further study. One possibility is that the tagging model does not perform as well on the ConLL 2017 Wikipedia subset, or the BNC test set, as it does on the PMB corpus. It may be the case that since the domains are somewhat different, the model is not able to accurately predict tags for our training and test sets. Similarly, we do not know the accuracy of the Stanford Dependency Parser on the BNC test set.

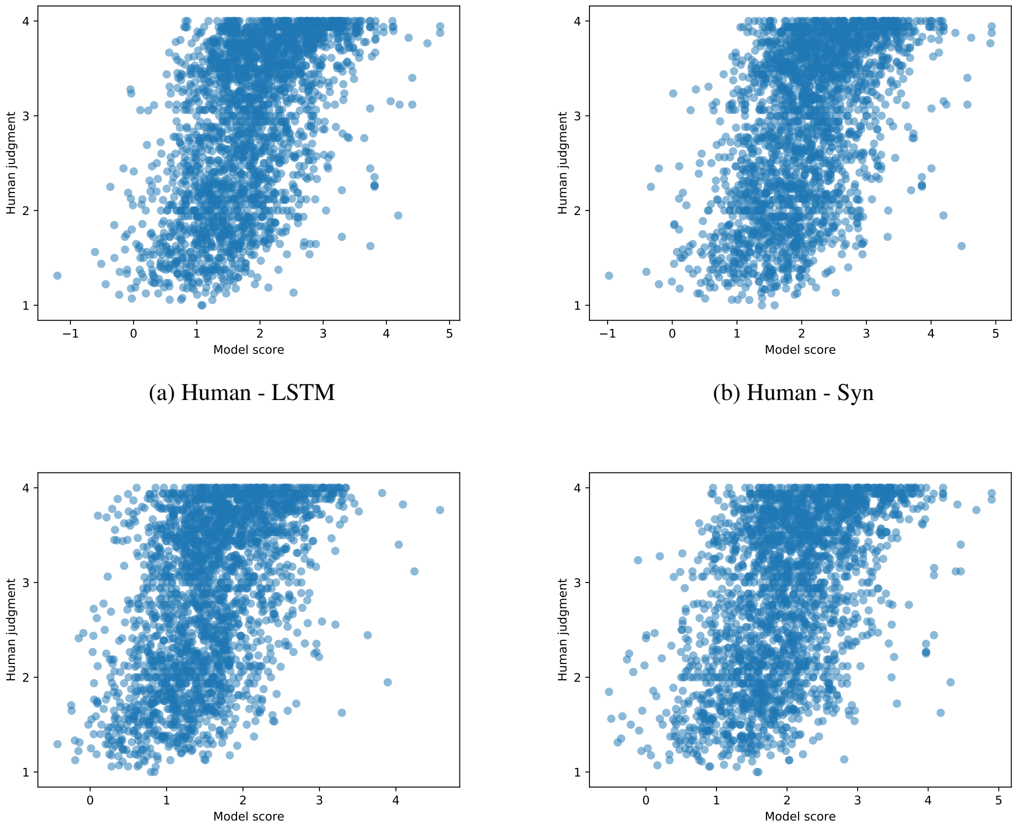
8.2 Syntactic Tags

Providing syntactic tags improves the language model, but not the correlation of its predictions with mean human acceptability judgments. However, shuffling the syntactic tags does lower the correlation substantially. This indicates that syntactic tags significantly influence the predictions of the language model.

8.3 Tree Depth

The depth marker enriched LSTM performs best of all the feature enhanced models. Shuffling the markers significantly degrades accuracy, and the non-shuffled depth model achieves a reduction in perplexity. However, it still performs below the simple LSTM on the acceptability prediction task

It may be the case that the plain LSTM already acquires a significant amount of latent syntactic information, and adding explicit syntactic role labeling does not augment this information in a way that is accessible to LSTM learning. This con-



(a) Human - LSTM

(b) Human - Syn

(c) Human - Sem

(d) Human - Depth

Figure 3: Scatter plots showing the weighted Pearson correlation between human acceptability judgments (y-axis) and model predictions (x-axis).

clusion is supported by the work of Bernardy and Lappin (2017) on syntactic agreement. They observe that replacing a significant portion of the lexicon of an LSTM with POS tags degrades its capacity to predict agreement.

In general, our results do not show that syntactic and semantic information plays no role in the performance of any LM for the acceptability prediction task. It seems clear that the simple LSTM model learns both semantic and syntactic relations among words and phrases, but represents these in a distributed way through the encoding of lexical embeddings in vectors. In fact, there is a body of work which shows that such LSTMs recognise complex long-distance syntactic relations (Linzen et al., 2016; Bernardy and Lappin, 2017; Gulordava et al., 2018; Lakretz et al., 2019).

8.4 Error analysis

We analyse the models in two ways. First, we explore how they score sentences in the test set as categorised by the round-trip translation language that the sentences went through. Second, we look at two example sentences for which no model did particularly well.

8.4.1 Model performance on test sentences

To analyse the scores assigned by the model in comparison to the human judgments we first need to normalise the scores. We do this by dividing the score assigned to each sentence by the maximum score assigned. Thus, the relative score of a sentence indicates how close it is to the highest acceptability judgment.

The mean relative score of the human judgments and model scores are presented in Table 4. We observe that the models generally appear to assign a lower relative score than humans. But all models also appear to follow the general trend of human judgments and assign a lower score to the Chinese and Japanese round-trip translated sentences compared to the Spanish, Norwegian and original sentences. However, looking at the numbers the difference in magnitude for Chinese and

Japanese sentences is rather large. The Chinese and Japanese sentences have a lower relative score of 0.27 and 0.35 respectively. But for models, this difference is only ≈ 0.07 and ≈ 0.12 respectively. This indicates that while the models are able to see some acceptability differences between the subclasses of test sentences, the models do not penalize these sentences as much as humans.

Table 4: Comparison of the average relative score assigned by the models and humans for the different sentences in the test set.

MODEL	EN	NO	ES	ZH	JA
Human	0.88	0.78	0.78	0.61	0.53
LSTM	0.41	0.40	0.40	0.34	0.29
+SYN	0.46	0.44	0.45	0.39	0.35
+SEM	0.39	0.36	0.37	0.30	0.28
+DEPTH	0.45	0.43	0.44	0.38	0.34

We also note that the models consistently assign much lower relative scores than the human annotators do to most of the sentences. This, biases their scores in favour of the Chinese and Japanese target sentences, since these are typically 'worse' than their original English sources, or the Norwegian and Spanish targets, according to the human judges (see Table 1).

We also compare the worst scoring sentences between the models. This was done by splitting the predictions into two sets: (a) model scores above the average[3] and (b) model scores below the average. We sort these sets by their difference to the humans and select the top 20 sentences for each model. Table 5 shows the intersection of sentence sets for the different models.

Table 5: Shared erroneous sentences between the models.

MODEL	LSTM	+SYN	+SEM	+DEPTH
LSTM	40			
+SYN	30	40		
+SEM	19	15	40	
+DEPTH	30	28	17	40

We observe that the syntactic tag and depth models share many sentences with each other, and with the plain LSTM, but not as many with the semantic model. This shows that the difficult sentences for the semantic model are different than those for the syntactic and plain models.

8.4.2 Model and human performance

We use the relative scores from the previous section to select sentences for examination. We look at two types of cases, one in which the model predicts a higher score than the human judgments, and the other where the model predicts a lower score than human judgments. For both cases we select a sentence at random.

We begin by considering an example to which the model assigns a higher score than humans do. The sentence went through Chinese:

(1) '1.5% Hispanic or Latino of any race population.'

The sentence lacks a verb, and the modifier-noun construction 'race population' is lexically strange. It is interesting to note that our syntactic models (+SYN and +DEPTH) both assign a high score to this sentence, while the semantic and plain LM assign a lower score (which is closer to the human judgment). We would think that the model using syntactic tags would pick up on the missing verb, and so penalize the sentence. The scores for the sentence (1) are shown in Table 6:

Table 6: Human judgments and model scores for sentence (1).

MODEL	RELATIVE	ABSOLUTE
HUMAN	0.40	1.62
LSTM	0.77	3.74
+SYN	0.90	4.47
+SEM	0.71	3.29
+DEPTH	0.85	4.17

For (1), the LM enhanced with semantic tags gave the sentence the lowest score. The syntactic and depth model gave the sentence a high score (0.90 and 0.85 respectively). This indicates that while still assigning the sentence a relatively high score, the semantic and plain LM rate the sentence closer to humans than the syntactical LM.

In the second case, (2), the sentence is one of the original English sentences:

(2) 'ACS makes a special "FAT" heavy duty BMX freewheel in 14T and 16T with 3/16 "teeth compatible only with 3/16" chains.'

[3] We compare scores by dividing each score by it's maximum value, as described previously.

The human annotators gave it an appropriately high score, but the models did not, as indicated in table Table 7.

Table 7: Human judgments and model scores for sentence (2).

MODEL	RELATIVE	ABSOLUTE
HUMAN	0.80	3.23
LSTM	-0.007	-0.03
+SYN	0.002	0.01
+SEM	0.26	1.20
+DEPTH	0.02	-0.01

Again, we can see that the LM enhanced with semantic tags performed the best (i.e. assigned the sentence the highest score). The sentence has a few features which might make it difficult for the standard LM and syntactically enhanced language models. The sentence contains a high number of quotations, acronyms (e.g. ACS) and specialized terms (e.g. 3/16). The dependency tags do not treat these words in any special way. Because the words are rare they are not likely candidates. The semantic tags will treat these words in a different manner, since it contains tags for named entities and quantities.

8.5 Pre-Trained Language Models

Recently several large pre-trained language models using transformation architecture, like BERT (Devlin et al., 2018), or bidirectional LSTM with attention, such as ELMo (Peters et al., 2018), have achieved state of the art results across a variety of NLP tasks. We opted not to experiment with any of these pre-trained language models for our task. The LSTM architecture of our LMs is far simpler, which facilitates testing the contribution of explicit feature representation to correlation in the acceptability prediction task, and perplexity for the language modeling task.

9 Related Work

There has been a considerable amount of work showing that encoding tree representations in deep neural networks, particularly LSTMs, improves their performance on semantic relatedness tasks. So, for example, Tai et al. (2015) show that Tree-LSTMs outperform simple LSTMs on SemEval 2014 Task 1, and sentiment classification. Similarly, Gupta and Zhang (2018) argue that by adding progressive attention to a Tree-LSTM it is possible to improve its performance on several semantic relatedness tasks.

Williams et al. (2018) describe a number of experiments with latent tree learning RNNs. These models learn tree structures implicitly, rather than through training on a parse annotated corpus. They construct their own parses. Williams et al. (2018) state that they outperform Tree-LSTM and other DNN models on semantic relatedness applications, and the Stanford Natural Language Inference task. Interestingly, the parse trees that they construct are not consistent across sentences, and they do not resemble the structures posited in formal syntactic or semantic theories. This result is consistent with our finding that LSTMs learn syntactic and semantic patterns in a way that is quite distinct from the classifications posited in classical grammatical and semantic systems of representation.

Finally, Warstadt and Bowman (2019) discuss the performance of several pre-trained transformer models on 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.

It is important to recognise that CoLA is a very different sort of test set from the one that we use in our experiments. It is drawn from linguists' examples intended to illustrate particular sorts of syntactic construction. It is annotated for binary classification according to linguists' judgments. By contrast, our 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. It is annotated through AMT crowd sourcing with gradient acceptability judgments. Given these significant differences in design and annotation between the two test sets, applying our models to CoLA would have taken us beyond the scope of the sentence acceptability task, as specified in (Lau et al., 2015, 2017; Bernardy et al., 2018),

Moreover, our experiments are not focused on identifying the best performing model as such. Instead, we are interested in ascertaining whether enriching the training and test data with explicit syntactic and semantic classifier representations contributes to LSTM learning for the sentence acceptability prediction task.

10 Conclusions

We present experiments that explore the effect of enhancing language models with syntactic and semantic tags, and dependency tree depth markers, for the task of predicting human sentence acceptability judgments. The experiments show that neither syntactic nor semantic tags, nor tree depth indicators improve the correlation between an LSTM LM and human judgments. Our experiments also show that syntactic tags provide information that is useful for language modeling, while semantic tags do not. However, further experiments are needed to verify our results for semantic tags. The model that we used for tagging, rather than the information in the tags themselves, may be responsible for the observed result.

Surprisingly our initial hypothesis that lower training perplexity produces better acceptability prediction has been overturned. We have not observed any correlation between the perplexity of an LM and its accuracy in acceptability prediction. The SLOR scoring function may mask an underlying connection between preplexity and prediction accuracty.

Our tentative conclusion from these experiments is that simple LSTMs already learn syntactic and semantic properties of sentences through lexical embeddings only, which they represent in a distributional manner. Introducing explicit semantic and syntactic role classifiers does not improve their capacity to predict the acceptability of sentences, although such information may be useful in boosting the performance of deep neural networks on other tasks.

In future work, we plan to test other sources of information for the language models. One possibility is to use constituency, rather than dependency tree depth. We also plan to experiment with different combinations of tags for the language models, such as models that use both semantic and syntactic roles.

11 Acknowledgments

The research reported in this paper was supported by grant 2014-39 from the Swedish Research Council, which funds the Centre for Linguistic Theory and Studies in Probability (CLASP) in the Department of Philosophy, Linguistics, and Theory of Science at the University of Gothenburg. We are grateful to Stephen Clark for helpful advice on encoding syntactic trees in lexical embedding vectors for LSTMs. We would also like to thank our colleagues at CLASP for helpful discussion of some of the ideas presented here.

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik Van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The parallel meaning bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. *arXiv preprint arXiv:1702.03964*.

Jean-Philippe Bernardy and Shalom Lappin. 2017. Using deep neural networks to learn syntactic agreement. *Linguistic Issues In Language Technology*, 15(2):15.

Jean-Philippe Bernardy, Shalom Lappin, and Jay Han Lau. 2018. The influence of context on sentence acceptability judgements. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 456–461.

Johannes Bjerva, Barbara Plank, and Johan Bos. 2016. Semantic tagging with deep residual networks. *arXiv preprint arXiv:1609.07053*.

BNC Consortium. 2007. The british national corpus, version 3 (bnc xml edition). 2007. *Distributed by Oxford University Computing Services on behalf of the BNC Consortium.*

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

François Chollet et al. 2015. Keras. `https://keras.io`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Carlos Gómez-Rodríguez and David Vilares. 2018. Constituent Parsing as Sequence Labeling. *arXiv:1810.08994 [cs]*. ArXiv: 1810.08994.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Amulya Gupta and Zhu Zhang. 2018. To attend or not to attend: A case study on syntactic structures for semantic relatedness. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2116–2125, Melbourne, Australia. Association for Computational Linguistics.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019. The emergence of number and syntax units in lstm language models. *arXiv preprint arXiv:1903.07435*.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2014. Measuring gradience in speakers grammaticality judgements. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 36.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2015. Unsupervised Prediction of Acceptability Judgements. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1618–1628, Beijing, China. Association for Computational Linguistics.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, Acceptability, and Probability: A Probabilistic View of Linguistic Knowledge. *Cognitive Science*, 41(5):1202–1241.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Joakim Nivre, Željko Agić, Lars Ahrenberg, and et al. 2017. Universal dependencies 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan T McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *LREC*.

Adam Pauls and Dan Klein. 2012. Large-scale syntactic language modeling with treelets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 959–968. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China. Association for Computational Linguistics.

Alex Warstadt and Samuel R. Bowman. 2019. Grammatical analysis of pretrained sentence encoders with acceptability judgments. *CoRR*, abs/1901.03438.

Adina Williams, Andrew Drozdov, and Samuel R Bowman. 2018. Do latent tree learning models identify meaningful structure in sentences? *Transactions of the Association of Computational Linguistics*, 6:253–267.