Unsupervised Prediction of Acceptability Judgements

Jey Han Lau, Alexander Clark, and Shalom Lappin

jeyhan.lau@gmail.com, alexsclark@gmail.com, shalom.lappin@kcl.ac.uk
King's College London

Abstract

In this paper we present the task of un-supervised prediction of speakers' accept-ability judgements. We use a test set generated from the British National Cor-pus (BNC) containing both grammatical sentences and sentences containing a va-riety of syntactic infelicities introduced by round trip machine translation. This set was annotated for acceptability judge-ments through crowd sourcing. We trained a variety of unsupervised language mod-els on the original BNC, and tested them to see the extent to which they could pre-dict mean speakers' judgements on the test set. To map probability to acceptability, we experimented with several normalisa-tion functions to neutralise the effects of sentence length and word frequencies. We found encouraging results with the unsu-pervised models predicting acceptability across two different datasets. Our method-ology is highly portable to other domains and languages, and the approach has po-tential implications for the representation and the acquisition of linguistic knowl-edge.

1 Introduction

Language modelling involves predicting the prob-ability of a sentence. Given a trained model, we can infer the quantitative likelihood that a sentence occurs. Acceptability, on the other hand, indicates the extent to which a sentence is permissible or ac-ceptable to native speakers of the language. While acceptability is affected by frequency and exhibits gradience (Keller, 2001; Sprouse, 2007; Lau et al.,

2014), there is limited research on the relationship between acceptability and probability. In this pa-per, we consider the the task of unsupervised pre-diction of acceptability.

Speakers have robust intuitions about accept-ability, and acceptability has been consistently rated on various scales (Sprouse and Almeida, 2012). The acceptability of a sentence appears to be relatively unaffected by its length (within cer-tain bounds), or the frequency of its words, prop-erties that we have confirmed experimentally. By contrast sentence probability does depend on these factors. To filter the effects of sentence length and word frequency, we devise normalising functions to map the probability of a sentence (inferred by our unsupervised language models) to an accept-ability score.

Keller (2001) and Lau et al. (2014) present ev-idence that acceptability exhibits gradience. Ac-cordingly, we treat acceptability as a continuous variable here. We train a variety of unsuper-vised models for the acceptability prediction task, and we assess the performance of these models by measuring the correlation between their nor-malised acceptability scores and the mean crowd-sourced acceptability judgements on a set of test sentences.

There are a number of NLP tasks to which our work can be fruitfully applied. It can be used to evaluate the fluency of the output for machine translation and other language generation systems. It could also contribute to automatic essay scoring, and to second language tutorial systems.

There are several reasons to favour unsuper-vised models. From an engineering perspective, unsupervised models offer greater portability to other domains and languages. Our methodology takes only unannotated text as input. Extending

our methodology to other domains/languages is therefore straightforward, as it requires only a raw training corpus in that domain/language.

Our work may also have significant implications for the cognitive foundations of the representation and acquisition of linguistic knowledge. The unannotated training corpora of our language models are impoverished input in comparison to the data available to humans language learners, who learn from a variety of data sources (visual and auditory cues, interaction with adults and peers in a non-linguistic environment, etc). If an unsupervised language model can reliably predict human acceptability judgements, then it provides a benchmark of what humans could, in principle, achieve with the same learning algorithm.

Success in this task raises interesting questions about the nature of grammatical knowledge. If acceptability judgments can be accurately modeled through these techniques, then it seems unnecessary to posit an underlying binary model of syntax which enumerates all and only the set of well-formed sentences. Instead it is reasonable to suggest that humans represent linguistic knowledge as a probabilistic, rather than as a binary system. Probability distributions provide a natural explanation of the gradience that characterises acceptability judgements. Gradience is intrinsic to probability distributions, and to the acceptability scores that we derive from these distributions.

While our results raise important questions concerning the nature of syntactic representation and of language acquisition, we leave them open for further research. We refrain from making strong claims on cognitive issues here. Clearly additional psychological evidence is required to motivate substantive conclusions on these issues, even if our results suggest them.

Our focus in this paper is on the task of predicting speakers' acceptability judgements through unsupervised language models. We take this to be a problem in natural language processing, whose solution has useful applications in language technology. All models described in this paper are implemented in an open source toolkit.[1]

We describe our dataset in Section 2, which consists of crowd sourced acceptability judgments applied to sentences with errors introduced through round trip machine translation. We de-

scribe the models and their results in Section 3. In Section 4 we present results with a different corpus based on English Wikipedia. The new dataset shows that our observations generalise to another domain. We compare our methodology to a supervised system in the acceptability prediction task in Section 5. We look more closely at the influence of sentence length and lexical frequency in Section 6, and we show that the normalising functions succeed in neutralising these effects. Finally, we discuss the implications of our results, and draw conclusions from them in Section 7 and Section 8.

2 Dataset and Methodology

For our experiments, we require a collection of sentences that exhibit varying degrees of grammatical well-formedness. We use the dataset that we discuss in Lau et al. (2014). We translated British National Corpus (BNC Consortium, 2007) English sentences to four other languages – Norwegian, Spanish, Japanese and Chinese – and then back to English using Google Translate. To collect human judgements of acceptability for the sentences, we used Amazon Mechanical Turk. A total of 2,500 sentences were annotated.

Three modes of presentation were used for rating a sentence: (1) binary with two options (unnatural vs. natural); (2) four options (extremely unnatural, somewhat unnatural, somewhat natural and extremely natural); and (3) a sliding scale with two extremes (extremely unnatural and extremely natural). To aggregate the ratings over multiple speakers for each sentence, we computed the arithmetic mean. As there is a high correlation of mean ratings among different modes of presentation, we take the judgements for the four-option mode of presentation as the gold-standard for our experiments.

To predict the ratings of the 2,500 test sentences, we trained several probabilistic models on the BNC, and then used the trained models to infer the probabilities of the test sentences. Models are trained on the written portion of the BNC, which has approximately 100 million words (henceforth referred to as BNC-100M).[2] We used only the words, and no forms of annotation information in the BNC, as input to training.

We first experiment with simple lexical N-gram models, and then move to Bayesian and neural

---

[1]This toolkit can be accessed at `https://github.com/jhlau/acceptability_prediction`.

[2]We removed sentences with less than 8 words, as well as the 2,500 test sentences, from the training data.

network models, increasing the complexity of the models to better capture word dependencies.

To translate probability into acceptability scores, we compute several *acceptability measures* extracted from the model parameters. The acceptability measures are variants of the sentence's log probability, devised to normalise sentence length and low frequency words. These measures are summarised in Table 1. For each measure (including *LogProb* as a baseline) we compute its Pearson correlation coefficient with the gold standard sentence mean rating to evaluate its effectiveness in predicting acceptability.

We tokenised the training data (BNC-100M) and the test sentences using OpenNLP, and we converted all words to lower case. To address out of vocabulary (OOV) words that are seen in the test sentences but not in the training data, we adopt the Berkeley Parser approach, where we replace low frequency or OOV words using the *UNK* signature. We capture additional surface characteristics of the original word by attaching features at the end of the signature (e.g. the OOV word *1949* would be replaced by the signature *UNK-NUM*).[3]

3 Unsupervised Models

3.1 Lexical N-gram Model

Lexical N-gram models were variously explored in tasks related to acceptability estimation (Heilman et al., 2014; Clark et al., 2013; Pauls and Klein, 2012). We use an N-gram model with Kneser-Ney interpolation (Goodman, 2001), and we train bigram, trigram, and 4-gram models on BNC-100M. The trained models are then used to compute the acceptability measures of the test sentences.

The results are detailed in Table 2 (columns: "2-gram", "3-gram" and "4-gram").[4] In general across all models, the *Norm LP (Div)* and *SLOR* measures consistently produce the best correlations.

We see a significant improvement when the context window is increased from 2-gram to 3-gram, but not so from 3-gram to 4-gram (2-gram best: 0.34; 3-gram best: 0.42; 4-gram best: 0.42). This result implies that increasing the context window

Acc. Measure	Equation
LogProb	$\log P_m(\xi)$
Mean LP	$\dfrac{\log P_m(\xi)}{|\xi|}$
Norm LP (Div)	$-\dfrac{\log P_m(\xi)}{\log P_u(\xi)}$
Norm LP (Sub)	$\log P_m(\xi) - \log P_u(\xi)$
SLOR	$\dfrac{\log P_m(\xi) - \log P_u(\xi)}{|\xi|}$

Table 1: Acceptability measures for predicting the acceptability of a sentence. Notations: *SLOR* is the syntactic log-odds ratio, introduced by Pauls and Klein (2012); ξ is the sentence ($|\xi|$ is the sentence length); $P_m(\xi)$ is the probability of the sentence given by the model; $P_u(\xi)$ is the unigram probability of the sentence. Note that the negative sign in *Norm LP (Div)* is given to reverse the sign change introduced by the division of log unigram probabilities.

of the lexical N-gram model does not correspond to a better representation of grammatical structure (insofar as the size of the dataset is fixed), and a more sophisticated model is necessary.

3.2 Bayesian HMM

Seeing that local context is insufficient for predicting acceptability, we explore various Bayesian models that incorporate richer latent structures. We chose a Bayesian implementation because its "rich gets richer" dynamics tends to work well for languages (Goldwater and Griffiths, 2007; Goldwater et al., 2009; Newman et al., 2012; Lau et al., 2012).

Lexical N-grams model the generation of a word based on its preceding words. We introduce a layer of latent variables on top of the words, which can be interpreted as the word classes, and we model the transitions between the latent variables and observed words using Markov processes. In this model we first generate a (latent) word class based on its preceding word classes, and we then generate the word based on its word class. Figure 1(b) illustrates the structure of a second order Hidden Markov model (HMM).
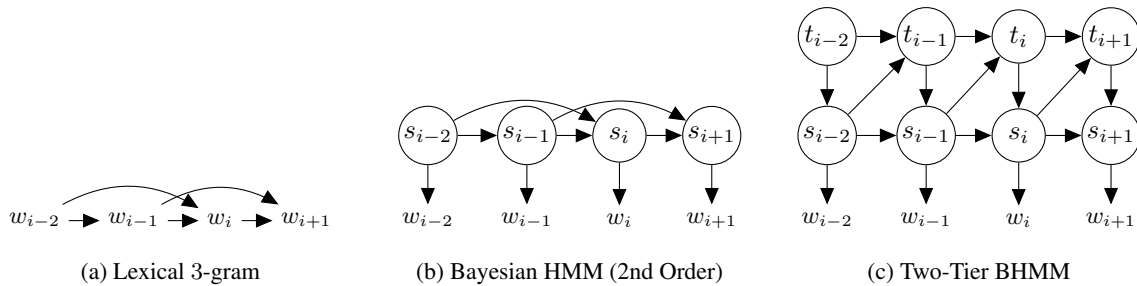
---

[3]Low frequency words are defined as words occurring less than 4 times in the BNC training data. A total of 15 features are used for the *UNK* signature.

[4]We do not present model perplexity values in the results, as we did not find any correlation between perplexity and task performance.

Figure 1: A comparison of word structures for 3-gram, BHMM and Two-Tier BHMM. w = observed words; s = tier-1 latent states ("word classes"); t = tier-2 latent states ("phrase classes").

For comparison, the structure of a lexical 3-gram model is given in Figure 1(a).

Goldwater and Griffiths (2007) propose a Bayesian approach for learning the HMM structure. The authors found that their Bayesian HMM (BHMM) significantly outperforms a HMM trained with Maximum Likelihood Estimation in unsupervised part-of-speech tagging. We adopt the methodology of Goldwater and Griffiths (2007), and train a 2nd order BHMM for our task, using collapsed Gibbs sampling for inference. BHMM has two sets of multinomials: the state transition multinomials and the word emission multinomials. To generalise the state transition probabilities for start probabilities, we use dummy words/states for empty preceding words/states.

BHMM has 3 parameters: (1) S, the number of latent states; (2) γ, the Dirichlet hyper-parameter for the state transition multinomials; and (3) δ, the Dirichlet hyper-parameter for the word emission multinomials. We assume symmetric Dirichlet priors for the hyper-parameters, and optimise the 3 parameters based on test perplexity using a greedy search approach, i.e. we optimise locally for one parameter at each stage, while keeping the default or previously optimised values for other parameters.[5] For the optimisation step models are trained using 10% of the full BNC (BNC-10M) for 2,000 iterations.[6]

Using the optimised parameters, we train BHMM on BNC-100M for 10,000 iterations. For test inference, we run the Gibbs sampler using the trained model for 5,000 iterations, and take 50 samples from the final 500 iterations (with a lag of 10 iterations). In each of the samples, we compute the test probabilities and acceptability mea-

sures using the MAP estimated states.[7] The final probabilities are computed as a harmonic mean of probabilities over the 50 samples.

We summarise the correlation results in Table 2 (column: "BHMM"). Compared to the N-gram models, we see an improvement in the correlation, indicating that the introduction of a layer of (latent) word classes produces a better structure for modelling acceptability.

3.3 LDAHMM and LDA

To better understand the role of semantics in acceptability, we experimented with LDAHMM (Griffiths et al., 2004), a model that combines syntactic and semantic dependencies between words.

The generative method of LDAHMM to generate a word in a document is to first decide whether to generate a syntactic state or a semantic state for the word. For the former, follow the HMM process to generate a state, and generate the word based on the chosen state. For the latter, follow the LDA (Blei et al., 2003) process to generate a topic based on the document's topic mixture, and generate the word based on the chosen topic.

We use a second order HMM for the HMM part and Collapsed Gibbs sampling for performing inference. LDAHMM has 4 sets of multinomials: the HMM multinomials (state transition and word emission) and the LDA multinomials (document-topic and topic-word).

LDAHMM has 6 parameters to optimise: (1) K the number of topics; (2) S the number of syntactic states (semantic state has only 1 state, designated as state 0); (3) α, the Dirichlet hyper-parameter for document-topic multinomials; (4) β, the Dirichlet hyper-parameter for topic-word multinomials; (5) γ, the Dirichlet hyper-

---

[5]When optimising for a parameter, we experimented with 4–6 values of various orders of magnitudes.

[6]The optimised parameters are: $S = 100$, $\gamma = 1.0$ and $\delta = 0.01$.

[7]As computing full probabilities gave little difference in the final outcome, we adopted the computationally more efficient MAP approach.

Measure	2-gram	3-gram	4-gram	BHMM	LDA	LDAHMM	2T	Chunker	RNNLM	PCFG*
LogProb	0.24	0.30	0.32	0.25	0.09	0.21	0.26	0.32	0.32	0.21
Mean LP	0.26	0.35	0.37	0.26	**0.14**	0.19	0.31	0.42	0.39	0.18
Norm LP (Div)	0.33	**0.42**	**0.42**	0.44	0.05	**0.33**	**0.50**	0.43	**0.53**	**0.26**
Norm LP (Sub)	0.12	0.20	0.23	0.33	0.01	0.19	0.46	0.14	0.31	0.22
SLOR	**0.34**	0.41	0.41	**0.45**	0.03	**0.33**	**0.50**	0.42	**0.53**	0.25

Table 2: Pearson's r of acceptability measure and mean sentence rating for all experimented models in BNC. Boldface indicates the best performing measure. Note that PCFG is a supervised model unlike the others.

parameter for state transition multinomials; and (6) δ, the Dirichlet hyper-parameter for word emission multinomials. We follow the same approach as with BHMM for optimising, training, and testing the model.[8] Note that as LDAHMM operates with documents, the training data is partitioned into documents, and each test sentence is treated as a document.

The results are summarised in Table 2 (column: "LDAHMM"). The result shows that LDAHMM underperforms in comparison to BHMM, indicating that the incorporation of LDA did not improve the model. To understand the impact of LDA alone, we repeat the experiments using LDA and find that it performs very poorly. Results are summarised in Table 2 (column: LDA). We suspect that the low performance of LDA and LDAHMM is due to the small context window of the test documents. The LDA part is unable to generate any meaningful topic mixtures, as there is insufficient context.

3.4 Two-Tier BHMM

BHMM uses (latent) word classes to drive word generation. Exploring a richer structure, we introduce another layer of latent variables on top of the word classes. This second layer can be interpreted as phrase classes. The idea behind this model is to use these phrase classes to drive word class and word generation. An illustration of its word structure is given in Figure 1(c).

We use collapsed Gibbs sampling for performing inference. We sample the tier-1 state s and tier-2 state t separately, and the sampling equations are given as follows:

$$P(t_i|\mathbf{t_{-i}}, \mathbf{s}, \mathbf{w}, \alpha, \gamma, \delta) \propto \frac{\#(t_{i-1}, s_{i-1}, t_i) + \alpha}{\#(t_{i-1}, s_{i-1}) + T\alpha} \times$$
$$\frac{\#(t_i, s_{i-1}, s_i) + \gamma}{\#(t_i, s_{i-1}) + S\gamma} \times \frac{\#(t_i, s_i, t_{i+1}) + \alpha}{\#(t_i, s_i) + T\alpha};$$
$$P(s_i|\mathbf{s_{-i}}, \mathbf{t}, \mathbf{w}, \alpha, \gamma, \delta) \propto \frac{\#(t_i, s_{i-1}, s_i) + \gamma}{\#(t_i, s_{i-1}) + S\gamma} \times$$
$$\frac{\#(t_i, s_i, t_{i+1}) + \alpha}{\#(t_i, s_i) + T\alpha} \times \frac{\#(t_{i+1}, s_i, s_{i+1}) + \gamma}{\#(t_{i+1}, s_i) + S\gamma} \times$$
$$\frac{\#(s_i, w_i) + \delta}{\#(s_i) + W\delta}$$

where s_i, t_i are the tier-1 and tier-2 state indices; $\mathbf{s}$, $\mathbf{t}$, $\mathbf{w}$ are the assignments for all tier-1 states, tier-2 states and words, respectively (subscript $_{-i}$ means the current assignment is excluded); α, γ and δ are the Dirichlet hyper-parameters; S = number of tier-1 states; T = number of tier-2 states; W = vocabulary size; and $\#(x, [y], [z])$ are the multinomial counts.

We follow the same process for optimising, training, and testing the model, and we summarise the results in Table 2 (column: "2T").[9] We see an improved correlation relative to BHMM (BHMM best: 0.45, Two-Tier BHMM best: 0.50). In fact it has the best performance of all models thus far. This is encouraging, as it implies that the introduction of the phrase layer produces a more optimal structure for representing acceptability.

3.5 Bayesian Chunker

Goldwater et al. (2009) propose a Bayesian approach to segment words in speech streams. Newman et al. (2012) extend the approach to segment phrases – i.e. multiword units – in sentences, and they apply it to the task of index term identification and keyphrase extraction.

The core machinery of the methodology is driven by the Dirichlet Process, where segments (words in Goldwater et al. (2009) or phrases in Newman et al. (2012)) are retrieved from a cache,

---

[8]The final optimised parameters are: $K = 100$, $S = 80$, $\alpha = 0.1$, $\beta = 0.0001$, $\gamma = 0.1$, and $\delta = 0.01$.

[9]The optimised parameters: $S = 100$, $T = 60$, $\alpha = 1.0$, $\gamma = 1.0$, $\delta = 0.01$.

or newly generated. Using Gibbs sampling for inference, the sampler considers one boundary point at a time, and computes the probability of two hypotheses: H_0, for *not* generating a boundary; and H_1, for generating a boundary.

Borrowing the notation of Newman et al. (2012), given $p_\#$ is the probability of generating a segment boundary, at the boundary point between words w_x and w_y, the probability of the hypotheses is computed as follows:

$$P(H_0|H^-) = \frac{n(w_{xy}) + \alpha P_0(w_{xy})}{n + \alpha};$$

$$P(H_1|H^-) = \frac{n(w_x) + \alpha P_0(w_x)}{n + \alpha} \times \frac{n(w_y) + \alpha P_0(w_y)}{n + 1 + \alpha}$$

where H^- is all of the structure shared by both hypotheses; w_{xy} is a multiword unit consisting of w_x and w_y; n is the number of multiword tokens; α is the concentration parameter of the Dirichlet process; $n(w)$ is the count of multiword w; and $P_0(w)$ is the probability of generating a novel w. i.e. $P_0(w_{xy}) = p_\#(1 - p_\#)P(w_x)P(w_y)$.

We extend their methodology to segment *word classes* to do unsupervised chunking, motivated by the idea that a well-formed sentence contains predictable patterns of word class chunks. We extend the sampling process to incorporate transitions between chunks. Given the word classes "$c_w c_x c_y c_z$", at the boundary point between word class c_x and c_y, the hypothesis H_0 to not generate a boundary (therefore producing a single chunk c_{xy}), and the hypothesis H_1 to generate a boundary (therefore producing two chunks c_x and c_y), are computed as follows:

$$P(H_0|H^-) = \frac{\#(c_w, c_{xy}) + \beta\left(\frac{n(c_{xy}) + \alpha P_0(c_{xy})}{n + \alpha}\right)}{\#(c_w) + m\beta} \times$$
$$\frac{\#(c_{xy}, c_z) + \beta\left(\frac{n(c_z) + \alpha P_0(c_z)}{n + \alpha}\right)}{\#(c_{xy}) + m\beta};$$

$$P(H_1|H^-) = \frac{\#(c_w, c_x) + \beta\left(\frac{n(c_x) + \alpha P_0(c_x)}{n + \alpha}\right)}{\#(c_w) + m\beta} \times$$
$$\frac{\#(c_x, c_y) + \beta\left(\frac{n(c_y) + \alpha P_0(c_y)}{n + \alpha}\right)}{\#(c_x) + m\beta} \times$$
$$\frac{\#(c_y, c_z) + \beta\left(\frac{n(c_z) + \alpha P_0(c_z)}{n + \alpha}\right)}{\#(c_y) + m\beta}$$

where m = number of chunk types; n = number of chunk tokens; β is the Dirichlet hyperparameter for the chunk transition multinomials; and $\#(x, [y])$ is the count for the chunk transition multinomials.

As the model takes word classes as input, we use the word classes induced by two-tier BHMM. We follow the same process for optimising, training and testing the model.[10] The results are summarised in Table 2 (column: "Chunker"). The model produces a moderate correlation, performing on par with the lexical 4-gram model.

3.6 Recurrent Neural Network Language Model

In recent years, we have seen a resurgence in the use of neural networks for deep machine learning and NLP. Rather than designing structures or handcrafting features that seem intuitive for a task, deep learning introduces an entirely general architecture for machine learning. It has yielded some impressive results for NLP tasks: automatic speech recognition, parsing, part of speech tagging, and named entity recognition, to name a few (Seide et al., 2011; Mikolov et al., 2011a; Collobert et al., 2011; Chen and Manning, 2014).

We experiment with a recurrent neural network language model (RNNLM: (Elman, 1998; Mikolov, 2012)) for our task. We choose this model because it has an internal state that keeps track of previously observed sequences, which is well suited for natural language problems. To train the model, we use stochastic gradient descent combined with back propagation through time. RNNLM is optimised to reduce the error in predicting the following word, based on the current word and its history (represented in a compressed dimension in the size of the hidden layer). Full details of RNNLM can be found in the original papers (Mikolov et al., 2011b; Mikolov, 2012).[11]

We experimented with some of the parameters of RNNLM using BNC-10M and found that most parameters have an intuitive setting. Its performance largely depends on the number of neurons in the hidden layer. Mikolov (2012) introduced a variant of RNNLM that does joint learning with a Maximum Entropy model which learns direct connections of N-gram features. We found that although there are advantages to using the ME model, the benefits disappear as we increase the number of neurons in the hidden layer. We saw optimal performance at 600 neurons, without using the ME model. All our results are based

---

[10]The optimised parameters are: $\alpha = 0.1$, $\beta = 0.001$, $p_\# = 0.5$.

[11]We use Mikolov's implementation of RNNLM for our experiment: http://rnnlm.org/.

Measure	2-gram	3-gram	4-gram	BHMM	LDAHMM	2T	Chunker	RNNLM
LogProb	0.31	0.36	0.38	0.32	0.33	0.35	0.42	0.44
Mean LP	0.28	0.36	0.37	0.28	0.28	0.35	0.45	0.46
Norm LP (Div)	0.34	**0.41**	**0.41**	0.44	0.42	0.49	**0.43**	0.55
Norm LP (Sub)	0.11	0.20	0.22	0.32	0.32	0.44	0.14	0.33
SLOR	**0.35**	**0.41**	**0.41**	**0.46**	**0.44**	**0.50**	0.41	**0.57**

Table 3: Pearson's r of acceptability measure and mean sentence rating for all experimented models in ENWIKI. Boldface indicates the best performing measure.

on the original RNNLM with 600 neurons in the hidden layer, trained on BNC-100M (Table 2 column: "RNNLM").[12] We see that RNNLM performs very well. It outperforms the other models, achieving a correlation of 0.53.

3.7 PCFG Parser (Supervised)

Although we are interested in unsupervised models, for purposes of comparison we experimented with a constituent PCFG parser for our task. We use the Stanford Parser (Klein and Manning, 2003a; Klein and Manning, 2003b), and tested both the unlexicalised and lexicalised PCFG parser with the supplied model. To compute the log probability of test sentences, we experimented with both top-1 and top-1K best parses.

We found that the unlexicalised variant gives better performance, but we saw little difference between using the top-1 and the top-1K best parses for computing log probability. In Table 2 (column: "PCFG"), we report results for the unlexicalised variant based on the top-1 best parse. The supervised PCFG parser performs poorly. This is not surprising, given that the parser is trained on a different domain.[13] Moreover, the log probability scores are not true probabilities, but arbitrary values used for ranking the parse trees.

4 English Wikipedia

For the BNC domain we saw that *SLOR* and *Norm LP (Div)* give the best acceptability measures, and that BHMM, two-tier BHMM and RNNLM are the best performing models. These findings are limited to a particular dataset. To better understand if these observations generalise to another domain, we developed an English Wikipedia dataset (ENWIKI), following the same process described in Lau et al. (2014) to generate test sentences through round-trip machine translation, and to collect annotations via Mechanical Turk.[14] As before, we follow the same procedures described in Section 3 to optimise, train, and test all models (excluding LDA and PCFG). The Pearson correlations with mean AMT annotations are presented in Table 3.

We identify similar trends in ENWIKI: *Norm LP (Div)* and *SLOR* are the best acceptability measures, and we see improvements when we use a richer structure in the language model (two-tier BHMM>BHMM>N-grams). Interestingly, LDAHMM performs much better in this domain (possibly due to increased coherence in the document structure of ENWIKI). RNNLM has the best performance of all models, surpassing two-tier BHMM by a substantial margin. Overall, the correlation values are very similar across the two domains, indicating that the models and the acceptability measures are robust.

5 Comparison with a Supervised System

Although not a focus of this paper, supervised learning can further improve the correlation performance of our models. The acceptability measures can be combined in a supervised context. We experimented with this approach in a support vector regression model (with an RBF kernel). We achieved a correlation performance of 0.64 in BNC and of 0.69 in ENWIKI.[15]

Heilman et al. (2014) propose a system for predicting acceptability. They built a dataset consisting of sentences from essays written by non-native speakers for an ESL test. Acceptability ratings were judged by the authors, and through crowdsourcing (henceforth we refer to this annotated data set as the GUG data set). They applied

---

[12]Other parameter values of RNNLM: number of classes = 550; bptt = 4; bptt-block = 100.

[13]The Stanford English model is trained on the parse tree hand annotated WSJ (section 1–21), Genia, and a few other datasets.

[14]Both the BNC and the English Wikipedia datasets are available at http://www.dcs.kcl.ac.uk/staff/lappin/smog/?page=research.

[15]We use only the unsupervised models, excluding the supervised PCFG parser. The models are trained and tested using 10-fold cross validation.

a 4-category ordinal scale for rating the sentences. To predict sentence acceptability, they employ a linear regression model that draws features from spelling errors, an N-gram model, precision grammar parsers, and the Stanford PCFG parser.

To better understand the performance of our system compared to other acceptability prediction systems, we evaluated our methodology against that of Heilman et al. (2014) on the GUG dataset. We preprocessed the GUG dataset minimally. We removed 15 short sentences that have less than 5 words, lowercased all words, and tokenised the sentences using OpenNLP. This yields 2255 sentences for the training and development subset, and 749 sentences for the test set. Using the output – i.e. the acceptability measures – of our unsupervised models (trained on BNC) as features, we trained an SVR model using GUG training and development subsets to predict acceptability ratings on GUG test sentences. We applied the default SVR parameters, and so it was not necessary to use the development subset separately to optimise the parameters. For evaluation we computed the correlation of the predicted ratings and mean human ratings.

We present a comparison of results in Table 4. We first tested the unsupervised models, with the best correlation of 0.472 produced by the lexical 4-gram model using the *Norm LP (Div)* measure. Combining the models in SVR, we achieve a correlation of 0.603.

Heilman et al. (2014) note that spelling is one of the important features in their regression model, as the dataset often contains spelling mistakes. We borrowed this feature, computed as the proportion of misspelled words, and incorporated into our model. It produced a significant improvement in the correlation (0.636), a performance almost on par with that of Heilman et al. (2014).[16]

Our results demonstrate the robustness and portability of our system in a new domain. Our SVR model requires significantly less supervision than that of Heilman et al. (2014), which relies on precision and constituent parsers. Moreover, our methodology provides a completely unsupervised alternative that requires only raw text for training.

---

[16]We use PyEnchant for spellcheck: http://pythonhosted.org/pyenchant/. Note that we also tried adding the spelling feature to our original BNC derived dataset, but it yielded no improvement in the correlation. This is not surprising, given that it contains few spelling errors.

System	Pearson's r
Heilman et al. (2014)	0.644
Unsupervised Best	0.472
SVR: All Models	0.603
SVR: All Models+Spell	0.636

Table 4: A comparison of results of our system and Heilman et al. (2014) on GUG.

6 Influence of Sentence Length and Lexical Frequency

Our primary motivation in doing this research has been to use acceptability predictions to explore whether acceptability can be represented through probability information. Unlike probability, acceptability is generally not influenced by sentence length or low frequency words.

The acceptability measures we apply normalise sentence length and word frequency. To evaluate their effectiveness, we computed two correlations in the BNC domain: (1) acceptability measure vs. sentence length (Table 5); and (2) acceptability measure vs. sentence minimum word frequency (Table 6).[17]

For comparison we additionally computed the correlation of these factors with human ratings. The correlations are: $+0.13$ with sentence length; and $+0.07$ with minimum word frequency. These observations confirm the view that acceptability is not affected by these two factors.

Table 5 shows that although *LogProb* yields a strong negative correlation with sentence length, *Mean LP*, *Norm LP (Div)* and *SLOR* all produce low correlations. The only exception is *Norm LP (Sub)*, which still has a significant correlation with sentence length.

In Table 6 we see some degree of correlation in *LogProb* with the minimum word frequency, but it is relatively small. In general, *SLOR* is the scoring function that most effectively normalises word frequency, producing low correlation for most models. *Norm LP (Div)* also does very well, for all models except N-grams.

7 Discussion

In principle, the upper bound of the correlation between our models' predicted acceptability values and mean human ratings is 1.0. But no individual human annotator will match mean judgements perfectly. It is more plausible to measure our models'

---

[17]We use BNC-100M for computing word frequency.

Measure	2-gram	3-gram	4-gram	BHMM	LDAHMM	2T	Chunker	RNNLM
LogProb	−0.89	−0.80	−0.84	−0.85	−0.86	−0.86	−0.83	−0.86
Mean LP	−0.16	−0.08	−0.18	+0.03	+0.05	−0.02	−0.01	+0.08
Norm LP (Div)	−0.15	−0.07	−0.17	+0.10	+0.15	±0.00	±0.00	+0.14
Norm LP (Sub)	+0.69	+0.63	+0.54	+0.46	+0.54	+0.11	+0.70	+0.62
SLOR	−0.07	+0.04	−0.03	+0.12	+0.17	+0.01	±0.00	+0.17

Table 5: Pearson's r of acceptability measure and sentence length for all models in BNC. For comparison the correlation with human ratings is +0.13.

Measure	2-gram	3-gram	4-gram	BHMM	LDAHMM	2T	Chunker	RNNLM
LogProb	+0.27	+0.27	+0.27	+0.27	+0.27	+0.27	+0.19	+0.28
Mean LP	+0.30	+0.28	+0.27	+0.29	+0.28	+0.29	+0.08	+0.26
Norm LP (Div)	+0.24	+0.23	+0.21	+0.11	+0.06	+0.12	+0.06	+0.11
Norm LP (Sub)	−0.04	−0.03	−0.03	−0.03	−0.09	+0.05	−0.13	−0.08
SLOR	+0.16	+0.14	+0.12	+0.06	±0.00	+0.10	+0.04	+0.03

Table 6: Pearson's r of acceptability measure and sentence minimum word frequency for all models in BNC. The correlation with the human ratings is +0.07.

rate of success against an estimated level of individual human performance. We do this by mimicking an arbitrary speaker, and testing the correlation of this construct's judgements with the mean scores of the annotators.

We simulate such an individual human judge by randomly selecting a single annotator rating for each sentence, and computing the Pearson correlation between these judgements and the mean ratings for the rest of the annotators (one vs the rest) in our test sets. We ran this experiment 50 times for each test set to reduce sample variation, producing a mean correlation of 0.67 for BNC and 0.74 for ENWIKI. For comparison, the best unsupervised model (RNNLM) achieves a correlation of 0.53 in BNC and 0.57 in ENWIKI (Section 3). The supervised model (SVR) produces a correlation of 0.64 in BNC and 0.69 in ENWIKI (Section 5). Although there is still room for improvement for the unsupervised methodology, it is encouraging to note that the supervised variant predicts acceptability at a level that approaches estimated human performance.

To test the robustness of our methodology across languages, we are currently developing datasets in other languages, based on Wikipedia. Our preliminary results show similar performance to that which we report here for ENWIKI, suggesting that these results hold across languages.

8 Conclusion

We developed a methodology for using unsupervised language models to predict human acceptability judgements. We experimented with a variety of unsupervised models. To map probability to acceptability we proposed a set of acceptability measures to normalise sentence length and lexical frequency. We achieved encouraging results across two datasets constructed through round trip machine translation, and the methodology is highly portable to other domains and languages. This research has potential implications for our understanding of human language acquisition and the way in which linguistic knowledge is represented.

References

D. Blei, A. Ng, and M. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

BNC Consortium. 2007. The British National Corpus, version 3 (BNC XML Edition). Distributed by Oxford University Computing Services on behalf of the BNC Consortium.

D. Chen and C. 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 2014)*, pages 740–750, Doha, Qatar.

Alexander Clark, Gianluca Giorgolo, and Shalom Lappin. 2013. Statistical representation of grammaticality judgements: The limits of n-gram models. In *Proceedings of the ACL Workshop on Cognitive Modelling and Computational Linguistics*, Sofia, Bulgaria.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

J. Elman. 1998. Generalization, simple recurrent networks, and the emergence of structure. In M. Gernsbacher and S. Derry, editors, *Proceedings of the 20th Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum Associates, Mahway, NJ.

Sharon Goldwater and Tom Griffiths. 2007. A fully Bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, pages 744–751, Prague, Czech Republic.

S. Goldwater, T. Griffiths, and M. Johnson. 2009. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112:21–54.

J.T. Goodman. 2001. A bit of progress in language modeling. *Computer Speech & Language*, 15(4):403–434.

Thomas L. Griffiths, Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. 2004. Integrating topics and syntax. In *Advances in Neural Information Processing Systems 17*, pages 537–544. Vancouver, Canada.

Michael Heilman, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel Tetreault. 2014. Predicting grammaticality on an ordinal scale. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), Volume 2: Short Papers*, pages 174–180, Baltimore, Maryland.

Frank Keller. 2001. *Gradience in Grammar: Experimental and Computational Aspects of Degrees of Grammaticality*. Ph.D. thesis, The University of Edinburgh.

D. Klein and C. Manning. 2003a. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, pages 423–430, Sapporo, Japan.

D. Klein and C. Manning. 2003b. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15 (NIPS-03)*, pages 3–10, Whistler, Canada.

J.H. Lau, P. Cook, D. McCarthy, D. Newman, and T. Baldwin. 2012. Word sense induction for novel sense detection. In *Proceedings of the 13th Conference of the EACL (EACL 2012)*, pages 591–601, Avignon, France.

J.H. Lau, A. Clark, and S. Lappin. 2014. Measuring gradience in speakers' grammaticality judgements. In *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, pages 821–826, Quebec City, Canada.

T. Mikolov, A. Deoras, S. Kombrink, L. Burget, and J. Èernocký. 2011a. Empirical evaluation and combination of advanced language modeling techniques. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH 2011)*, pages 605–608, Florence, Italy.

T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Èernocký. 2011b. Rnnlm - recurrent neural network language modeling toolkit. In *IEEE Automatic Speech Recognition and Understanding Workshop*, Hawaii, US.

T. Mikolov. 2012. *Statistical Language Models based on Neural Networks*. Ph.D. thesis, Brno University of Technology.

David Newman, Nagendra Koilada, Jey Han Lau, and Timothy Baldwin. 2012. Bayesian text segmentation for index term identification and keyphrase extraction. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 2077–2092, Mumbai, India.

A. Pauls and D. Klein. 2012. Large-scale syntactic language modeling with treelets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 959–968. Jeju, Korea.

F. Seide, G. Li, and D. Yu. 2011. Conversational speech transcription using context-dependent deep neural networks. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH 2011)*, Florence, Italy.

J. Sprouse and D. Almeida. 2012. Assessing the reliability of textbook data in syntax: Adger's core syntax. *Journal of Linguistics*, 48(3):609–652.

J. Sprouse. 2007. Continuous acceptability, categorical grammaticality, and experimental syntax. *Biolinguistics*, 1:123–134.