Towards a Statistical Model of Grammaticality

Alexander Clark, Gianluca Giorgolo, and Shalom Lappin
firstname.lastname@kcl.ac.uk
Department of Philosophy, King's College London

Abstract

The question of whether it is possible to characterise grammatical knowledge in probabilistic terms is central to determining the relationship of linguistic representation to other cognitive domains. We present a statistical model of grammaticality which maps the probabilities of a statistical model for sentences in parts of the British National Corpus (BNC) into grammaticality scores, using various functions of the parameters of the model. We test this approach with a classifier on test sets containing different levels of syntactic infelicity. With appropriate tuning, the classifiers achieve encouraging levels of accuracy. These experiments suggest that it may be possible to characterise grammaticality judgements in probabilistic terms using an enriched language model.

Keywords: enriched language models, probability distribution, grammaticality judgements, probabilistic syntax

Introduction

The past two decades have seen a lively debate over whether linguistic knowledge is probabilistic or categorically rule-based in nature (see the papers in Bod, Hay, and Jannedy (2003) for some of this discussion). Given the success of probabilistic accounts of learning, representation, and inference across a wide range of cognitive domains, this debate has considerable importance for the way in which knowledge of language is integrated into our general view of human cognition.

On the classical view of syntax developed within linguistic theory over the past sixty years, the grammaticality and the probability of a sentence are entirely distinct properties with no direct relationship. Chomsky (1957) presents the original argument for the irrelevance of probability in determining grammaticality.[1] This argument depends on the inability of a simple word n-gram model to predict a distinction in probability between a syntactically well-formed but unlikely (semantically anomalous) sentence like *Colorless green ideas sleep furiously*, and a word salad like *Furiously sleep ideas green colorless*. Pereira (2000) shows that a smoothed class-based n-gram model trained on a newspaper corpus predicts a significant distinction in probability between the two sentences.

While it is certainly the case that grammaticality cannot be directly reduced to probability, the question of whether there is a significant correlation between the two remains open and interesting. Our general approach is as follows. We train a smoothed class-based trigram model on a filtered subclass of the BNC. We test this model on two corpora. One is divided into original sentences of part of the BNC and their reversed counterparts. The second consists of a subset of original BNC sentences and their permuted variants in which a word in each sentence is randomly exchanged with another word three positions away from it. These distortions constitute syntactic infelicities. The first case involves gross structural ill-formedness similar to the word salad example, while the second introduces subtler, more local mistakes. We score the test corpora using three alternative conditions. Our binary classifiers predict that a string is well-formed (original) or distorted (either reversed or permuted) on the basis of a score derived through normalising its log probability (logprob) value in various different ways. We also test different standard deviations from the distributional norm in setting cut off points for our binary classifiers In our best cases we obtain an accuracy rate of 98.9% for the original-reversal test set, and 79.1% for the original-permutted test set.

These results suggest that by looking at the internal components of a probability distribution and the stages through which it is computed we can identify additional information that may be used to specify significant correlations between probability and grammaticality. This opens up an interesting set of research questions on the relationship between speakers' knowledge of the probability distribution for a language and their grammaticality judgements.

Probability and Grammaticality

As has often been noted, it is not possible to reduce grammaticality directly to probability. First, short ungrammatical sentences generally receive higher probability values than long, complex grammatical sentences containing words with low frequencies. Second, if one specifies a probability value (or even a range of such values) as the minimal threshold for grammaticality, then one is committed to the existence of a finite number of grammatical sentences. The sum of the probabilities of the possible strings of words in a language sum to 1, and so at most $1/\varepsilon$ sentences can have a probability of at least ε.

On the other hand, probabilistic inference does appear to be pervasive throughout all domains of cognition (Chater, Tenenbaum, and Yuille (2006)). Moreover, language models do seem to play a crucial role in speech recognition and sentence processing. Without them we would not be able to identify speech sounds, and meaningful syntactic and semantic structures in noisy environments. Finally, grammaticality appears to track speakers' acceptability judgements, and these are, in many cases, graded. Probability provides a natural basis for generating such a gradient (Crocker and Keller (2006)).

Our starting point is a language model: a statistical model that defines a probability distribution over sentences.

[1] See Fong, Malioutov, Yankama, and Berwick (2013) for a recent discussion of some of the issues involved in identifying grammaticality with probability of occurrence.

We construct a log-linear model, parameterised by some vector of parameters $\Theta = \langle \theta_1, \ldots, \theta_k \rangle$. This framework covers a wide range of different models from n-gram models to PCFGs.[2]

The probabilities defined by this model cannot be used to define a notion of grammaticality for several reasons. First, as the sentences increase in length, the probability of the sentence will always decrease exponentially, for sufficiently long sentences, while we assume that long sentences can be as grammatical as short sentences. Second, one can often substitute a rarer semantically related word for an open class word of the same POS without affecting grammaticality, but the substitution will reduce probability. Figure 1 shows that the log probabilities for sentences that have been reversed or permuted, and are thus generally ungrammatical, overlap completely with the log probabilities of normal sentences (see the next section for details of the experimental protocols). We need to augment our model with an additional component to convert probability into a score that correlates with grammaticality in an interesting way.

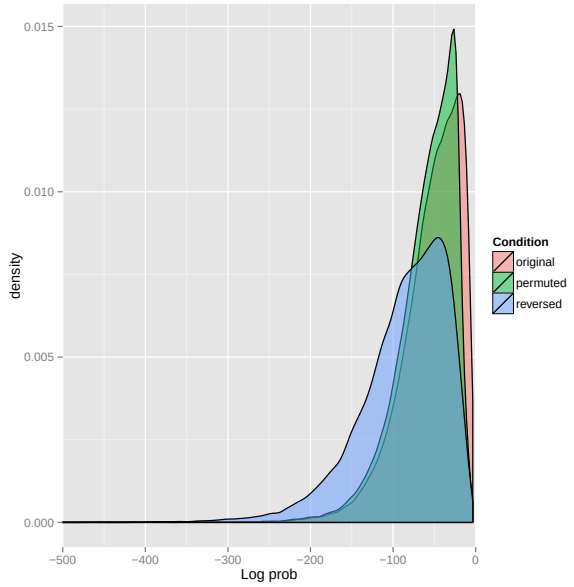

Figure 1: Histograms for the distributions of log probabilities under the three conditions.

We use statistical properties of the parameters of the model. In order to compute the probability of a sentence with respect to a model we do calculations on the parameters. For a log linear model, this gives a linear function of certain indicator variables; a weighted sum. To compute a score that correlates with grammaticality, we consider other functions, such as a weighted *mean*, or a minimum over certain scores.

In a trigram model each parameter will then correspond to the log conditional probability of one word, given the two preceding words: $\theta_{w_i|w_{i-1}w_{i-2}}$. To compute the probability of a sentence we sum the relevant parameters to obtain the log probability. For a sentence $\langle w_1, \ldots, w_n \rangle$ the log probability is

$$\log P_{\text{TRIGRAM}}(\langle w_1, \ldots, w_n \rangle) = \sum_{i=1}^{n} \theta_{w_i|w_{i-1}w_{i-2}}$$

We take the sequence of relevant parameters $\langle \theta_{w_1|w_0w_{-1}}, \ldots, \theta_{w_n|w_{n-1}w_{n-2}} \rangle$, and, rather than summing them, we perform other computations. We consider the average or the minimum of the set of parameters as alternatives for defining values that correspond to grammaticality.

Our most basic score is the mean of this value, the logprob divided by the word length of the sentence:

$$\text{Meanlogprob} = \frac{1}{n} \log P_{\text{TRIGRAM}}(\langle w_1, \ldots, w_n \rangle)$$

This eliminates the dependence of the logprob on the length. Our next score divides the logprob of the original trigram model by the logprob with respect to a unigram model.

$$\text{Normalised} = \frac{\log P_{\text{TRIGRAM}}(\langle w_1, \ldots, w_n \rangle)}{\log P_{\text{UNIGRAM}}(\langle w_1, \ldots, w_n \rangle)}$$

This removes the variation in logprob caused by rare lexical items. Note that if the unigram model is uniform (if we had equal numbers of each word in the training corpus), then the log of the unigram model would be a multiple of the length, and so it would reduce to the previous value.

Our third score uses the observation that a sentence with one grammatical error in it is ungrammatical. In order to measure grammaticality we look at the minimum of some score over the parts of the sentence. We take the minimum of the ratio of the log trigram probability to log unigram probability.

$$\text{Minimum} = \min_i \left[\frac{\log \theta_{w_i|w_{i-1}w_{i-2}}}{\log \theta_{w_i}} \right]$$

None of these measures will produce a score which is in the range $[0, 1]$, though it would be possible to map them into this range. This value will also vary even for grammatical sentences. The scores will be numbers that are distributed in some way. Figure 2 shows the distribution of these scores for the test data. As this score specifies a continuum of values, we are able to accommodate a gradient notion of grammaticality.

Given these three measures we use various standard techniques to see whether new sentences are anomalous or not. For a collection of naturally occurring grammatical sentences we train our models, and then we consider the distribution of these scores. We estimate the mean and standard deviation of the score. We can then judge new sentences as ungrammatical if they are unusually low in score- more than a few standard deviations away from the mean.

Pauls and Klein (2012) apply a related approach to another problem. They use scores based on the logprob values of a language model to discriminate between grammatical and ungrammatical sentences in order to improve the performance of natural language processing systems.

[2]We use smoothing techniques that, in general, can take the model outside the class of log-linear models, but we pass over this technical detail here.

Experiments and Results

For our experiments, we use the standard *n-gram* language model, which is an instance of a Markov model for sequences. To estimate the probability of a sequence of words $w_1 \dots w_k$ we use the chain rule of probability, as in (1).

$$P(w_1 \dots w_k) = P(w_1)P(w_2|w_1) \dots P(w_k|w_1 \dots w_{k-1}) \quad (1)$$

The problem with this approach is that we have to estimate the conditional probability of an extremely large number of possible subsequences. Therefore a common method is to reduce the conditional dependencies to a smaller predefined sequence of a given length n, the so called *order* of a model. Using this assumption we approximate the components in (1) using (2).

$$P(w_i|w_1 \dots w_{i-1}) \approx P(w_i|w_{i-n+1} \dots w_{i-1}) \quad (2)$$

The probability assigned to a sequence of words is given by the product in (3)

$$P(w_1 \dots w_k) \approx \prod_{i=1}^{k} P(w_i|w_{i-n+1} \dots w_{i-1}) \quad (3)$$

A common choice for n, that we adopt for our experiments, is three (trigrams).

The standard strategy to estimate the probability of each n-gram is *maximum likelihood estimation* (MLE), which counts the number of times the n-gram appears in a training corpus, and normalizes the count by the sum of the counts of all n-grams that share the same initial subsequence:

$$P(w_i|w_{i-n+1} \dots w_{w-1}) = \frac{C(w_{i-n+1} \dots w_i)}{\sum_w C(w_{i-n+1} \dots w)} \quad (4)$$

To avoid assigning 0 probability to unseen n-grams (a common case, given the huge number of possible n-grams) we use *smoothing* or *discounting*, which transfers a small portion of probability mass from seen n-grams to unseen ones. A large number of smoothing techniques have been proposed in the literature (see Chen and Goodman (1999) for a thorough overview). In our experiments we use a form of interpolated smoothing known as Interpolated Kneser-Ney (Goodman (2001)), which has been shown to give consistently good results with different types of metrics.

To reduce the search space of our language model we also employ *clustering*, which groups together words that occur in similar contexts. In this way we can better estimate the probability of a word following a certain sequence, given the observations we have made of similar words in the same context. Brown, deSouza, Mercer, Pietra, and Lai (1992) introduced the standard technique for using clustering in language models. The general form of a cluster-based language model is given in equation (5), where C_i is the cluster to which word w_i is assigned to.

$$P(w_i|w_{i-n+1} \dots w_{i-1}) = P(w_i|C_i)P(C_i|C_{i-n+1} \dots C_{i-1}) \quad (5)$$

The probability $P(w_i|C_i)$ is given by the count of occurrences of w_i divided by the count of occurrences of C_i, while the other factor of the product can be estimated with a smoothed model like Interpolated Kneser-Ney. Brown et al. (1992) describes a technique for generating the optimal clustering in a corpus, given a parametrically specified number of classes.

We implemented our own procedures for the training and the assessment of n-gram language models, using Interpolated Kneser-Ney as the smoothing technique. For clustering we applied the improved version of Brown et al. (1992)'s algorithm described in Liang (2005).

Both the training of the language models and the measurement of their performance in the given tasks are performed on portions of the BNC. The BNC is a heterogenous collection of linguistic data. To obtain a more consistent sample of English we first restricted the available texts by excluding transcriptions of spoken language, poetic texts and technical/scientific material. The corpus used for training and the one used for testing were generated from this subset of the BNC by randomly selecting 600k sentences for training, and 60k for testing. This gave us a training corpus of slightly less than 13 million words, and a testing corpus of approximately 1.3 million words.

To avoid the problem of unknown words in testing, we reconstructed both the training and the testing corpus. We substituted, in both the training and the test corpus, the POS tag for each word which appears less than five times in the training corpus. This insures that the test corpus vocabulary is a subset of the training corpus vocabulary.

Three different types of test corpus (conditions) were generated. The *original* condition is left intact, and we assume that it contains only grammatical sentences. The *permuted* condition is generated from the original by randomly swapping two words, separated by two intervening words, in each sentence. The sentences in this corpus are taken to be less grammatical than those in the original condition. Finally, the third test corpus was produced by reversing the order of the words in the original sentences. This *reversed* condition is considered to be the most syntactically distorted of the three.

We used a simple binary classifier to measure the performance of our language model in predicting the grammaticality of a sentence. After calculating the three scores (Mean log prob, Normalised, and Minimum) in all three conditions we designed two different binary classifiers that assign a label to every sentence in each condition. The first classifier is a simple threshold set to different values for the mean and the standard deviation of the distributions of the alternative normalised scores for the original condition. For each binary comparison the classifier assigns a label to the sentence z using the following rule:

$$c_1(z) = \begin{cases} \text{original} & \text{if } score(z) \geq m - S \cdot s \\ \text{other} & \text{otherwise} \end{cases} \quad (6)$$

where m is the mean for the score in the original condition, s is the standard deviation and S is a factor by which we move

the threshold away from the mean. The principle of this classifier is that the normalised logprob scores for ungrammatical sentences will be lower than those for grammatical ones, making it possible to distinguish between the two conditions. We adapted this procedure to distinguish between local ungrammaticality (permutation), and more global ungrammaticality (reversed cases).

The second classifier is a simple linear classifier constructed on the basis of the first one. It combines the information from two different scores. This second classifier uses the following general rule:

$$c_2(z) = \begin{cases} \text{original} & \text{if } score_2(z) \geq -score_1(z) + t_1 + t_2 \\ \text{other} & \text{otherwise} \end{cases} \quad (7)$$

where $score_1(z)$ is the first of the scores assigned to the sentence, $score_2(z)$ is the second one, t_1 is the best performing threshold for this specific comparison as found in the case of the first type of classifier for the first score, and t_2 is the same kind of threshold for the second score. We simply check whether the two scores are above or below the bisector of the second and the fourth quadrant in the space formed by the two scores, and translated by the best thresholds for the same two scores. The intuition here is, again, that grammatical sentences will have consistently better scores than ungrammatical ones.

We performed experiments using both the standard and the cluster-based language models. For the standard case we trained models using words and part-of-speech tags as tokens. In what follows we report only the results for the cluster-based experiments, as these achieved better accuracy. We used 250 clusters. The language model was trained on the training corpus, and the three scores are computed for the sentences in each condition (original, permuted and reversed). Figure 2 summarises the distributions of the three scores for each condition of the cluster-based language model. It is clear that all scores are reasonably good at distinguishing between the original and the reversed conditions, given the small overlap between the distributions. As expected, the overlap between the original and permuted conditions is much higher. It is also interesting to note that the while in the case of Mean log prob and Normalised score the distributions for all the conditions are roughly normal (with some degree of skewing), the Minimum score gives a more irregular distribution, at least for the ungrammatical cases.

On the basis of these distributions we created the first type of classifier. The results for the two comparisons we performed (original/permuted and original/reversed) are summarised in figure 3. The graphs show the accuracy for each score obtained by varying the S parameter as described in (6). In our experiments we let S vary in the interval $[0, 2.75]$, using a step interval of 0.25.

In the case of the original/permuted comparison we obtained the best accuracy (77.3%) by using the Normalised score and setting the threshold at 0.75 standard deviations to the left of the mean. However the Minimum score seems to perform better in general for this comparison, obtaining a maximum accuracy of 77.1%.

Table 1: Linear classifier accuracy

Accuracy	permuted	reversed
Mean log prob + Normalised	71.2	97.9
Mean log prob + Minimum	77.1	97.2
Normalised + Minimum	**79.1**	98.1
Threshold classifier baseline	77.3	**98.9**

Not surprisingly, all three scores perform very well when distinguishing between the original and the reversed version of the sentence, with accuracies above 95%. The sharp drop in accuracy in the case of the Minimum score that we observe when setting the S parameter to 2.75 is due to the spike we have in the case of the reversed condition (see rightmost graph in figure 2).

Table 1 reports the accuracy for the linear classifier that combines the results of two threshold classifiers (with the best single classifier scores listed in the bottom row as a baseline comparison). Despite the simplicity of this linear classifier, we observe an improvement in the original/permuted comparison.

Error analysis

It is interesting to analyse the cases where our classifiers fail. We looked at the cases that form the tails of the distributions for the Normalised threshold, as it is this score that gives the best general level of classifier accuracy.

The following ten sentences receive the lowest Normalised score according to our language model for the original condition: *interview · Swims · / · contracts · then · TELEPHONE · 75% · Hotel deal · mimic each item across · Ian ! 90%* These cases are very marginal English sentences. Their presence in the corpus may well be due to transcription error in the BNC, or to the idiosyncratic nature of the text from which they are extracted. However, other cases of false ungrammatical sentences include perfectly acceptable sentences like the following: *Amnesty has been given Greetings Magazine's "Best Charity Card of the Year" award .*

For permuted sentences, when we analyse the tail of the distribution, we encounter many cases where the permutation produces the same sentence as the original, because the permuted words are identical. In other cases the permutation generates semantically odd, but otherwise well-formed sentences, as in *It should be a match of a humdinger*. These are the ten permuted sentences that receive the highest Normalised scores (and they are therefore mislabelled as original): *He glanced round the bar from the door. · He said that he had not been informed of the dissolution of the National Assembly on Jan. 4. · There 's something I hear you to want. · Sometimes , of course , it does not work. · Don't know, I worry why. · I assure you I'm not. · It should be a match of a humdinger. · She put her hand to her brow. · "Yes , I understand ," said Drew quietly. · But there was nothing there.*

Finally in the case of reversed sentences, we observe that sentences that are assigned extremely high Normalised scores tend to be proper names. Due to their low frequency in the training corpora, proper names are most likely to be replaced by their POS tag in the training and testing phase. Therefore, the language model cannot distinguish the original and the reversed versions of the sequence, given that they appear identical. Again we report here the ten reversed sentences that receive the highest Normalised score. *Terrazze Alle · seven - eight - nine · 2 TN. WOKINGHAM , 3 TN. FARNHAM · Debts : MAIDSTONE · Gloucester / BROCKWORTH · FLAUBERT MME · VALLI FRANKIE · PATEL GARGY · BATTERSEA HORSMAN UDO · REUNITE / JAFFE LUCKY*

Discussion and Conclusions

Clark and Lappin (2011) propose an outline for a stochastic model of indirect negative evidence. In this outline a function maps the probability value of a string, and a set of properties of the string and of the probability distribution over strings of the language, to a threshold value that gives the minimum frequency with which the string must occur in the primary linguistic data in order to be well-formed. The threshold specifies the normalised minimal expectancy of occurrence for a sentence of a certain type (length, lexical class sequence, etc.). This model provides a language learner with a procedure for querying the data to which he/she is exposed in order to determine the extent to which the absence of a string in the data indicates its ungrammaticality.

Here we effectively invert this strategy. We identify a set of structural properties of a string together with parameters for the distribution of logprob-derived scores, in order to define a grammaticality threshold, which we use to classify strings as grammatical or ill-formed. This model offers a stochastic characterisation of grammaticality without reducing grammaticality to probability. It represents a core element of what speakers know about the syntax of their language through a set of parameters in a model whose values correspond to properties of the modified probability distributions that the model generates.

We are not, of course, suggesting that enriched n-gram models are adequate to express the full content of speakers' syntactic knowledge. However, the fact that simple models of the sort that we have used are able to achieve a relatively high degree of accuracy on wide coverage, domain general grammaticality classification tasks suggests that there is an interesting correlation between properties of the probability distribution over the sentences of a language and a speaker's grammaticality judgements.

Should the correlation prove robust it suggests that grammatical knowledge is, to a significant extent, determined by the stochastic patterns of the primary linguistic data to which speakers are exposed. This result will have significant consequences for both the representation of syntactic competence and the nature of the language acquisition process.

In current work we are exploring this correlation further with more sophisticated language models, different distributional parameters and stochastic classifiers, and test data that includes realistic syntactic infelicities. We are evaluating these models against native speakers' acceptability judgements.

Acknowledgments

The research described in this paper was done in the framework of the Statistical Models of Grammaticality (SMOG) project at King's College London, funded by grant ES/J022969/1 from the Economic and Social Research Council of the UK. We are grateful to Rens Bod, Stephen Clark, Aarne Ranta, Khalil Sima'an, and Jelle Zuidema. for helpful comments on an earlier draft of this paper. We also thank our PhD students, Jekaterina Denissova and Monika Podsiadlo, for useful discussion of the experimental design of this work, and for logistical support.

References

Bod, R., Hay, J., & Jannedy, S. (2003). *Probabilistic linguistics*. MIT Press.

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, *18*, 467–479.

Chater, N., Tenenbaum, J., & Yuille, A. (2006). Probabilistic models of cognition: Conceptual foundations. *Trends in Cognitive Sciences*, *10*(7), 287–291.

Chen, S., & Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, *13*(4), 359–393.

Chomsky, N. (1957). *Syntactic structures*. The Hague: Mouton.

Clark, A., & Lappin, S. (2011). *Linguistic nativism and the poverty of the stimulus*. Malden, MA: Wiley-Blackwell.

Crocker, M., & Keller, F. (2006). Probabilistic grammars as models of gradience in language processing. In G. Fanselow, C. Féry, M. Schlesewsky, & R. Vogel (Eds.), *Gradience in grammar: Generative perspectives* (pp. 227–245). Oxford University Press.

Fong, S., Malioutov, I., Yankama, B., & Berwick, R. (2013). Treebank parsing and knowledge of language. In A. Villavicencio, T. Poibeau, A. Korhonen, & A. Alishahi (Eds.), *Cognitive aspects of computational language acquisition* (p. 133-172). Springer Berlin Heidelberg.

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

Liang, P. (2005). *Semi-supervised learning for natural language processing*. Unpublished master's thesis, Department of Electrical Engineering and Computer Science, MIT.

Pauls, A., & Klein, D. (2012). Large-scale syntactic language modeling with treelets. In *Proceedings of the 50th annual meeting of the association for computational linguistics* (pp. 959–968). Jeju, Korea.

Pereira, F. (2000). Formal grammar and information theory: together again? *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, *358*(1769), 1239–1253.

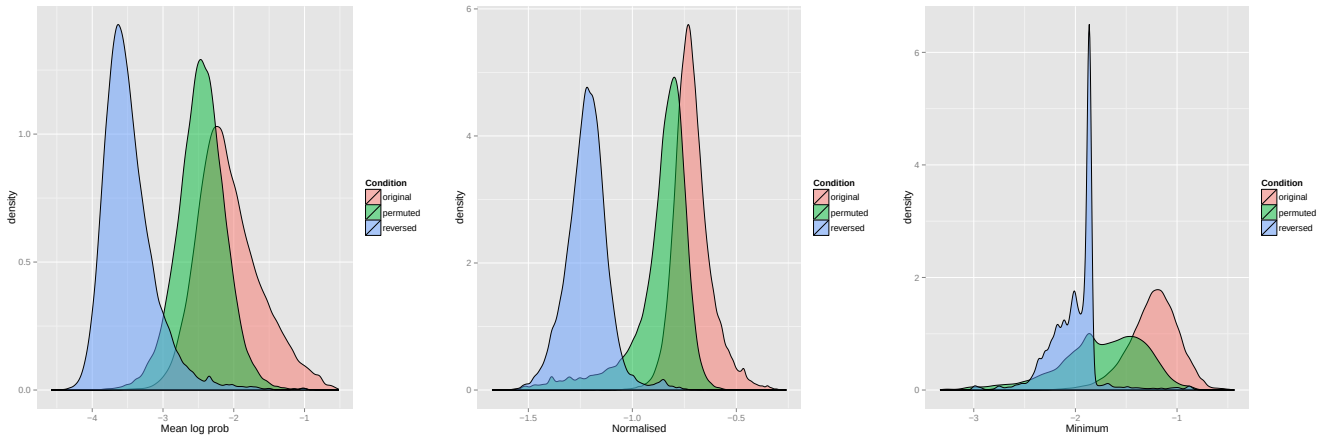

Figure 2: Histograms for the distributions of sentence scores. Each graph shows the distribution of a single score for the three conditions. The x-axis represents the value of the score and the y-axis gives a measure of the frequency with which the score is represented in the data. On the left are the scores given by taking the mean (equivalently normalising by length). In the middle are the scores given by normalising with the unigram probability. On the right are the scores using the minimum condition. These scores still overlap significantly, but much less so than the raw logprobs as shown in Figure 1.

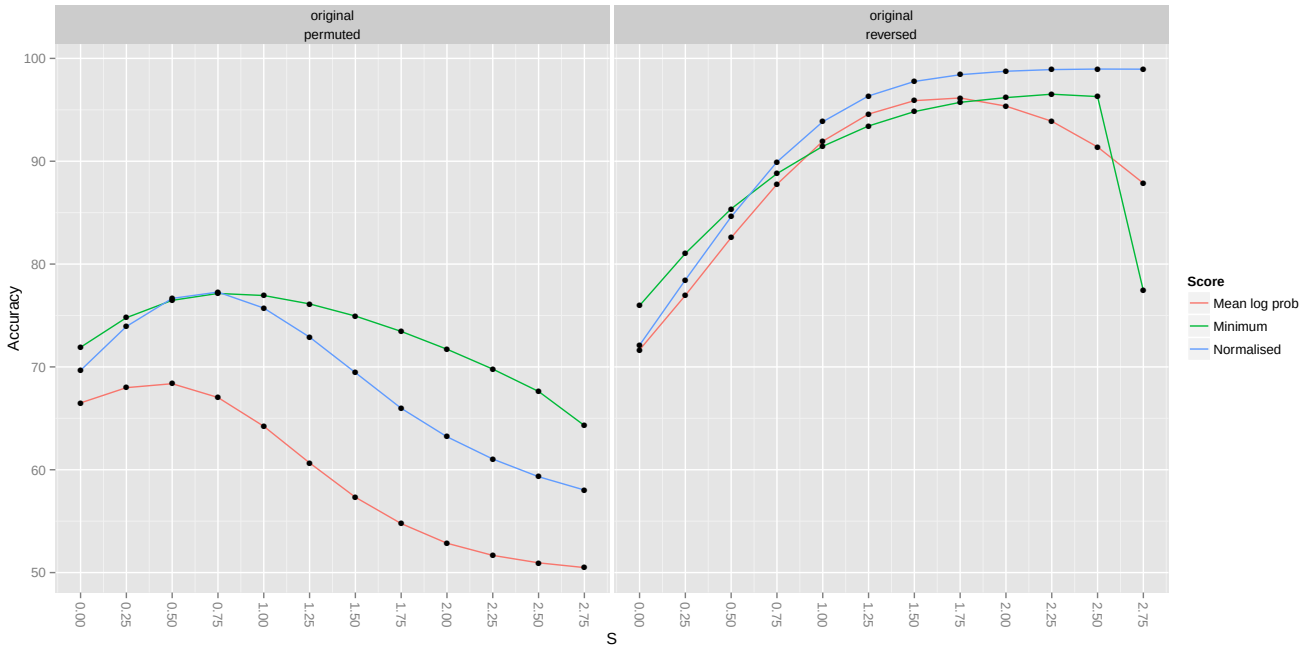

Figure 3: Results for the threshold classifier. The two graphs show two comparisons: original/permuted and original/reversed. The x-axis represents the different values that control the distance from the mean of the threshold, while the y-axis shows the accuracy expressed in percentages.