Understanding Word Order with an Information-Theoretic Approach

Yukti Makhija (2019BB10067)
Prof. Sumeet Agarwal and Prof. Mausam

Table of contents

Abstract

Human language processing research involves identifying features that play a role in word order prediction and sentence comprehension. These features are derived using probability, information theory, and cognitive sciences. Dependency length and n-gram log-likelihood are two classic examples. This project aims to create new parameters using Pairwise Mutual Information (PMI) and the Principle of Information Locality. In addition to that, we compare the performance of these to pre-existing memory-based and surprisal-based features on word-order prediction tasks in English. Further, we find correlations between these features by testing multiple combinations of old and new features and identifying the strongest set of features.

Background

Psycholinguistics aims at searching for patterns in processing difficulties of sentences. Certain utterances are easier to comprehend than others, and different components of a sentence require varying levels of processing difficulty. Theories that justify and predict processing difficulty are broadly classified into two categories: memory-based and expectation or surprisal-based approaches.
This section briefly explains the various influential theories that form the backbone of our project. These concepts are imperative in understanding the relationship between the feature value and expected word order.

Dependency Locality Theory

DLT predicts the human tendency to prefer shorter dependencies.
According to the DLT, the syntactic complexity of a sentence can be divided into two parts. These are the storage cost and integration cost.
Storage cost is the cost of maintaining the requirements of previous words or the syntactic predictions in memory.
Integration cost is based on the fact that activation of words decays as they recede in time (which makes integration difficult). So, the integration cost for a word increases with the distance to previous words to which it is connected. This distance is measured in terms of the nature and number of intervening discourse referents.

Surprisal

Surprisal represents the difficulty of comprehension. Given the preceding context c, incremental processing difficulty for a word w is proportional to its surprisal given the context.

$$Difficulty(w|c) \propto -\log p(w|c)$$

It is an information-theoretic characterisation and is expressed in bits, with its value increasing with increasing processing load. We can also think of it as a measure of the predictability of a word, as more predictable words will have lower surprisal values than less predictable words.

Consequently, less processing load and more predictable words make processing time faster.
We estimate surprisal using either simple lexical models (e.g. n-gram models) or syntax-based PCFGs (Probabilistic Context-Free Grammars).

Information-Theoretic features

1. **PCFG log-likelihood**: This is estimated using a PCFG parser with state of the art parsing performance. The likelihood of a sentence is calculated as the sum of probabilities of all parse trees for that sentence. The negative of the log-likelihood gives cumulative surprisal.
2. **n-gram log probability**: This is estimated using a 5-gram language model from the English Gigaword corpus (popular in many mainstream NLP applications). For this, too, likelihood is found by summing the log probabilities of all words in a sentence. The total n-gram surprisal is also the negative of the log-likelihood.

Memory-based features

1. **Dependency length in terms of discourse referents**
2. **Weighted embedding depth:** The more likely a parse hypothesis is, the more cognitive resources should be allocated to it. This is measured by the weighted embedding depth. Based on its embedding depth and its parse likelihood, a lexical item at the kth position is given a complexity score, which is summed over the set of all active parse trees (T_k).

$$weighted\ embedding\ depth = \sum_{t \in T_k} P_t(w_k | w_0 \ldots w_{k-1}) \cdot depth_t(w_k)$$

3. **Lexical (1-best) embedding depth**: This is the sum of embedding depths from mist probable final parse T. Parsing may occur serially (one hypothesis at a time), and so the best parse could be the only one that has significant influence during the sentence generation.

$$1 - best\ embedding\ depth = \sum_{w \in T} depth_T(w)$$

Methods and Results

In the previous semester, Ms Rashi did her final year thesis project on the same topic. She took inspiration from the work done by Futrell on Head Dependency Mutual Information (HDMI) and quantifying the strength of dependency using PMI value. She had defined four PMI based features and performed further experimentation to prove that the Information Locality Hypothesis holds true. I started out by replicating her results to get a better understanding of the concepts involved. In this process, I found some oversights and, after correcting those, certain results changed.

Data

All the experimentation was performed on famous English corpora: Brown Corpus and Wall Street Journal Corpus. Both of these corpora are syntactically annotated. The Brown Corpus contains sentences written in American English belonging to 15 different genres, and the WSJ Corpus contains newswire text.

We take a 'reference' sentence from the two corpora and change the ordering of dependents of the root verb in the dependency tree of this sentence to create 'variant' sentences. So the reference and variant sentences have the same meaning but with different orderings. These reference-variant pairs make up our data.

The details of the dataset we used and the techniques used for creating it can be seen in Rajkumar et al., 2016.

PMI-based Feature Definitions and Calculations

Feature 1: PMI between verbal root and the nearest dependent

$$HDMI = \log \frac{p(h,d)}{p(h)p(d)}$$

where,
h is the verbal root (head)
d is the nearest dependent of the root verb

Both the corpora contain dependency trees for all the sentences. Using the dependency tree, we located the root verbs for a given sentence and identified its nearest dependent. The POS Tag of the dependent was determined, and the PMI value for that verb and dependent pair was used.

The Information Locality Hypothesis (ILH) states that the PMI value should be higher if the dependency length is low. This implies an inverse relationship between the two. This feature is calculated for both the reference and variant sentences. Since the word order of reference sentence is favoured over that of the variant's we expect PMI_{ref} to have a greater value than PMI_{var}.

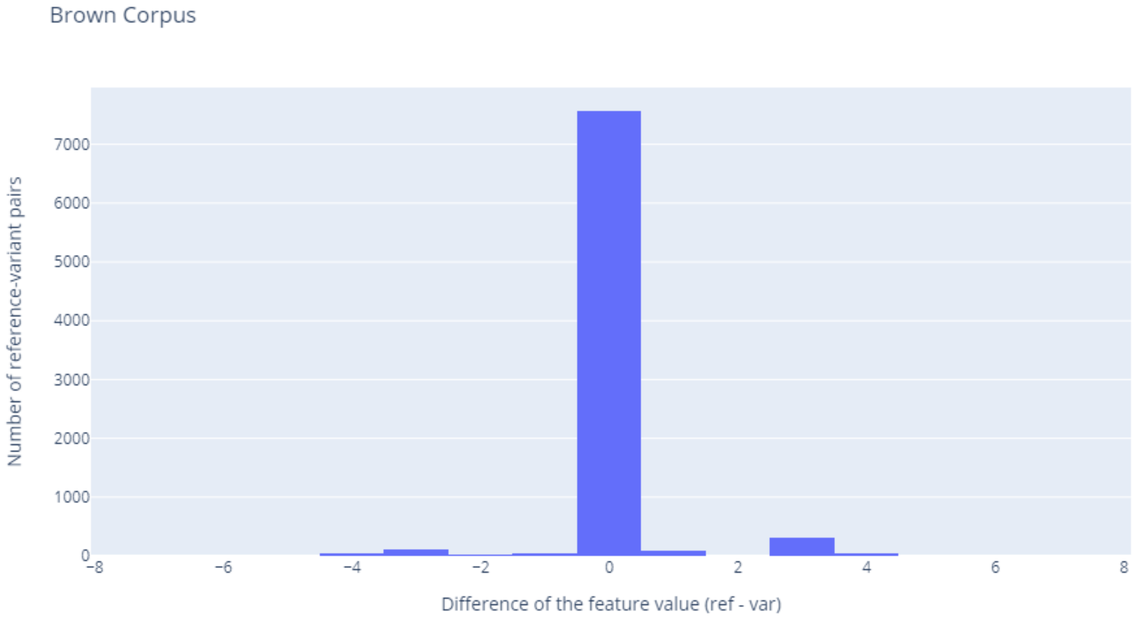
Note: In the previous work, this feature was defined as the PMI value between root and its closest dependent, which is different for the reference and variant sentences. But there were some flaws in the algorithm used. Hence, we simplified this definition to consider only the nearest dependent, irrespective of whether it is the same reference and variant sentences. This modification ensures that the analysis is uniform and helps in avoiding situations where the first nearest dependent of the reference is compared to the second nearest dependent of the variant.

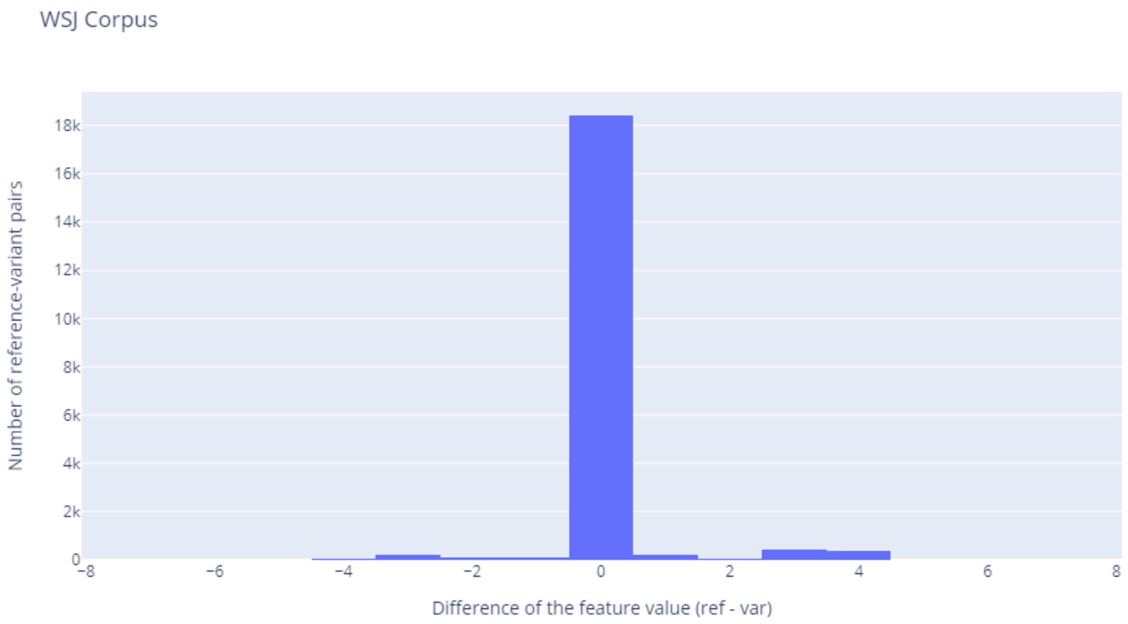
I have summarized the obtained results in the following table:

Corpus	Total number of reference and variant pairs	$PMI_{ref} > PMI_{var}$	$PMI_{ref} = PMI_{var}$	$PMI_{ref} < PMI_{var}$
Brown	8264	666 (8.06%)	7216 (87.32%)	382 (4.62%)
WSJ	19990	1356 (6.78%)	18007 (90.08%)	628 (3.14%)

In the majority of the cases, the nearest dependent was the same for the reference and variant sentences. Therefore this feature can't differentiate between them. For the remaining sentences, the trend predicted by ILH is followed as $PMI_{ref} > PMI_{var}$ is observed for a larger number of sentences. The results are very similar for both corpora.

The following histograms assist in visualizing the distribution of the difference between the PMI value of reference and variant sentences.





Feature 2: PMI between verbal root and post-verbal adjacent word

$$HDMI = \ log \frac{p(h, w)}{p(h)p(w)}$$

where,
h is the verbal root (head)
w is the word adjacent to the root verb

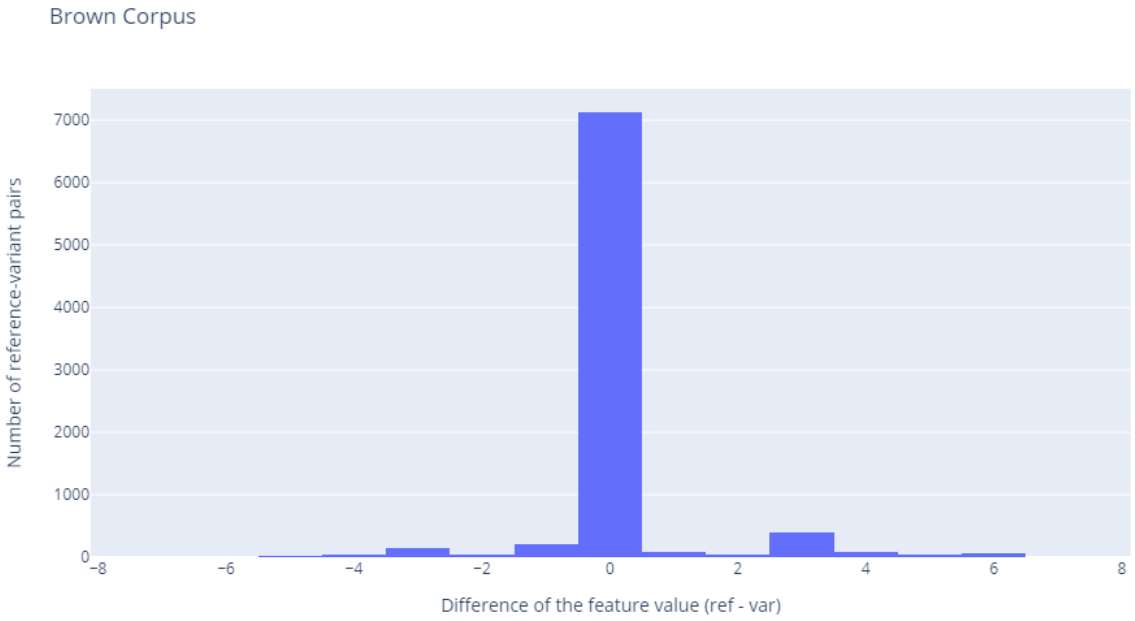
We used the dependency trees to find the root verb and extract the POS tag of the word following it. We found the feature value using the POS Tags for head and dependent. In English, sentence reordering takes place after the head root. Hence we expect to see the hypothesized trend post-verbally. This was the reason for choosing post-verbal adjacent words.

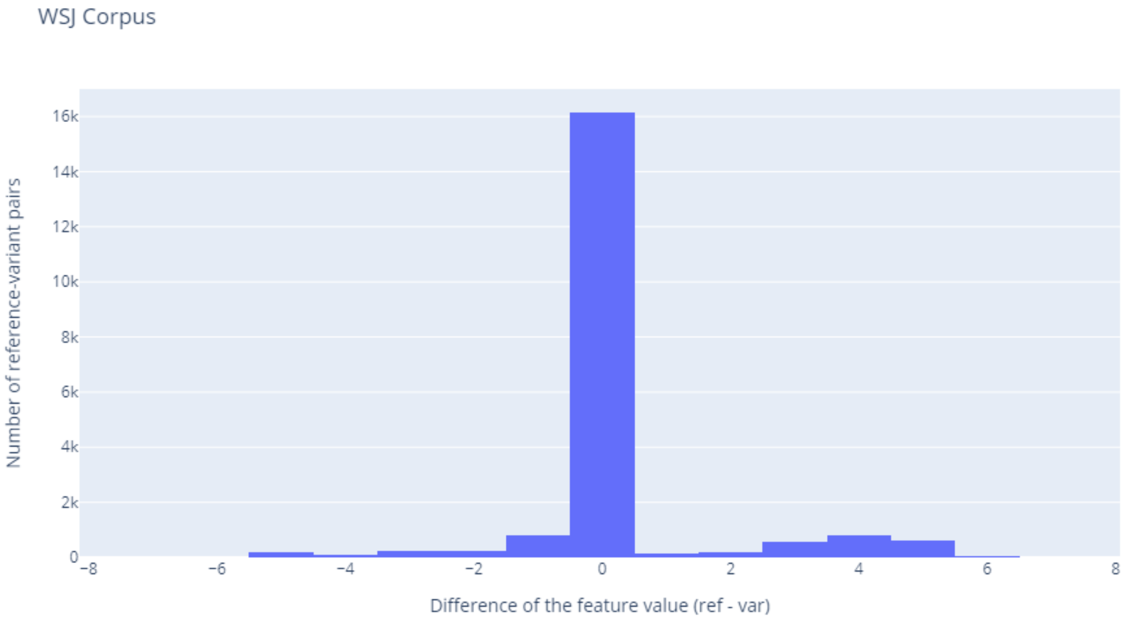
Applying ILH to this case predicts PMI_{ref} to have a larger value than PMI_{var}. I have summarized the obtained results in the following table:

Corpus	Total number of reference and variant pairs	$PMI_{ref} > PMI_{var}$	$PMI_{ref} = PMI_{var}$	$PMI_{ref} < PMI_{var}$
Brown	8264	881 (10.66%)	6803 (82.32%)	580 (7.02%)
WSJ	19990	2981 (14.91%)	13655 (68.31%)	3355 (16.78%)

The results for the Brown Corpus are in accordance with ILH, with a higher percentage of pairs having PMI_{ref} greater than PMI_{var}. But strangely, this pattern is not observed for WSJ Corpus, where the number of sentence pairs having PMI_{ref} lesser than PMI_{var} is more. For a large number of sentence pairs, the post-verbal neighbouring word is the same.

The following histograms assist in visualising the distribution of the difference between the PMI value of reference and variant sentences.





Feature 3: Spearman's correlation coefficient between the dependency length and the PMI value between the verbal root and all dependents

$$\rho(PMI, \frac{1}{dlg}) = 1 - \frac{6\sum(rank(\frac{1}{dlg_n}) - rank(PMI_n))^2}{N(N^2-1)}$$

where

ϱ is the Spearman's Correlation Coefficient

N is the total number of root-dependent pairs in a sentence

PMI_n is the PMI between n^{th} root-dependent pair in a sentence

dlg_n is the dependency length for n^{th} root-dependent pair

Note: ϱ is calculated separately for pre and post head-dependent pairs

Once the root verb of the sentence has been located, we store the POS Tags of all post-verbal dependents along with the dependency length from this root. We find the corresponding PMI value for each root-dependent pair. The next step involved ranking the PMI values and dependency length. These ranks were later used to compute Spearman's correlation coefficient, which helped define the nature of the relation between these variables. ILH states that distance and PMI values share an inverse association, so to get a positive value of the coefficient, we rank PMI Values in decreasing order of value and length in increasing order.

As per ILH, the value of Spearman's correlation coefficient should be positive. The word ordering of the reference sentence is preferred over that of the variant, and hence we expect ILH to be strictly followed for reference. This indicates a stronger correlation would exist between PMI value and distance in the case of the reference sentence. Therefore, PMI_{ref} should be considerably more than PMI_{var}.

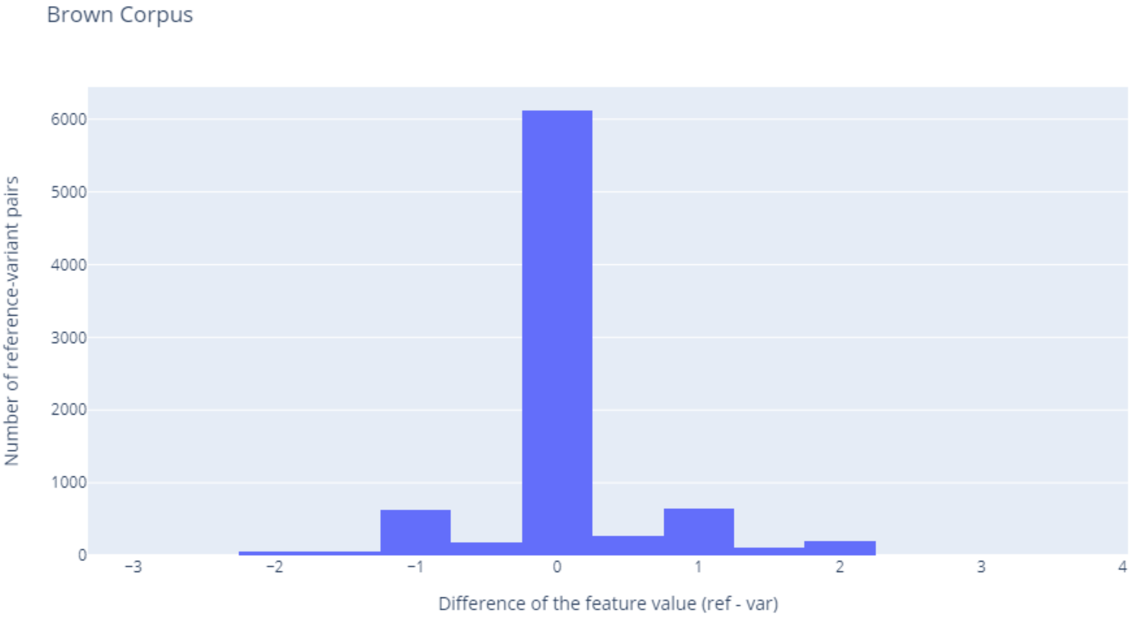
Note: In the previous work, the ranking was done in increasing order for both parameters, and we also spotted a sign inversion error. These have now been rectified.

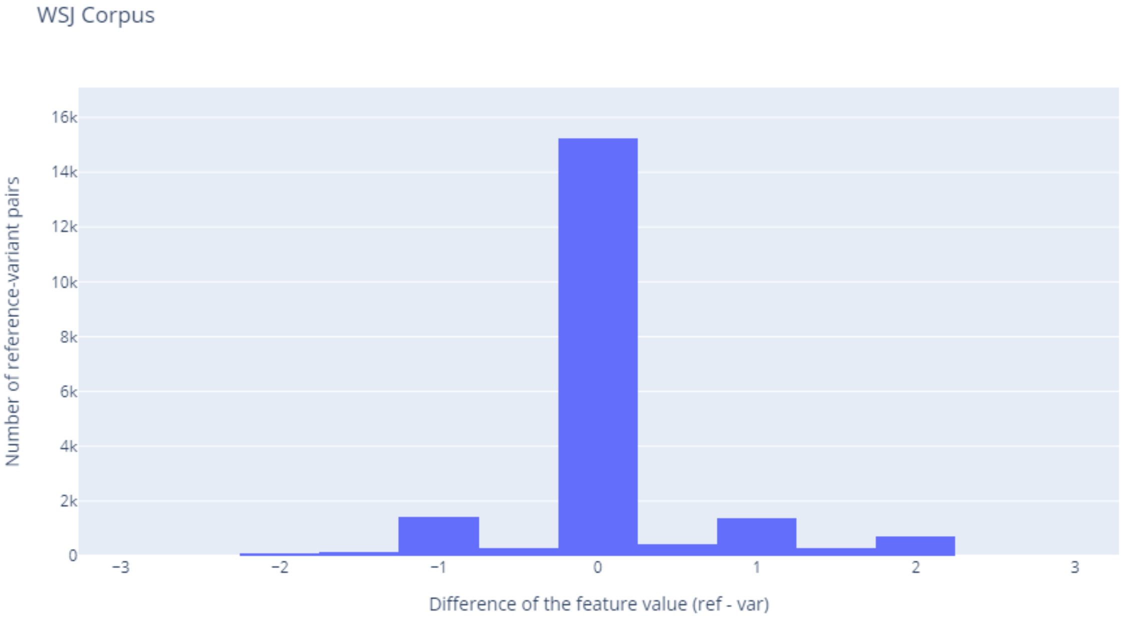
I have summarised the obtained results in the following table:

Corpus	Total number of reference and variant pairs	$PMI_{ref} > PMI_{var}$	$PMI_{ref} = PMI_{var}$	$PMI_{ref} < PMI_{var}$
Brown	8264	1388 (16.8%)	5833 (70.58%)	1043 (12.62%)
WSJ	19990	2978 (14.89%)	14830 (74.19%)	2183 (10.92%)

The results obtained were in line with the prediction made using ILH. Interestingly, we came across two types of cases where Spearman's correlation coefficient was not defined, the first where only one post-verbal dependent was present and the second where no post-verbal dependents existed. In both these, the value should be zero, but we assume it to be zero for further analysis. This assumption had also been made in the earlier work.

The following histograms assist in visualising the distribution of the difference between the PMI value of reference and variant sentences.





Feature 4: Spearman’s correlation coefficient between the distance and the PMI value between the verbal root and all words (post-verbal)

$$\rho(PMI, \frac{1}{lg}) = 1 - \frac{6\sum(rank(\frac{1}{lg_n}) - rank(PMI_n))^2}{N(N^2-1)}$$

where

ϱ is the Spearman's Correlation Coefficient

N is the total number of root-word pairs in a sentence

PMI_n is the PMI between nth root-word pair in a sentence

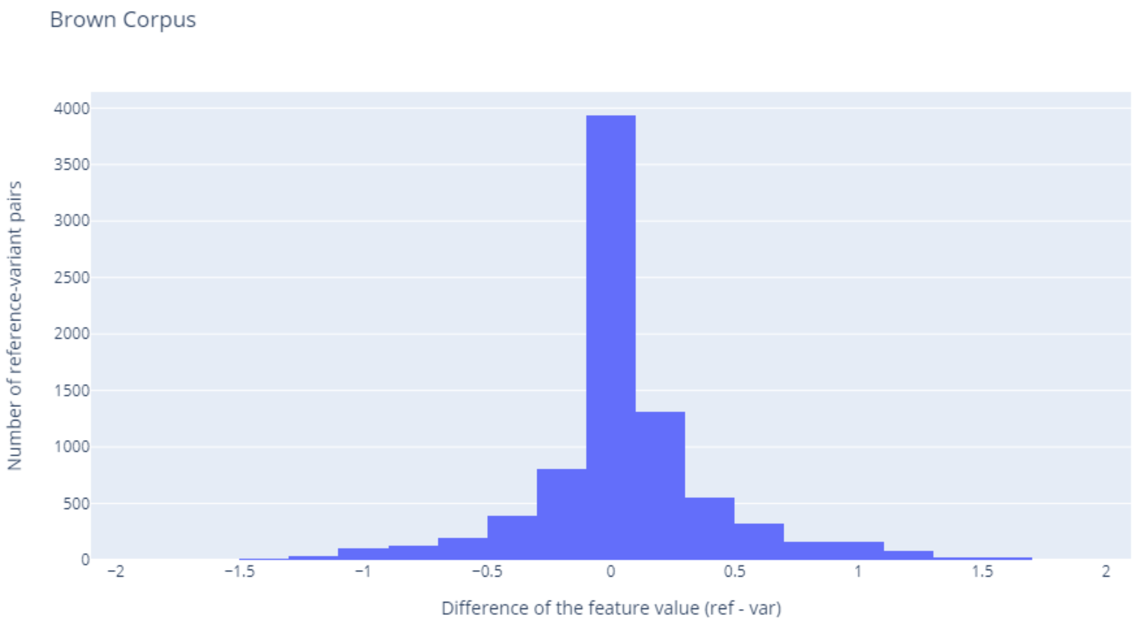
lg_n is the distance between root verb and word for nth pair

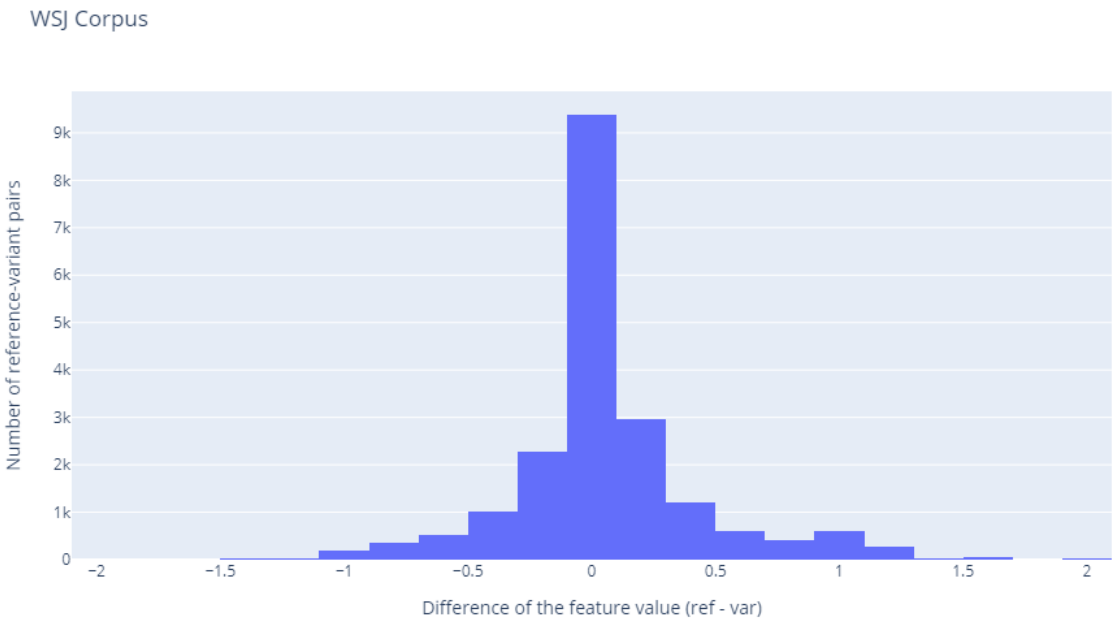
Note: ϱ is calculated separately for pre and post head-dependent pairs

The steps for calculation are similar to feature 3, but instead of taking only the dependents, we consider all the post-verbal words. We store the POS Tag and linear distance between all post-verbal words and the roots. Subsequently, we ranked the PMI values in increasing order of value and distance in decreasing order. Applying ILH to predict the trend
I have summarised the obtained results in the following table:

Corpus	Total number of reference and variant pairs	$PMI_{ref} > PMI_{var}$	$PMI_{ref} = PMI_{var}$	$PMI_{ref} < PMI_{var}$
Brown	8264	4174 (50.51%)	1220 (14.76%)	2870 (34.73%)
WSJ	19990	9082 (45.43%)	4249 (21.26%)	6660 (33.32%)

The following histograms assist in visualising the distribution of the difference between the PMI value of reference and variant sentences.





Experiments with Logistic Regression Models

After calculating the four PMI-based features, we moved on to training logistics regression models by combining them with other surprisal and memory features. This is a binary classification setting where the models are trained to distinguish between reference and variant sentences, thereby predicting the preferred word ordering. We compared different metrics of these models against three types of baselines: first containing only surprisal-based features, second with memory-based features, and third containing both surprisal and memory-based elements. The primary evaluation metric used for model comparison is the accuracy since it indicates whether the corresponding theories about language processing are well-founded.

Our target is not to build a complex model that achieves the highest accuracy to solve this prediction task but rather to combine the effects of different features. Therefore, logistic regression forms the ideal tool that allows us to understand the relationship between interpretable variables and the output.

The surprisal-based features considered for this study are:

1) n-gram log-likelihood (lm)
2) Latent variable PCFG log-likelihood (bkpsl)

The memory-based features used are:

1) Dependency length in terms of discourse referents (dlg)
2) Weighted embedding depth (wtembdep)
3) Lexical (1-best) embedding depth (ldep)

The features had been calculated for both reference and variant sentences for both the Brown and WSJ corpus in Rajkumar et al., 2016. We have utilised these values directly in our study.

Ranking Model

There was a lot of skewness present in the dataset as there were multiple variant sentences for each reference sentence. To tackle this problem, we set up a ranking model to ensure that only one reference and one variant sentence are being compared at a time. This approach was used in Rajkumar et al., 2016. To make the distribution between classes 0 and 1 equal, with any pair of reference and variant sentences, we generate two data points (two pairs):

1. Class label of 1 with features defined as features of reference - features of var, and
2. Class label of 0 with features defined as features of var - features of ref.

Now we have an equal number of pairs belonging to classes 0 and 1. Then we normalised the data points to generate the final dataset.

Logistic Regression Models

10-fold cross-validation was coupled with Logistic Regression were trained on the ranked dataset without any regularisation. Three types of baselines models were chosen:

1) Containing only memory-based features (dlg, wtembdep, ldep)
2) Containing only surprisal-based features (lm, bkpsl)
3) Containing all the five features from Rajkumar et al., 2016 (dlg, wtembdep, ldep, lm, bkpsl)

The classification accuracy and coefficients were compared for different combinations of features.

Note: In the previous work, only one baseline model containing all five features was used. We have also tried to study the effect of the addition of PMI features to surprisal and memory models separately. Also, since we redefined one feature and corrected the errors in the calculation of feature values, the results obtained are very different.

Regression Results

The following tables contain the results of the three kinds of models for both the corpora. The first column of each table represents the baseline model, and the last column contains the metrics for the best model. The best models were attained by building a model with all the features and then dropping them in a principled manner while noting the corresponding changes in classification accuracy. This process was repeated multiple times to identify the most performant models. We have reported the training and testing cross-validation accuracies and the bias and coefficients of the model.

Note: We expect all the PMI-based features to have a larger value for the reference sentence as compared to the variant. Hence, the value of the coefficient of these features should be positive.

1) Models made using combinations of memory-based and PMI features
    a) Brown Corpus

Accuracy and Coefficients	Dlg(1) + wtembdep(2) + ldep(3)	1+2+3+ PMI (adjacent word) (4)	1+2+3 +PMI (nearest dependent of root verb) (5)	1+2+3+ Spearman's correlation for all words (Post) (6)	1+2+3+ Spearman's correlation for dependents (Post) (7)	1+2+3+6+7
Training	69.2871	68.69421	69.0626	69.60579	69.29251	69.70663
Testing	69.1673	68.69548	69.02218	69.48208	69.19156	69.6031
Bias	-0.0005	9.27E-05	-0.00065	0.000131	-0.00061	0.000170
Coefficients	1) -1.2693 2) -0.0929 3) -0.0653	1) -1.2325 2) -0.0940 3) -0.0714 4) 0.2631	1) -1.2494 2) -0.0906 3) -0.0660 5) 0.1747	1) -1.2344 2) -0.0842 3) -0.0690 6) 0.2305	1) -1.2512 2) -0.0909 3) -0.0650 7) 0.0605	1) -1.24037 2) -0.06923 3) -0.08461 6) -0.02307 7) 0.23821

The addition of PMI-based features one at time, leads to an increase in the 10-fold cross validation testing accuracy, and a positive coefficient is observed for the PMI-based feature in all the four models.

    b) WSJ Corpus

Accuracy and Coefficients	Dlg(1) + wtembdep(2) + ldep(3)	1+2+3+ PMI (adjacent word) (4)	1+2+3 +PMI (nearest dependent of root verb) (5)	1+2+3+ Spearman's correlation for all words (Post) (6)	1+2+3+ Spearman's correlation for dependents (Post) (7)	1+2+3+6+7
Training	69.8608	69.76362	69.70526	71.95849	69.98816	72.01740

Testing	69.8914	69.7514	69.75138	72.00741	70.02648	72.04742
Bias	0.0053	0.005273	0.004887	0.005386	0.005263	0.00555
Coefficients	1) -1.4463 2) 0.1582 3) -0.1241	1) -1.4917 2) 0.1398 3) -0.1284 4) 0.4156	1) -1.4369 2) 0.1499 3) -0.1260 5) 0.2287	1) -1.4904 2) 0.1561 3) -0.1281 6) 0.4056	1) -1.4221 2) 0.1544 3) -0.1249 7) 0.1053	1)-1.51113 2) -0.12792 3) 0.15832 6) -0.06666 7) 0.43084

The best models obtained for both Brown and WSJ Corpus contain the same combination of features. This

2) Models made using combinations of surprisal-based and PMI features

a) Brown Corpus

Accuracy and Coefficients	Lm (1) + Bkpsl (2)	1+2+ PMI (adjacent word) (3)	1+2 +PMI (nearest dependent of root verb) (4)	1+2+ Spearman's correlation for all words (Post) (5)	1+2+ Spearman's correlation for dependents (Post) (6)	1+2+3+4+6
Training	78.2995	78.53205	78.47289	78.39222	78.325	78.51457
Testing	78.2792	78.55762	78.47289	78.35181	78.26714	78.60604
Bias	0.0059	0.00566	0.005533	0.005093	0.006275	0.00551
Coefficients	1) 1.4269 2) 1.1219	1) 1.4203 2) 1.0940 3) 0.1885	1) 1.4241 2) 1.1069 4) 0.1415	1) 1.4129 2) 1.1115 5) 0.1846	1) 1.4210 2) 1.1169 6) 0.0753	1) 1.42123 2) 1.09414 3) -0.01226 4) 0.16418 6) 0.03919

b) WSJ Corpus

Accuracy and Coefficients	Lm (1) + Bkpsl (2)	1+2+ PMI (adjacent word) (3)	1+2 +PMI (nearest dependent of root verb) (4)	1+2+ Spearman's correlation for all words (Post) (5)	1+2+ Spearman's correlation for dependents (Post) (6)	1+2+3+4+6
Training	84.6347	84.62697	84.63309	84.64754	84.68088	84.73368
Testing	84.6531	84.6481	84.6481	84.6481	84.68812	84.75314
Bias	-0.0025	-0.00269	-0.00352	-0.00279	-0.00213	-0.00290
Coefficients	1) 1.5841	1) 1.5798	1) 1.5857	1) 1.5817	1) 1.5897	1) 1.59982

	2) 2.2574	2) 2.2506 3) 0.0332	2) 2.2434 4) 0.1063	2) 2.2525 5) 0.0246	2) 2.2630 6) -0.0460	2) 2.24911 3) -0.11325 4) -0.00018 6) 0.16650

3) Models made using combinations of surprisal-based, memory-based, and PMI features.

a) Brown Corpus

Accuracy and Coefficients	Dlg(1) + wtembdep (2) + ldep(3) + lm(4) + bkpsl(5)	1+2+3+4+5+ PMI (adjacent word)(6)	1+2+3+4+5 + PMI (nearest dependent of root verb)(7)	1+2+3+4+5 + Spearman's correlation for all words (Post)(8)	1+2+3+4+5 + Spearman's correlation for dependents (Post) (9)	1+2+3+4+5+6 +7+8+9
Training	79.3495	79.40733	79.32263	79.49876	79.3428	79.54985
Testing	79.3196	79.36807	79.24711	79.3077	79.29548	79.48922
Bias	0.0071	0.00724	0.006777	0.006711	0.007093	0.00620
Coefficients	1) -0.7105 2) -0.3890 3) -0.0891 4) 1.3782 5) 1.0767	1) -0.7011 2) -0.3860 3) -0.0919 4) 1.3750 5) 1.0499 6) 0.1726	1) -0.7021 2) -0.3854 3) -0.0886 4) 1.3772 5) 1.0635 7) 0.1190	1) -0.6981 2) -0.3802 3) -0.0901 4) 1.3679 5) 1.0670 8) 0.1583	1) -0.7097 2) -0.3889 3) -0.0891 4) 1.3780 5) 1.0766 9) 0.0032	1) -0.71866 2) -0.09179 3) -0.38311 4) -0.12663 5) 0.14699 6) 0.05673 7) 0.13831 8) 1.36988 9) 1.04493

b) WSJ Corpus

Accuracy and Coefficients	Dlg(1) + wtembdep(2) + ldep(3) + lm(4) + bkpsl(5)	1+2+3+4+5 + PMI (adjacent word)(6)	1+2+3+4+5 + PMI (nearest dependent of root verb)(7)	1+2+3+4+5 + Spearman's correlation for all words (Post)(8)	1+2+3+4+5 + Spearman's correlation for dependents (Post) (9)	1+2+3+4+5+ 6+9
Training	85.3562	85.29616	85.35285	85.33673	85.3912	85.35507
Testing	85.3634	85.28339	85.36842	85.32842	85.37343	85.38343
Bias	-0.0013	-0.00169	-0.00223	-0.00185	0.000198	0.00020
Coefficients	1) -0.7566 2) -0.3356 3) -0.1302 4) 1.4950 5) 2.1805	1) -0.764 2) -0.3344 3) -0.1308 4) 1.484 5) 2.1595	1) -0.7549 2) -0.3352 3) -0.1305 4) 1.4972 5) 2.1649	1) -0.7607 2) -0.3330 3) -0.1307 4) 1.4887 5) 2.1654	1) -0.7932 2) -0.3391 3) -0.1286 4) 1.5064 5) 2.1945	1) -0.82270 2) -0.12907 3) -0.33854 4) 1.48899 5) 2.15739

		6) 0.0778	7) 0.1034	8) 0.0557	9) -0.1270	6) -0.17900 9) 0.14991

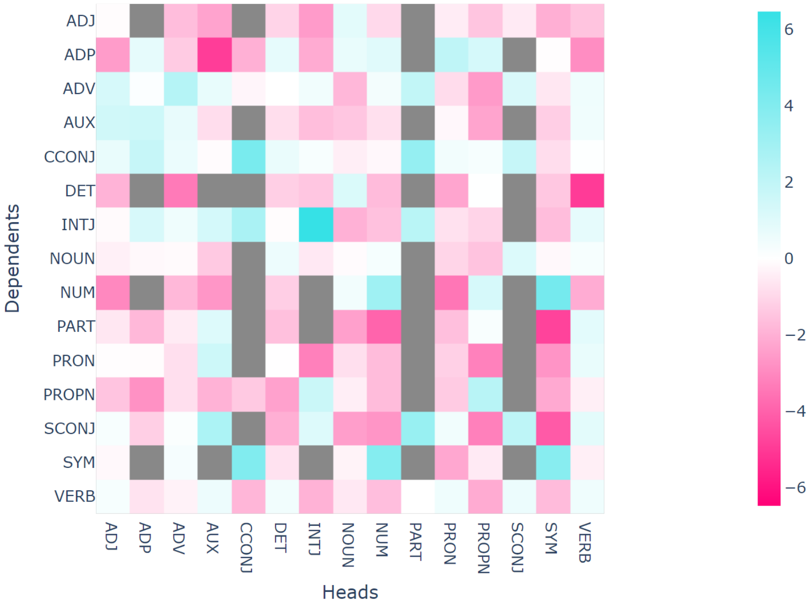
PMI Calculation on Universal Dependencies Treebank

We started by replicating the work done by Futrell on Head Dependency Mutual Information (HDMI) and quantifying the strength of dependency using PMI value.

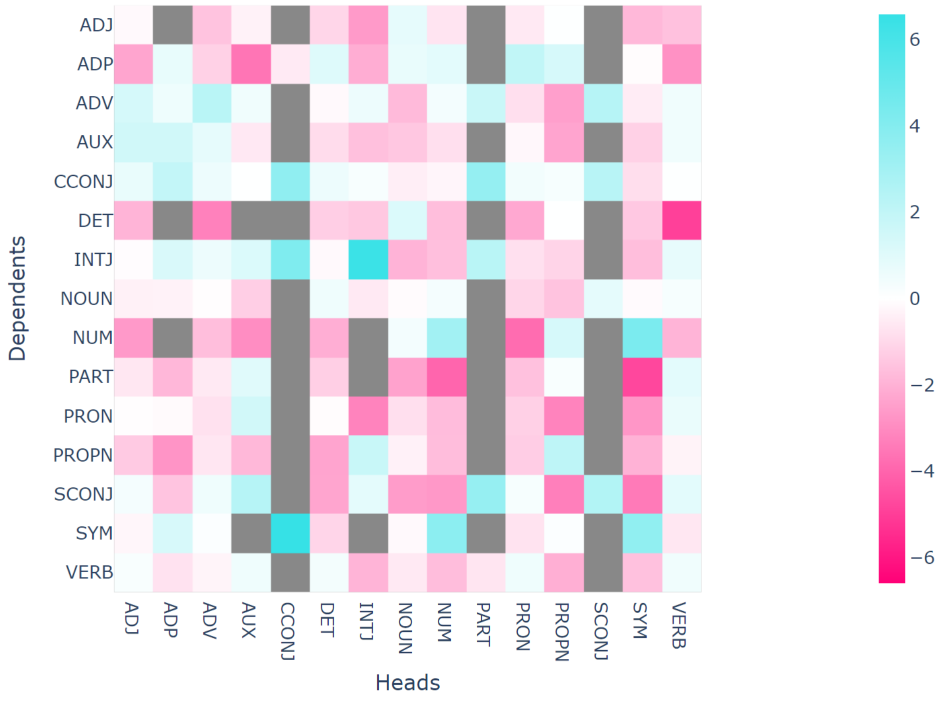
$$HDMI = \log \frac{p(h,d)}{p(h)p(d)}$$

This is the largest corpus of the UD Treebank, EWT is assumed to be representative, and all the calculations have been performed on it. The overall approach was simple; we iterated over all the words in each sentence and counted all the head-dependent relations that were occurring in a 2-d matrix. Using this, we calculated the p(h,d), which gives us the probability of two POS Tags having a head-dependent relation. p(h) and p(d) were also computed for all the POS Tags, which is the probability of that Tag becoming a head and dependent in a sentence, respectively. We compared our results to the ones in the paper.

PMI Value between Head-Dependent Pairs (Futrell et al. (2019))

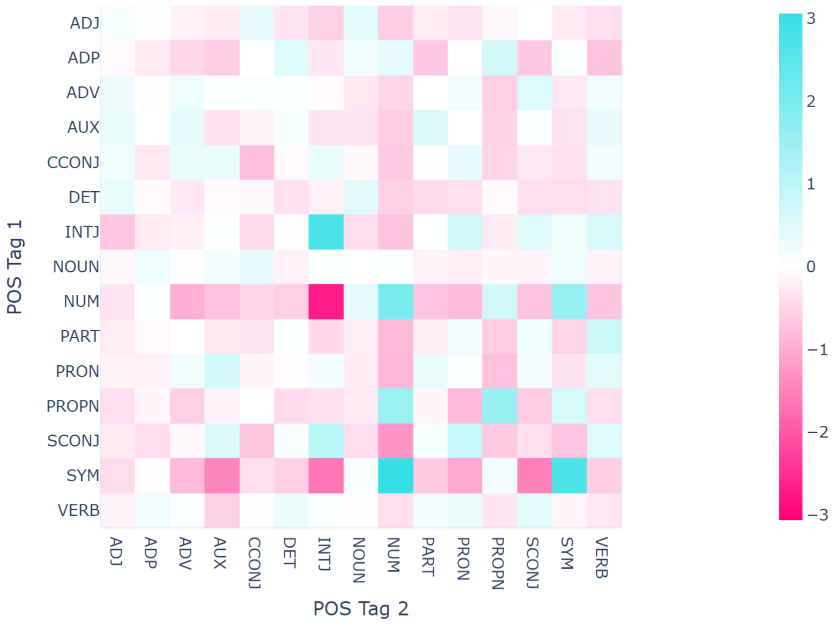


My Results



We then extended this approach to find PMI Values between all the words irrespective of head-dependent syntactic relations. If a sentence has n words, then we look at all n-choose-2 pairs. We count all word pairs having POS Tag 1 and POS Tag 2, where the word with POS Tag 1 occurs before the other in the sentence.

PMI Values between all word pairs



Note: As per the hypothesis, the positive PMI values indicate an attraction between POS Tags, whereas negative values represent repulsion between them.

Next, we recalculated Spearman's correlation coefficient between PMI values and distance for all words using these new PMI values. Following this, we repeated some of the previous logistic regression experiments.

Testing Accuracy (%)	**Memory baseline** Dlg(1) + wtembdep (2) + ldep(3)	1+2+3+ Spearman's correlation for all words (Post) (6)	1+2+3+ Spearman's correlation for all words (Post)(6)
Brown	69.167	69.482	69.736
WSJ	69.891	72.007	71.487

Feature	Brown Corpus (old)	Brown Corpus (new)
dlg	-1.2344	-1.2640
wtembdep	-0.0842	-0.0914
ldep	-0.0690	-0.0644
Spearman's correlation for all words (Post)	**0.2305**	**0.0487**

Feature	WSJ Corpus (old)	WSJ Corpus (new)
dlg	-1.4904	-1.4898
wtembdep	0.1561	-0.1351
ldep	-0.1281	-0.1253
Spearman's correlation for all words (Post)	**0.4056**	**-0.2756**

SVM Results

Performed similar types of experiments with both Linear and RBF SVMs.

WSJ Corpus

Model	**Memory baseline** Dlg(1) + wtembdep (2) + ldep(3)	1+2+3+ Spearman's correlation for all words (Post)(4)	1+2+3+ Spearman's correlation for dependents(Post)(5)
Logistic Regression	69.89	**71.49**	70.03
SVM (Linear)	69.72	**72.01**	69.73
SVM (RBF)	71.98	**73.13**	72.14

Brown Corpus

Model	**Memory baseline** Dlg(1) + wtembdep (2) + ldep(3)	1+2+3+ Spearman's correlation for all words (Post)(4)	1+2+3+ Spearman's correlation for dependents(Post)(5)
Logistic Regression	69.17	**69.48**	69.19
SVM (Linear)	69.85	**69.87**	69.38
SVM (RBF)	72.13	**71.2**	71.62

Maximum Entropy Model for POS Tagging

A paper from 1996 by Ratnaparkhi generates features which capture context and uses a Maximum Entropy Model, a statistical model for POS Tagging.

$$p(h,t) = \pi\mu \prod_{j=1}^{k} \alpha_j^{f_j(h,t)} \qquad \text{defined over } \mathcal{H} \times \mathcal{T}$$

sequence of words $\{w_1, \ldots, w_n\}$

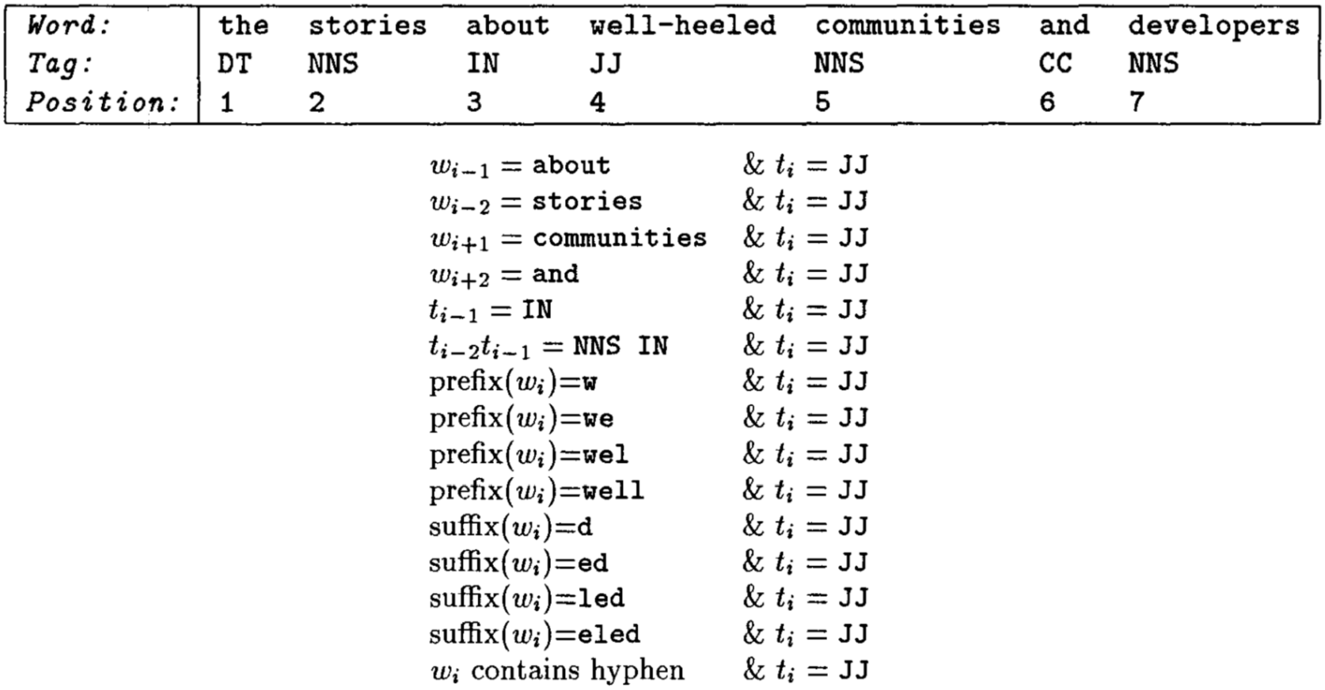
tags $\{t_1, \ldots t_n\}$

$$h_i = \{w_i, w_{i+1}, w_{i+2}, w_{i-1}, w_{i-2}, t_{i-1}, t_{i-2}\}$$

$$f_j(h_i, t_i) = \begin{cases} 1 & \text{if suffix}(w_i) = \text{“}\mathtt{ing}\text{”} \ \& \ t_i = \mathtt{VBG} \\ 0 & \text{otherwise} \end{cases}$$

Here, h_i represents the history for the word $w_{i,}$ whose POS Tag is t_i. The probability of tag t and history h is given by the above formula. Π is a constant (for normalisation), and {μ, α_j} are the model parameters. {f_j } are binary contextual features which carry information about the spelling of the current word or the tags of the previous words.

The following figure from the paper shows the features that were generated for the word ‘well-heeled’.



Word:	the	stories	about	well-heeled	communities	and	developers
Tag:	DT	NNS	IN	JJ	NNS	CC	NNS
Position:	1	2	3	4	5	6	7

w_{i-1} = about	& t_i = JJ
w_{i-2} = stories	& t_i = JJ
w_{i+1} = communities	& t_i = JJ
w_{i+2} = and	& t_i = JJ
t_{i-1} = IN	& t_i = JJ
$t_{i-2}t_{i-1}$ = NNS IN	& t_i = JJ
prefix(w_i)=w	& t_i = JJ
prefix(w_i)=we	& t_i = JJ
prefix(w_i)=wel	& t_i = JJ
prefix(w_i)=well	& t_i = JJ
suffix(w_i)=d	& t_i = JJ
suffix(w_i)=ed	& t_i = JJ
suffix(w_i)=led	& t_i = JJ
suffix(w_i)=eled	& t_i = JJ
w_i contains hyphen	& t_i = JJ

Features Generated From h_4 (for tagging well-heeled)

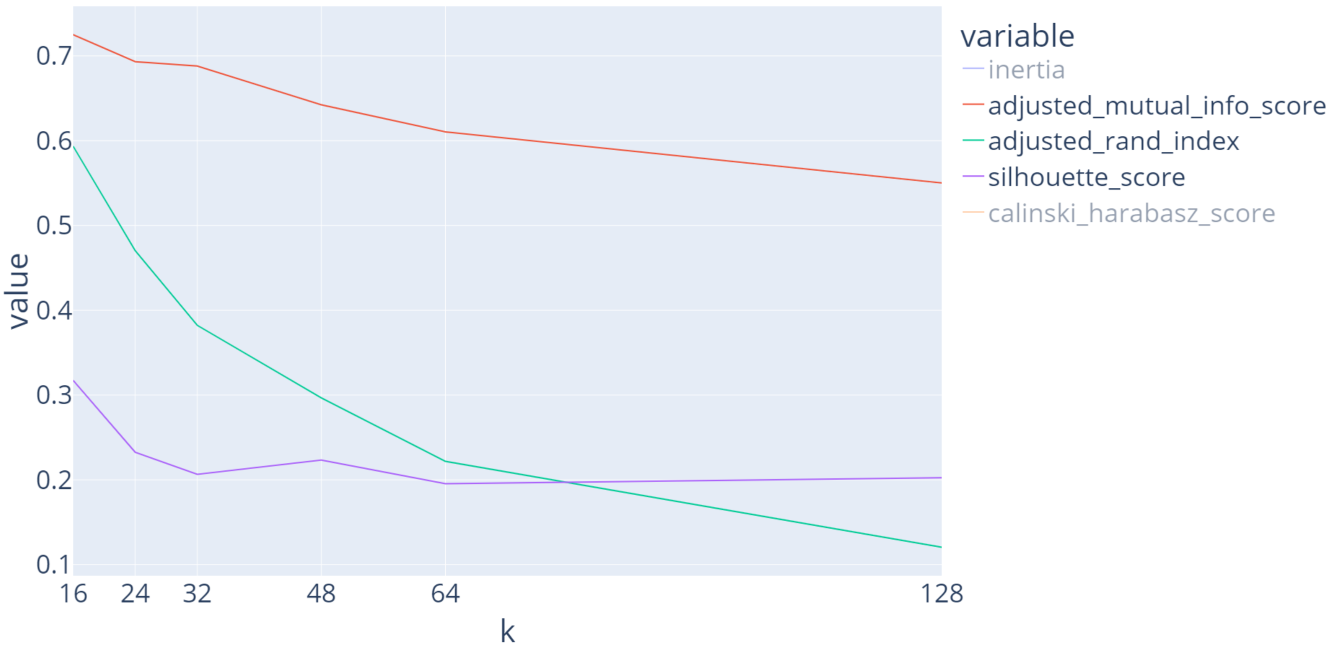
This approach is highly interpretable, and only basic known features are generated. Hence, we decided to replicate the feature generation part to obtain word-level embeddings for our work. We can then perform unsupervised clustering techniques like k-means to devise a new tagging mechanism. We can take large values of k to obtain a fine-grained version of POS tags, and these refined models of PMI will carry more information about the word order.

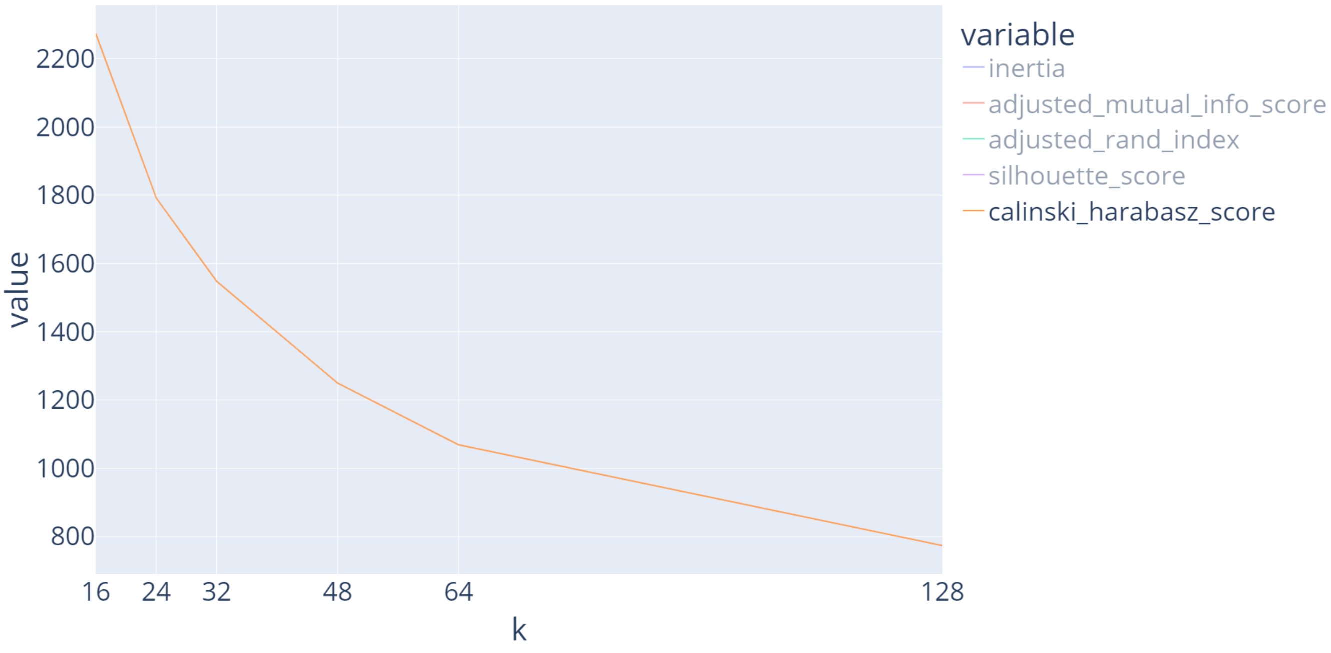
After training the maxent model on UD Treebank, we received word-level embeddings, which were then used to train a k-means model. Then we used these models to find PMI Values for both the head-dependent and all-words settings.

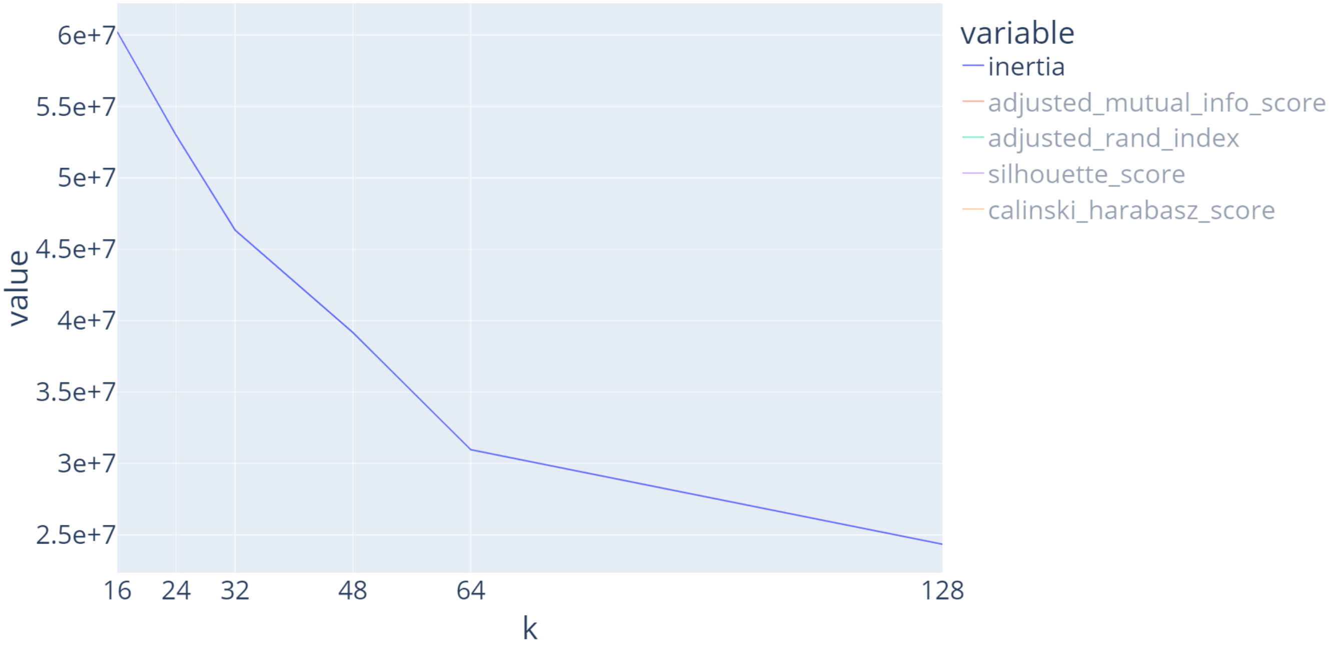
K-Means Clustering

A few well-known supervised and unsupervised metrics are calculated on the K-Means Clustering results, which are:

1. **AMI (Adjusted Mutual Information)** score: This measures the agreement between the ground truth labels and the predicted clusters (permutation invariant). Higher is better; upper bound = 1.
2. **ARI (Adjusted Rand Index)**: This measures the similarity of the predicted clusters to the ground truth labels (again, permutation invariant). Higher is better, range = [-1, 1].
3. **Silhouette Coefficient**: This is an unsupervised measure of how dense and well separated the clusters are. Higher is better, range = [-1, 1].
4. **Calinski Harabasz** score: This is the sum of ratios of inter-cluster dispersion and intra-cluster dispersion for all clusters. The score is higher for dense and well-separated clusters, with no upper bound.
5. **Inertia**: This is the within-cluster sum of squared distances. It defines how coherent the clusters are internally. There is no upper bound, but a score of 0 is optimal.

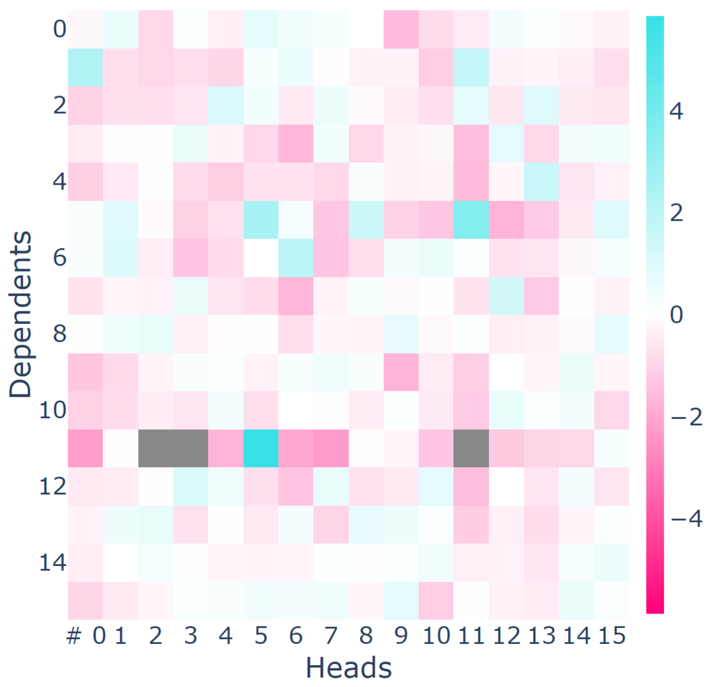


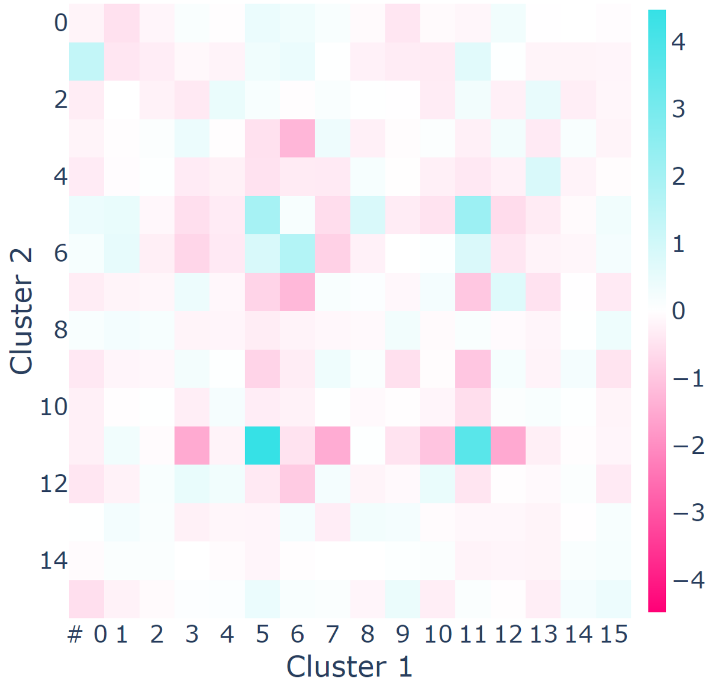

variable
inertia
adjusted_mutual_info_score
adjusted_rand_index
silhouette_score
calinski_harabasz_score
value
2200
2000
1800
1600
1400
1200
1000
800
k
16
24
32
48
64
128


variable
inertia
adjusted_mutual_info_score
adjusted_rand_index
silhouette_score
calinski_harabasz_score
value
6e+7
5.5e+7
5e+7
4.5e+7
4e+7
3.5e+7
3e+7
2.5e+7
k
16
24
32
48
64
128

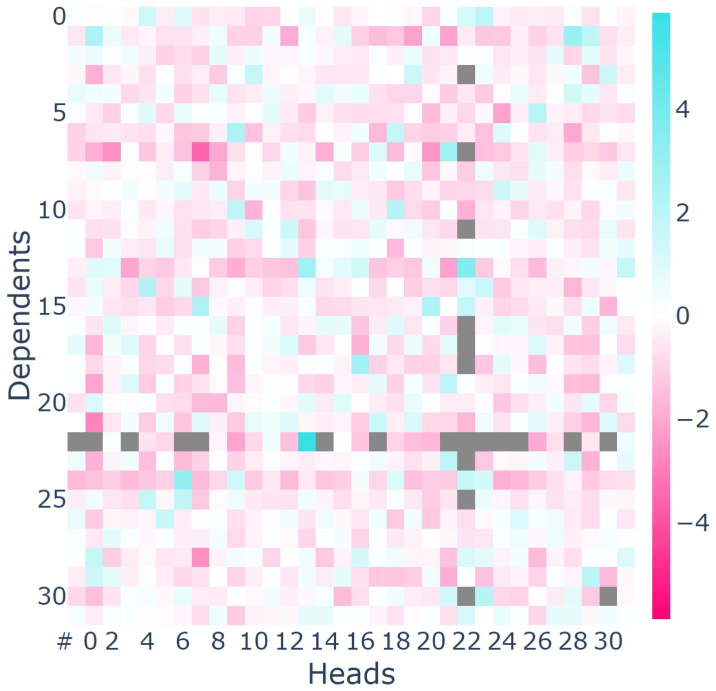
PMI Calculation

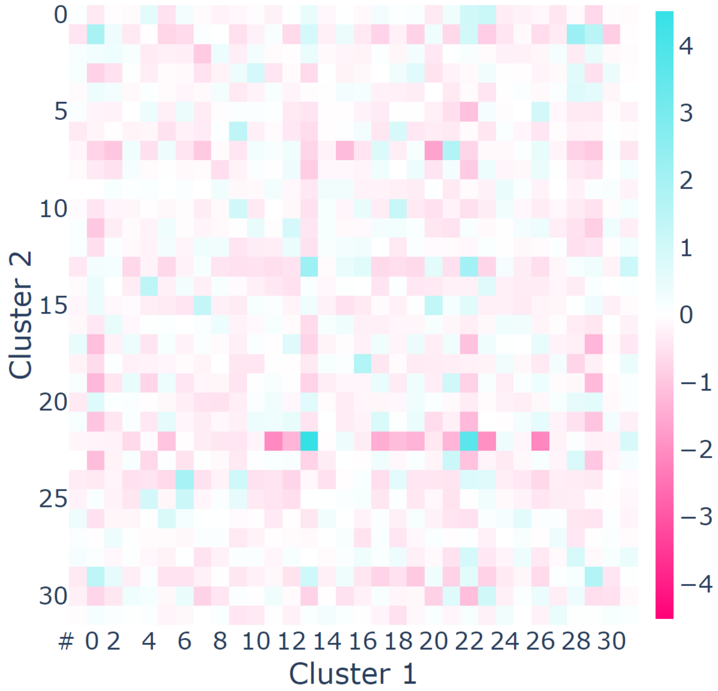
K=16



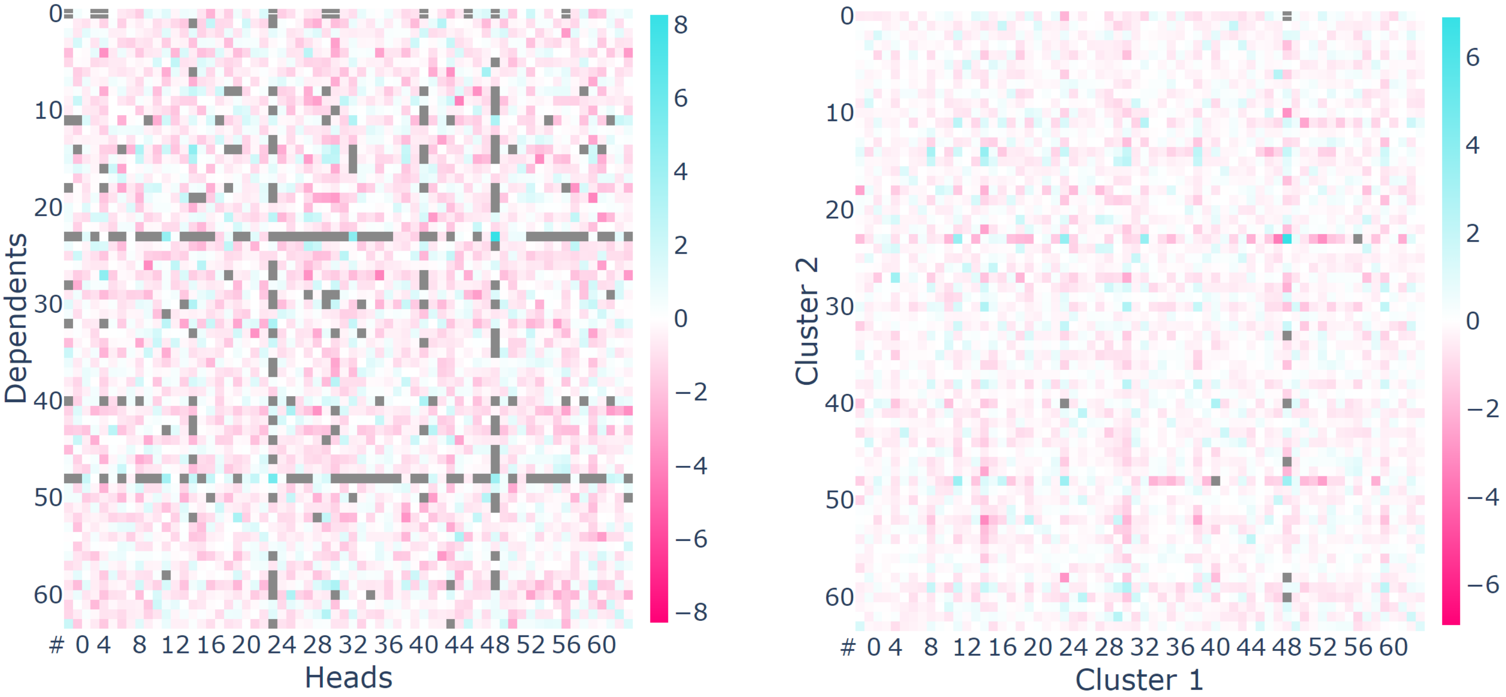


K=32



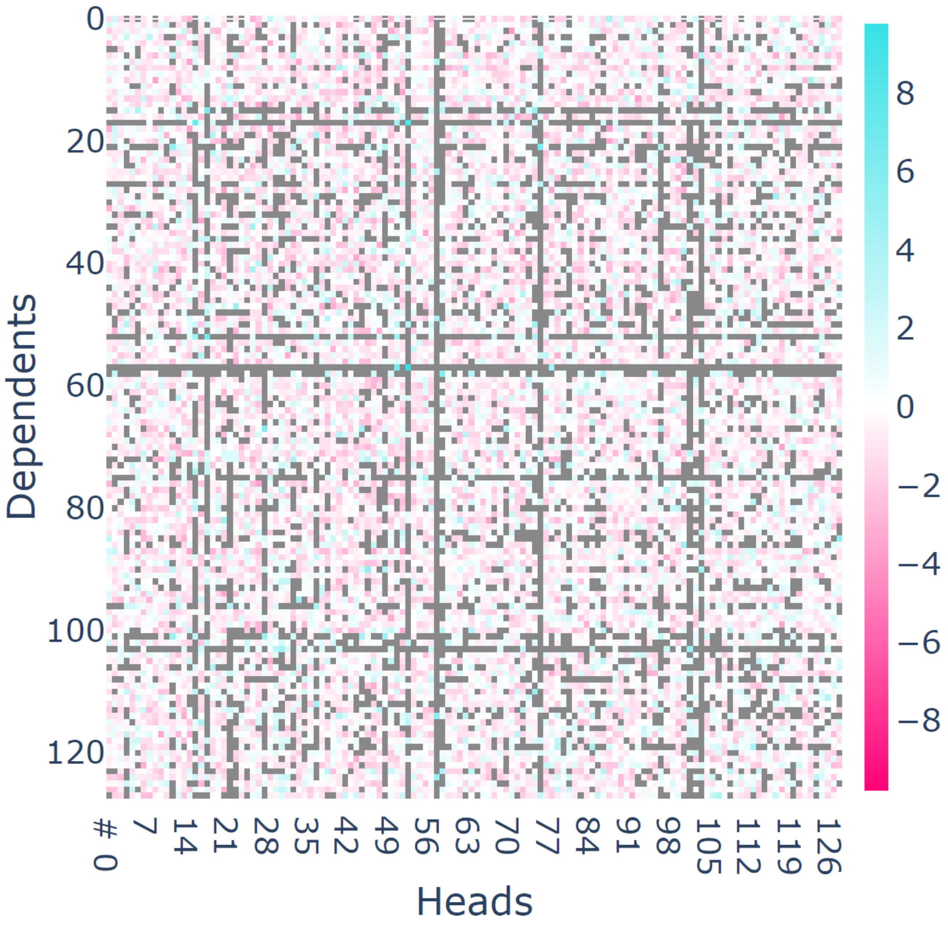


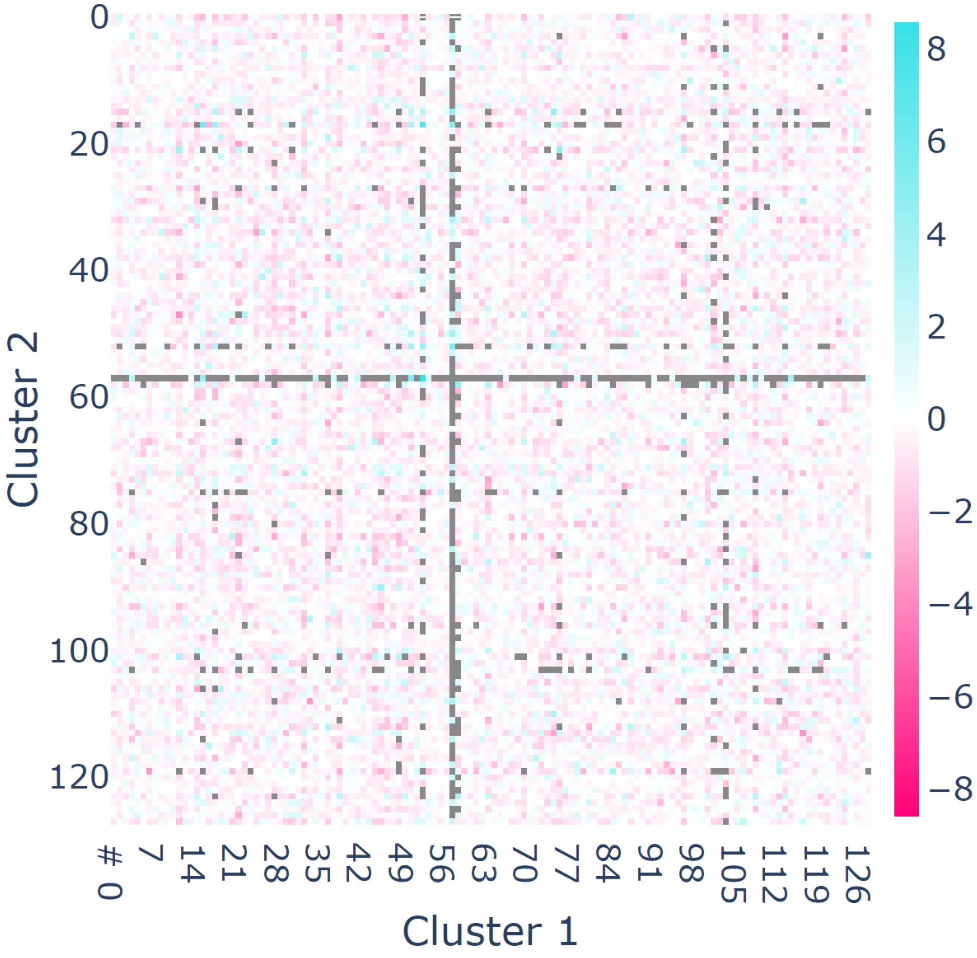
K=64



Dependents
0
10
20
30
40
50
60
#
0 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60
Heads
8
6
4
2
0
−2
−4
−6
−8
Cluster 2
0
10
20
30
40
50
60
#
0 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60
Cluster 1
6
4
2
0
−2
−4
−6

K=128


Dependents
0
20
40
60
80
100
120
Heads
# 0
7
14
21
28
35
42
49
56
63
70
77
84
91
98
105
112
119
126
8
6
4
2
0
−2
−4
−6
−8



These PMI Values were used to compute the four PMI-based features: pmi_adjacent, pmi_neardep, spearman_depedents, and spearman_allwords. We then experimented with logistic regression models containing different combinations of PMI, memory and surprisal based features.

K	Testing Accuracy (%)	**Memory baseline** Dlg(1) + wtembdep (2) + ldep(3)	1+2+3+ Spearman's correlation for all words (Post)(4)	1+2+3+ Spearman's correlation for dependents(Post)(5)
16	Brown	69.167	71.247	69.097
	WSJ	69.891	70.275	71.291
32	Brown	69.167	71.483	71.207
	WSJ	69.891	71.081	73.052
64	Brown	69.167	69.859	69.929

	WSJ	69.891	70.978	69.964
128	Brown	69.167	69.577	70.641
	WSJ	69.891	71.605	70.105

We also evaluated the logistic regression coefficients for these features:

K	WSJ		Brown	
	Spearman_allwords	Spearman_dependents	Spearman_allwords	Spearman_dependents
16	-0.0277	-0.1994	-0.111	-0.0967
32	-0.2863	0.2904	-0.1904	0.267
64	0.1278	-0.1216	-0.0158	-0.0772
128	0.1381	0.08295	0.0071	0.0407

Coefficients for K=32

Features	WSJ	Brown
dlg	-1.478	-1.30
wtembdep	0.1396	-0.1035
ldep	-0.1204	-0.0671
Spearman_allwords	-0.2863	-0.1904

Features	WSJ	Brown
dlg	-1.551	-1.29
wtembdep	0.1575	-0.1187
ldep	-0.1218	-0.0643
Spearman_dependents	0.2904	0.2670

K	Testing Accuracy (%)	**Surprisal baseline** lm(1) + bkpsl (2)	1+2+3+ Spearman’s correlation for all words (Post)(3)	1+2+3+ Spearman’s correlation for dependents(Post)(4)
16	Brown	78.2995	78.205	78.592
	WSJ	84.6347	84.778	84.717
32	Brown	78.2995	78.300	78.315
	WSJ	84.6347	84.701	84.685
64	Brown	78.2995	78.326	78.471
	WSJ	84.6347	84.614	84.744
128	Brown	78.2995	78.255	78.388
	WSJ	84.6347	84.651	84.807

Results for Different Construction Types

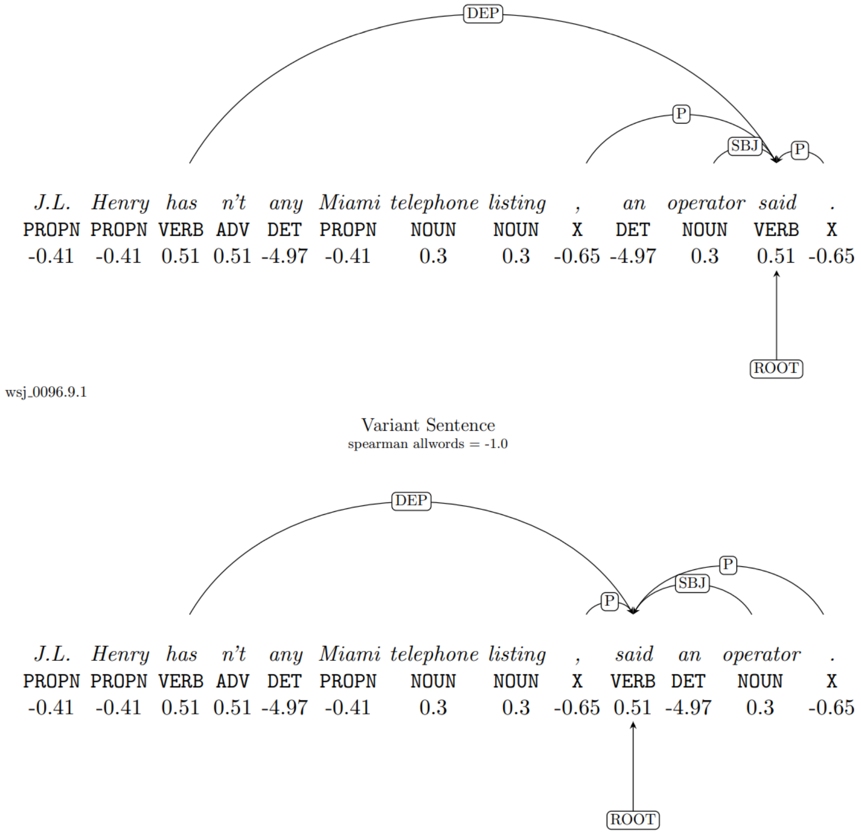
There are four main construction types which were used to generate the variant sentences for Brown and WSJ Corpora: Dative alternation (dat), Quotations (quote), Postverbal adjuncts (post), and Preverbal adjuncts (pre). Construction type has been studied in syntactic priming experiments, where it was found that speakers prefer certain construction types like active voice over passive.
We found the accuracies for each construction separately.

Model		Constructions			
		dat	**post**	**pre**	**quote**
SVM (linear)	Memory Baseline	74.7126	80.0513	40.3514	**34.0156**
	With spearman_all words	73.1211	80.23126	43.29	**55.09582**
SVM (rbf)	Memory Baseline	74.7126	81.2505	40.7987	**46.8641**
	With spearman_all words	73.5632	80.9507	42.3322	**61.5853**

Logistic Regression	Memory Baseline	75.96	80.29	41.77	**45.91**
	With spearman_all words	71.32	81.95	43.29	**44.45**

An interesting observation was that in case of quote construction type, we see a significant increment in the accuracy with both linear and rbf kernel SVMs. This was not observed in case of a logistic regression model, which suggests that SVMs are capturing additional information about this construction type which was being missed previously.

We constructed the dependency trees for some reference variant sentence pairs to learn more about the sentence structures. In most cases, the root verb of the reference sentence was appearing at the end of the sentence and so the spearman's correlation coefficient for all words wasn't calculated.


DEP
P
SBJ
P
J.L. Henry has n't any Miami telephone listing , an operator said .
PROPN PROPN VERB ADV DET PROPN NOUN NOUN X DET NOUN VERB X
-0.41 -0.41 0.51 0.51 -4.97 -0.41 0.3 0.3 -0.65 -4.97 0.3 0.51 -0.65
ROOT
wsj_0096.9.1
Variant Sentence
spearman allwords = -1.0
DEP
P
SBJ
P
J.L. Henry has n't any Miami telephone listing , said an operator .
PROPN PROPN VERB ADV DET PROPN NOUN NOUN X VERB DET NOUN X
-0.41 -0.41 0.51 0.51 -4.97 -0.41 0.3 0.3 -0.65 0.51 -4.97 0.3 -0.65
ROOT

Conclusion

The key conclusions were as follows:

- The addition of PMI features to baseline models increases accuracy, which implies that it successfully captures additional information that is being missed by baseline features.
- The combination of features present in the best surprisal and memory models is the same for both the corpora. This means that the results obtained are generalisable.
- Successfully generated PMI Values using maxent followed by k-means clustering on Universal Dependencies Treebank.
- Extended PMI Calculation to cover all post-verbal words.
- Completed qualitative analysis using dependency trees for different construction types.

Future Work

Some possible directions to expand this work and refine the analysis are:

- Try more complex models such as pre-trained BERT to obtain word-level embeddings.
- Define new PMI-based features and perform similar experiments.
- Currently, only the root verb and it's dependents are used in PMI Features. Try including other head-dependent relations present in the sentence.
- Examine the data sparsity problem which occurs when we compute PMI values. PMI values may affected by the frequency of occurrences pair under consideration.

References

1. Ratnaparkhi A. (EMNLP, 1996) A Maximum Entropy Model for Part-Of-Speech Tagging https://aclanthology.org/W96-0213
2. Futrell, R., Gibson, E. and Levy, R.P. (2020), Lossy-Context Surprisal: An Information-Theoretic Model of Memory Effects in Sentence Processing. Cogn Sci, 44: e12814. https://doi.org/10.1111/cogs.12814
3. Rajkumar R, van Schijndel M, White M, Schuler W. Investigating locality effects and surprisal in written English syntactic choice phenomena. Cognition. 2016;155:204-232. https://doi.org/10.1016/j.cognition.2016.06.008
4. Futrell, R. (2015, August 18). Large-scale evidence of dependency length minimization in 37 languages. PNAS. https://www.pnas.org/content/112/33/10336
5. Richard Futrell. 2019. Information-theoretic locality properties of natural language. In Proceedings of the First International Conference on Quantitative Syntax, pages 2–15, Paris Futrell, ACL, 2019