# Stylistic approaches to predicting Reddit post scores in a diglossia

**Huikai Chua**
Computer Science Dept.
University of Cambridge
hc496@cam.ac.uk

## 1 Introduction

Reddit is a popular social media platform which is organized into different sub-forums, called subreddits. Users can submit original content as top-level posts to each subreddit, which other users can then comment on and either up- or down-vote. The higher the score the post receives, the more 'karma' the author earns.

Although the site sells traditional advertising, many companies also try to promote their products by disguising promotional material as regular user posts. Many subreddits are aware of this and require accounts to meet certain age and/or karma thresholds before they can post on the subreddit. This has led to a rise of a market for Reddit accounts with high karma, and therefore to an interest in engineering posts specifically to farm karma points.

But what exactly makes a post popular? In this project, I apply natural language processing (NLP) techniques to predicting the popularity of a Reddit post. In particular, I focus on style approaches inspired by Bergsma et al. (2012), as past research have found style to be a strong predictor of community response (Tran and Ostendorf, 2016). For comparison, I follow the annotation scheme and evaluation metrics used in a recent paper, Fang et al. (2016), which are described in Sections 4.3 and 6.1 respectively.

In addition to implementing and quantitatively evaluating the models, I also qualitatively evaluate and analyze the results. I investigate whether the observation that conformation to community style elicits greater endorsement (Tran and Ostendorf, 2016) still applies in a diglossic situation. Finally, I examine the top features from each model.

## 2 Related Work

Much research has gone into investigating what makes a social media post popular, including some specifically focused on Reddit. Lakkaraju et al. (2013) controlled for the content of the post by concentrating on image submissions, which are frequently re- or cross-posted to different communities by different authors (or a single particularly shameless author). They found that the title of a submission played a role in determining its success, where titles specifically engineered towards the community it was posted in (for example, by using community-specific words) performed better (Lakkaraju et al., 2013).

Tran and Ostendorf (2016) took this a step further and trained separate models for the content (using Latent Dirichlet Allocation (LDA)) and the style of the language used (by replacing topic words with their part-of-speech tags). They computed the Spearman rank correlation between scores and post representations, and found that the style model was much better at predicting of the success of a post than the content model (Tran and Ostendorf, 2016). In other words, they found that these subreddits had their own community style, and posts which are stylistically more similar to it are more likely to be well-received.

Fang et al. (2016) is the paper which is closest to the aim of this project. They divided posts into eight different bins which are automatically determined by the score distribution of that particular subreddit, and evaluated model performance using a somewhat modified macro-averaged F1 score. However, while Fang et al. (2016) focused on modelling the conversational context of a post, I instead apply syntactic approaches to model the style of the community.

I take cues from Bergsma et al. (2012) to achieve this. They grouped their features into

three broad categories: word (bag-of-words), style, and syntax features. For style features, they defined style words to be punctuation, stopwords, or Latin abbreviations, and replaced all non-style words with their part-of-speech (POS) tags. Meta-features such as average word and sentence lengths were also used. For syntax features, they included a feature for every unique context free grammar and tree substitution grammar rule, as well as Charniak and Johnson re-ranking features.

In this project, I adopt the evaluation approach of Fang et al. (2016). Lakkaraju et al. (2013) used regression to evaluate model performance, which does not adequately account for the Zipfian distribution of scores – as can be seen in Table 1, the majority of posts fail to garner any positive attention and only a small handful of posts achieve a high score.

## 3 Approach

I adopt Bergsma et al. (2012)'s three-pronged approach to stylometry. For word features, I use BERT; for style features, I used term frequency – inverse document frequency (TF–IDF) stopword vectors; for syntax features, I used dependency relations and part-of-speech (POS) tags.

### 3.1 TF–IDF

TF–IDF is a standard word vectorization technique using a 'bag of words' model, i.e. it assumes that each document is made of a collection of words, where the probability of a given word occurring is independent of what other words are in the document or its position in the text. It consists of two main components, a term frequency component which estimates how likely a word $w$ is to appear in documents of a class $c$:

$$tf(w, c) = \frac{count(w, c)}{count(w)}$$

and an inverse document frequency component (Jones, 2004) which estimates how informative a word is by measuring how rarely it is seen:

$$idf(w) = -\log \frac{\sum_{i=1}^{N} 1_{w \in d_i}}{N}$$

Here, $N$ is the total number of documents and $1_b$ is an indicator function which returns 1 if the condition $b$ is true and 0 otherwise, i.e. if the word $w$ is seen in document $d_0$, then the indicator function $1_{w \in d0}$ would return 1.

Using the TF–IDF scheme, the overall probability of a word $w$ appearing in a given document is estimated to be:

$$p(w) = tf(w) * idf(w)$$

Usually, when using TF–IDF to vectorize a document, each word in its vocabulary is included as a feature. In this project, I use TF–IDF with stopwords and punctuation only.

### 3.2 BERT

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) was a landmark language representation model which built on the then-recent transformer (Vaswani et al., 2017) model to achieve state-of-the-art performance on a wide variety of tasks.

The underlying transformer architecture consists of an encoder-decoder structure (Vaswani et al., 2017). Each of the encoder and decoder layers use attention mechanisms to attend to the most relevant parts of the input in generating outputs. The key innovation in the BERT paper was the bidirectional approach they used to train the model (Devlin et al., 2018), which they named Masked Language Model (MLM). In MLM, random tokens in the input are masked and then predicted during training. The deep bidirectional representation learned allows BERT to achieve state-of-the-art performance on many tasks without the need for task-specific architecture. This has led BERT and its variants to become ubiquitous in every kind of NLP task.

Past investigations have suggested that BERT is not just good at capturing the meaning of sequences, but is also sensitive to the syntax of phrases. Goldberg (2019) ran a series of syntactic test cases and found that BERT performed well on all; Jawahar et al. (2019) suggested that BERT layers encode linguistic information hierarchically, with surface information in lower layers, syntactic in the middle, and semantic information at the top. Thus, it seems that BERT would be able to capture both the contents of posts as well as their style, making it particularly suitable for this task.

### 3.3 Part-of-speech tags

Part-of-speech (POS) is a concept used to classify words according to their grammatical function. POS tags are used to group words with sim-

ilar syntactic functions together. Some are treated as fundamental POS tags, such as nouns and verbs, while others are defined iteratively in terms of their relation to other POS tags. For example, pronouns are defined relative to nouns as words which can take the place of a noun or noun phrase.

When modelling the style of a text, it is often desirable to capture its linguistic form while ignoring its subject. Vectorizing text by using their POS tags, rather than words, allows us to do so while minimizing the influence of its content.

### 3.4 Dependency relations

In this project, I supplement the POS tag of each word with its dependency rule. Dependency relations express the grammatical relationships between different words in a phrase. For example, we have rules to express "object of a verb" ('dobj') and "noun subject of a verb" ('nsubj'). There is generally one rule per word, i.e. each word has a single grammatical head, although it is possible for a single word to be the head of more than one word. Together, these relations help to capture the grammatical structure of a phrase or sentence.

Dependency parsers extract this grammatical structure according to their particular dependency scheme. In this project, I use the dependency parser used by spaCy which uses a modified version of the Universal Dependencies set, called ClearSTYLE by ClearNLP[1]. The scheme includes common dependency rules like 'nsubj' and 'dobj' as mentioned, as well as some less common ones like 'nsubjpass' for the subjects of passive verbs.

## 4 Dataset

### 4.1 Data collection

Data was scraped from Reddit by querying the Pushshift API[2]. 3 years' worth of posts, ranging from 1 January 2017 to 31 December 2019, were collected for each subreddit. The most recent year, 2020, was excluded as I expected social media activity in 2020 to be significantly different compared to earlier years due to the unique global situation and extensive lockdowns around the world. To ensure each post had sufficient linguistic content, I excluded any posts containing less than 101 characters.

Due to the limited time frame of the project, I decided to only use top-level posts as past research had shown that the complex Reddit discourse structure plays a role in determining the score a comment receives (Fang et al., 2016). Intuitively, we can attribute this to Reddit's ranking system, where higher-scoring comments appear higher on a post's page. Since people cannot vote on a comment they do not read, replies to a lower-scoring comment are less likely to receive attention and hence a high score, through no inherent textual fault of its own. Other factors such as time also play an important role, as the system also ranks more recent comments higher on the page. Therefore, the actual content of the comment plays a comparatively limited role in final score it receives. Therefore, to allow me to focus on linguistic characteristics and avoid the complexity of modelling the discourse structure, I chose to only include top-level posts.

### 4.2 Subreddit selection

In the original paper, the authors had used data from three subreddits: r/AskMen, r/AskWomen, and r/Politics. Unfortunately, none of these were suitable for this project. As the names of the first two imply, the top-level posts to their communities mainly consist of questions, and hence contain limited linguistic content to use: a scrape of 2019 yielded only a few hundred posts with more than 100 characters. The last subreddit, r/Politics, contained even less textual posts as most are simply links to news articles – a scrape of 2019 yielded around 30 posts with more than 100 characters.

Therefore, I selected another set of subreddits to investigate. As one of my aims was to investigate the stylistic characteristics of communities, I selected a subreddit with what I expected to be a distinctive linguistic style – the Singaporean subreddit[3] (SG). Singaporeans speak a distinctive flavour of English dubbed "Singlish", which has drawn much linguistic interest as the *lingua franca* of different cultural communities. It serves as the vernacular in the diglossic Singapore, where the Standard British English serves as the acrolect.

Therefore, for comparison, I also select the United Kingdom subreddit[4] (UK). Although the population sizes of the two countries are quite different (roughly 5 million Singaporeans versus over

[1] https://spacy.io/api/annotation#dependency-parsing-english

[2] https://github.com/pushshift/api

[3] reddit.com/r/singapore

[4] reddit.com/r/unitedkingdom

| Level | r/singapore | | r/UnitedKingdom | |
|---|---|---|---|---|
| | Size | Score Cap | Size | Score Cap |
| 0 | 15633 | 2 | 9246 | 2 |
| 1 | 4797 | 14 | 2466 | 14 |
| 2 | 2394 | 36 | 1246 | 64 |
| 3 | 1200 | 74 | 633 | 191 |
| 4 | 620 | 151 | 318 | 531 |
| 5 | 310 | 284 | 159 | 1086 |
| 6 | 156 | 507 | 79 | 1762 |
| 7 | 156 | - | 80 | - |
| Total | 25266 | | 14227 | |

Table 1: Distribution of classes for both subreddits.

60 million UK citizens), I found that the subreddit sizes were similar, with roughly 300k participants in SG and 400k participants in UK.

### 4.3 Annotations

In order to compare results, I followed the annotation procedure described in Fang et al. (2016). First, all posts with a score below 2 were labelled as the lowest class, Level 0. This threshold was selected for the base class as all new posts are initialized with a score of 1 (Fang et al., 2016). For the next level, the median of the remaining posts was computed and all posts with a score lower than the median labelled as 1. This process is repeated for each of the levels 2-6. Finally, the remaining posts are labelled as the highest class, Level 7. The distribution for each subreddit along with the respective class thresholds are summarized in Table 1.

## 5 Implementation

### 5.1 Style features

I used stopword TF–IDF vectors for the style features, implemented using sklearn's TfidfVectorizer[5]. The vocabulary is predefined to be either a stop-word, using NLTK's English stop-word list, or a punctuation character, from Python's inbuilt string module (accessible via 'from string import punctuation'). NLTK's English stop-word list, consists of 179 stop-words including determiners ('the', 'a'), pronouns ('he', 'she'), prepositions ('before', 'after'), quantifiers ('all', 'some'), among others.

[5] scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

### 5.2 Word features

To capture the content of a post, I used the pretrained BERT model to produce post embeddings. Since BERT is designed to encode sequence-level representations (Devlin et al., 2018), I first split each Reddit post into sentences using NLTK's sent_tokenize. Then, each of the sentences were tokenized and converted to BERT embeddings in the usual way. Finally, the embeddings for each sentence were averaged to produce the overall post-level representation.

### 5.3 Grammatical features

I used the spaCy parser to extract dependency relations and part-of-speech (POS) tags. First, I hand-compiled the lists of relations and POS tags from the documentation[6]. The lists can be found in the appendix.

Then, the dependency and POS labels for each word were replaced by their positions in the respective lists. I also included the POS labels of the heads of each word. Before being passed to the model as input, each of the three vectors (dependency tag, POS label, head POS label) were normalized with L2 norm.

### 5.4 Model

As I wanted to focus on feature rather than the model engineering, I used a tried-and-tested model for imbalanced class distributions: random forest classifiers. I opted to use the RandomForestClassifier from sklearn[7] as it was easy to set up. In particular, it had a class_weight hyperparameter which we can use to weight the relative importances of different classes. For this project, I set the weights for each Level $i, 0 \leq i \leq 7$ to be

$$weight_{Level i} = \frac{\#samples_{Level 0}}{\#samples_{Level i}}$$

This sets the weight of Level 0 to the default of 1.

## 6 Quantitative evaluation

### 6.1 Evaluation metric

For comparability, I replicate the evaluation procedure described in Fang et al. (2016). First, the F1 score for each of the Levels 1-7 were computed, treating each sample with a score below that level

[6] https://spacy.io/api/annotation

[7] scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

as a negative example. Concretely, the number of true positives (TP), false positives (FP) and false negatives (FN) for a given level *i* are computed as

$$TP_i = \#\text{samples } s, s_{true} \geq i \ \& \ s_{predicted} \geq i$$

$$FP_i = \#\text{samples } s, s_{true} < i \ \& \ s_{predicted} \geq i$$

$$FN_i = \#\text{samples } s, s_{true} \geq i \ \& \ s_{predicted} < i$$

Then, the F1 score for that level $i$ is computed using the usual formula

$$F1_i = \frac{TP_i}{TP_i + \frac{1}{2} * (FP_i + FN_i)}$$

Finally, the final score for that model is obtained by averaging over the F1 scores for each level. Fang et al. (2016) had designed this evaluation metric such that the higher levels, which are of greater interest, are weighted more highly. For example, for the SG score distribution, a model which predicts only Level 1s would obtain an F1 of 0.0789, while a model which predicts only Level 8s would obtain an F1 of 0.176. Level 0 is excluded in computing the average, as using the scheme described above the F1 score would always be 1.

### 6.2 Results

In total, I tried six different combinations of the three different types of features. First, I tried each of the style features, BERT embeddings, and grammatical (POS and dependency labels) features separately. Then, I tried individually adding the other two types to the weakest baseline, which in this case was the grammatical model. Finally, I tried a combination of all features together. In this project, I used stratified five-fold cross-validation and report the average F1 score (computed as described above) across all folds. The results can be found in Table 2.

In all cases, the models clearly out-performed the simplistic baseline of 0.176 for a model which predicts only the highest class. The results are also better than the roughly 0.50 F1 scores achieved by Fang et al. (2016), though they are not directly comparable due to differences in the data used. Of particular note is that their model appeared to perform better on lower levels, with F1 scores for Level at least 0.60 for each of the three subreddits used compared to the average of 0.50. However, in this project, the models used performed better at higher levels. The individual F1 scores for each level, from the model with all features, can be found in Table 3.

| | SG | UK |
|---|---|---|
| **Style** | 0.748 | 0.788 |
| **BERT** | 0.749 | 0.793 |
| **Gram.** | 0.733 | 0.781 |
| **Gram. + style** | 0.75 | 0.792 |
| **Gram. + BERT** | 0.751 | 0.793 |
| **All** | 0.751 | 0.793 |

Table 2: F1 scores for each subreddit for each model.

Although the scores for each model are similar, the results are consistent across the two subreddits, r/Singapore (SG) and r/UnitedKingdom (UK). In both cases, BERT performs the best out of the three baselines, and indeed was improved only slightly by 0.02 for SG when other features were added, and not at all for UK.

Between SG and UK, all models performed significantly better on the UK dataset. This is possibly due to there being a more consistent group style for UK, compared to the diglossic situation in Singapore. From Table 3, the F1 scores for the lowest Levels 1 and 2 for SG are very similar to UK scores. Furthermore, the gap between the two increases as the level increases, for a total gap of 0.85 points at the highest level (compared to an average difference of only 0.42 points). As we will see in Section 7.2, a diglossic situation with two competing dialects makes it a bit more difficult to craft an effective style.

## 7 Qualitative evaluation and analysis

### 7.1 Feature importances

To get a better picture of which features contributed more to model performance, I also looked at the feature_importances_ property of each model. To differentiate the POS tags of head words from the POS tags of words, I appended 'head_' in front of head POS tags (henceforth referred to as head-POS). The top 10 non-BERT features, i.e. either a POS, dependency, head_POS tag, or a stopword or punctuation for SG and UK are tabulated in Table 4 and 5 respectively.

| | Level 1 | Level 2 | Level 3 | Level 4 | Level 5 | Level 6 | Level 7 |
|---|---|---|---|---|---|---|---|
| **SG** | 0.721 | 0.727 | 0.711 | 0.734 | 0.746 | 0.781 | 0.835 |
| **UK** | 0.728 | 0.724 | 0.743 | 0.768 | 0.813 | 0.851 | 0.920 |

Table 3: F1 scores for each individual level for the model with all features.

| Rank | Gram. only | Style only | Gram + Style | Gram + BERT (position) |
|---|---|---|---|---|
| **1** | punct | . | . | head_ADV (15) |
| **2** | PUNCT | the | ? | empty dep relation (22) |
| **3** | ROOT | ? | PUNCT | head_SCONJ (25) |
| **4** | DET | to | punct | prt (42) |
| **5** | advmod | , | ROOT | head_PUNCT (48) |
| **6** | poss | and | the | dative (50) |
| **7** | aux | i | advmod | DET (52) |
| **8** | NOUN | a | head_VERB | poss (104) |
| **9** | det | of | AUX | PUNCT (118) |
| **10** | ADJ | in | DET | PART (128) |

Table 4: Top 10 non-BERT features for selected models on the SG dataset.

| Rank | Gram. only | Style only | Gram + Style | Gram + BERT (position) |
|---|---|---|---|---|
| **1** | amod | . | / | head_X (5) |
| **2** | ROOT | / | . | CCONJ (30) |
| **3** | NOUN | the | ROOT | PART (46) |
| **4** | DET | to | head_VERB | amod (48) |
| **5** | PUNCT | a | punct | cc (57) |
| **6** | punct | i | i | conj (137) |
| **7** | head_VERB | and | AUX | advmod (173) |
| **8** | PRON | , | head_NOUN | relcl (174) |
| **9** | aux | ” | PUNCT | SPACE (176) |
| **10** | cc | ? | amod | NOUN (220) |

Table 5: Top 10 non-BERT features for selected models on the UK dataset.

#### 7.1.1 Does BERT capture grammatical features?

Although model performance improved only a little bit when BERT embeddings were added to grammatical features, the most informative features were completely taken over by BERT features. For SG, the highest ranking non-BERT feature was 'head_ADV' at 15th place, with the next one, 'head_SCONJ', coming in 10 places lower. For UK, the top two were at 5th ('head_X') and 30th ('CCONJ') place respectively. This does suggest BERT is capable of capturing the syntax of a sentence in its embedding, as it seems to have replaced grammatical features when it was added to the model.

Of particular note are the changes in the individual features' rankings. In the grammatical features only model, the top features are occupied by dependency and POS tags; the highest ranking head-POS features for SG and UK are 'head_NOUN' and 'head_VERB' at 12th and 7th place respectively. The relatively higher rankings of head-POS tags after adding BERT suggest that it might not be as good at capturing more complex grammatical relationships.

#### 7.1.2 Overlap between grammatical and style features

There is a noticeable overlap between grammatical and style features, where the top-ranked features for grammatical and style mirror each other. For example, punctuation ranks among the most informative style features, particularly for UK where they occupy 5 out of the top 10 spots despite making up only 15% of the roughly 200 style features. Among the 100 grammatical features, the dependency rule 'punct' and POS tag 'PUNCT' also rank highly. A similar trend can be seen for determiners, which rank highly as both style features (in the form of the stopwords 'the' and 'a') as well as grammatical features (in the form of the dependency rule 'DET'). This possibly contributes to the very similar performances of the style and grammatical models.

However, it appears that the more specific style features generally perform better. When grammatical and style features were combined for SG, the specific punctuation characters '.' and '?' appear before 'PUNCT' and 'punct'. Similarly, the determiner 'the' appears before the dependency rule 'DET'. This might explain the difference between the individual style and grammatical models, where the style model performed better on both the SG and UK datasets. Although the top features from both form a common subset, the more specific features found in the style model are better predictors.

### 7.2 Does group style conformation still apply with a sociolect continuum?

Linguists have observed that Singapore English is organized along a sociolect continuum from an informal basilect (Singlish), to a formal acrolect, which has minimal features of Singlish and is essentially Standard British English. Use of the acrolect is generally associated with better education, and therefore higher social economic status. On the other hand, despite top-down efforts from the Singaporean government, the basilect is the dialect used by the average Singaporean in everyday situations, and is closely associated with the Singaporean identity. In fact, Singaporean politicians intentionally include Singlish phrases in election speeches in efforts to appear more down-to-earth and likeable.

Therefore, it would be very interesting to investigate if Tran and Ostendorf (2016)'s observation that conformation to group style results in greater community endorsement still applies in this diglossic situation, when there are two competing dialects. Although Singlish is the community dialect and is closely linked with group identity, the Standard English acrolect has higher prestige.

#### 7.2.1 Syntactical features

Since the acrolect should be close to Standard British English, I decided to assess this by first computing the Euclidean centre of Level 7 posts from UK. Then, for each of the Levels 0-7, I computed the average Euclidean distances from Singaporean posts to the UK centre. For comparison, I also compute the average distances for UK posts. The distances for each of the three types of features are tabulated in Table 6. Note that, due to different dimensions and normalization, the distances for each feature are not directly comparable to that of other features.

Across all three features, Level 0 SG posts are generally less similar to the UK centre than Level 0 UK posts, possibly due to greater presence of the basilect. However, at the top level, SG posts are even *more* similar than the original posts the centre was calculated from. This suggests that in-

deed the Standard British English acrolect holds more prestige and draws greater community endorsement.

Separately, the consistent trend in the Style column where posts from higher levels are more similar to the Level 7 centre than lower level posts supports the hypothesis that there is a community style and posts which are more similar to it receive greater community endorsement.

#### 7.2.2 Lexical features

We can see that stylistically and grammatically, the most popular posts from SG are very similar to British English. However, what about lexically? Singlish has a vocabulary full of borrowed words and phrases from the different cultural groups of Singapore. As mentioned earlier, politicians often try to build rapport by sprinkling speeches with Singlish terms. Would we observe something similar on Reddit? I decide to investigate the prevalence of Singlish terms by level.

Compiling a Singlish lexicon can be very tricky due to several reasons:

- Because many vocabulary items are borrowed from other languages, there are different alternate spellings possible. For example, the common exclamation 'wa lao' can also appear as 'wa lau', 'wah lao', 'walao', or any combination thereof.
- Many items have also taken on a different meaning from in their original languages, and this is particularly problematic for words which come from English. For example, the English word 'mug' has been re-purposed to mean 'to study very hard'. It can be challenging to differentiate Singlish and English uses of the word, given that 'mug' can be used as a verb in both.
- Other phrases have taken on entirely new meanings in the local context. For example, if you see or hear the Hokkien phrase 'kee chiu', it is not very likely to be in reference to its original meaning of 'hands up', but rather a local politician who had the misfortune of coming across as inauthentic when using it in a speech[8].

With that in mind, I compiled a list of 56 everyday Singlish words and phrases, including alternative spellings where practical. I excluded phrases with specific niches, like the names of foods or military terms (common in Singapore where all males have to enlist for 2 years), as well as any crude terms (also unfortunately common due to said enlistment). For the full list of phrases used, please refer to the appendix.

I computed the average number of such Singlish words or phrases used per 1000 words per post for each of the Levels 0-7. The results, which can be found in Table 7, confirm the earlier hypothesis that effective use of Singlish words helps earn more community endorsement. We see a somewhat U-shape in the frequency of Singlish terms; the least popular posts include more Singlish than the middlingly popular posts, likely due to greater influence of the basilect, while posts on the highest levels utilise Singlish vocabulary in tandem with the acrolect to achieve the most popularity.

A reading of the Level 7 texts including Singlish terms confirm that this is indeed the case. For example, one post is written in very eloquent standard English[9], but includes Singlish quotes as well as specific, appropriate Singlish terms (with English explanations in brackets) .

## 8 Future Work

In this project, I attempted to include context-free grammar (CFG) features using CoreNLP. Unfortunately, I ran into unexpected issues and was not able to get it or a different Python parser to work in time for the project. Given more time, I would have liked to include these features. It would probably be interesting to also augment the parsers with Singlish context-free grammar features, in order to further confirm or disprove the theory that the most popular posts use the acrolect, i.e. the least syntactical features from Singlish, despite having the highest prevalence of Singlish terms.

Other avenues to explore include the extensive metadata included with each post from the Pushshift dataw. Since I wanted to focus on modelling the language used in each community, I largely ignored the use of other important metadata such as author reputation (Wei et al., 2016) and posting time (Lakkaraju et al., 2013). As can be seen in Figures 1 and 2, the number of posts in each class can vary greatly with time. Ideally, we would also factor in the impact of these metadata when predicting the class of a post, for example

[8] https://en.wikipedia.org/wiki/Chan_Chun_Sing

[9] https://www.reddit.com/r/singapore/comments/8gfewd/the_singaporean_male_version_of_metoo_an_exguards/

| Level | BERT | | Style | | Gram. | |
|---|---|---|---|---|---|---|
| | SG | UK | SG | UK | SG | UK |
| 0 | 4.45 | 4.35 | 0.875 | 0.847 | 0.704 | 0.715 |
| 1 | 4.40 | 4.31 | 0.874 | 0.845 | 0.688 | 0.697 |
| 2 | 4.33 | 4.34 | 0.877 | 0.832 | 0.678 | 0.716 |
| 3 | 4.39 | 4.34 | 0.848 | 0.830 | 0.676 | 0.733 |
| 4 | 4.17 | 4.35 | 0.826 | 0.826 | 0.632 | 0.722 |
| 5 | 4.11 | 4.20 | 0.808 | 0.829 | 0.629 | 0.711 |
| 6 | 3.79 | 4.18 | 0.797 | 0.814 | 0.591 | 0.722 |
| 7 | 3.55 | 3.91 | 0.751 | 0.800 | 0.513 | 0.659 |

Table 6: Average Euclidean distances from the UK Level 7 centre.

| Level | # terms (per 1000) |
|---|---|
| 0 | 0.115 |
| 1 | 0.108 |
| 2 | 0.0996 |
| 3 | 0.0906 |
| 4 | 0.128 |
| 5 | 0.0923 |
| 6 | 0.163 |
| 7 | 0.225 |

Table 7: Average number of everyday Singlish terms per 1000 words.

by including them as inputs to the model or by normalizing the scores against time and other relevant metadata.

Finally, in this project, I limited text examples to top-level posts to avoid the complex and limiting Reddit discourse structure. It would be interesting to extend the investigations done to comments as well as posts.

## 9 Conclusion

In summary, in this project, I look at the linguistic factors that predict the community response of Reddit posts. I collected data from two Reddit subforums, the Singaporean and UK subreddits. Following Bergsma et al. (2012), I extracted three types of features, broadly grouped as grammatical, stylistic and word features. The models generally show good results, with the stylistic and grammatical models performing comparable to state-of-the-art BERT embeddings.

I investigate also the hypothesis that posts conforming to a group's style receive greater community endorsement (Tran and Ostendorf, 2016). I show that in a diglossic situation, although the acrolect draws greater prestige, the most successful posts draw on features from the basilect in order to connect with the audience.

## References


Shane Bergsma, Matt Post, and David Yarowsky. 2012. Stylometric analysis of scientific articles. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 327–337, Montréal, Canada. Association for Computational Linguistics.

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

Hao Fang, Hao Cheng, and Mari Ostendorf. 2016. Learning latent local conversation modes for predicting community endorsement in online discussions.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *CoRR*, abs/1901.05287.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

K. Jones. 2004. A statistical interpretation of term specificity in retrieval. *Journal of Documentation*, 60:493–502.

Himabindu Lakkaraju, Julian McAuley, and Jure Leskovec. 2013. What's in a name? understanding the interplay between titles, content, and communities in social media.

Trang Tran and Mari Ostendorf. 2016. Characterizing the language of online communities and its relation to community reception. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1030–1035, Austin, Texas. Association for Computational Linguistics.

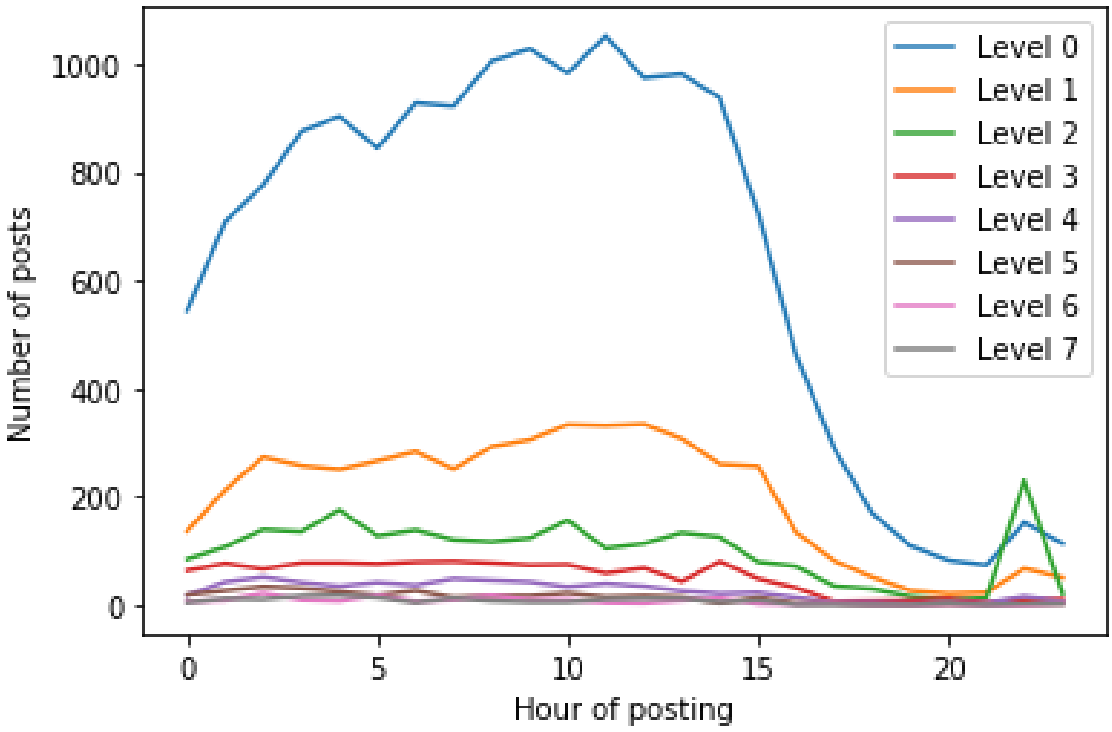


Figure 1: Class distribution against hour of posting for r/singapore

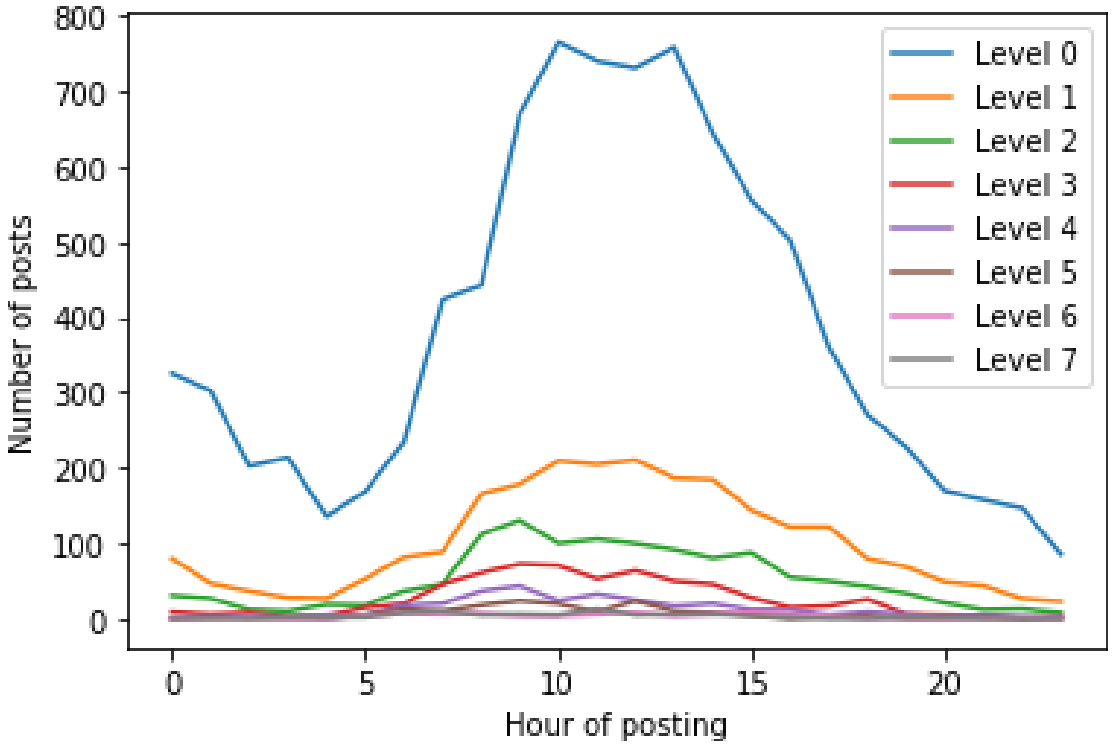


Figure 2: Class distribution against hour of posting for r/UnitedKingdom

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Zhongyu Wei, Yang Liu, and Yi Li. 2016. Is this post persuasive? ranking argumentative comments in online forum. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 195–200, Berlin, Germany. Association for Computational Linguistics.

# A Appendices

## A.1 List of POS tags

```
pos_tags = [
    "ADJ", "ADP", "ADV", "AUX", "CONJ", "CCONJ", "DET",
    "INTJ", "NOUN", "NUM", "PART", "PRON", "PROPN",
    "PUNCT", "SCONJ", "SYM", "VERB", "X", "SPACE",
]
```

## A.2 List of dependency tags

```
dep_tags = [
  "acl", "acomp", "advcl", "advmod", "agent", "amod",
  "appos", "attr", "aux", "auxpass", "case", "cc", "ccomp",
  "clf", "compound", "conj", "cop", "csubj", "csubjpass",
  "dative", "dep", "det", "discourse", "dislocated", "dobj",
  "expl", "fixed", "flat", "goeswith", "intj", "iobj",
  "list", "mark", "meta", "neg", "nn", "nmod", "npadvmod",
  "npmod", "nsubj", "nsubjpass", "nummod", "oprd", "obj",
  "obl", "orphan", "parataxis", "pcomp", "pobj", "poss",
  "predet", "preconj", "prep", "prt", "punct", "reparandum",
  "quantmod", "relcl", "ROOT", "vocative", "xcomp", "",
]
```

## A.3 List of Singlish words

```
sg_words = [
  'abuden', 'act blur', 'agak', 'ai', 'aiya', 'alamak', 'ang mo',
  'ang moh', 'atas',
  'bao toh', 'barang', 'bo', 'bodoh',  'bojio', 'boliao', 'botak',
  'chao', 'chee bai', 'chim', 'cheem', 'chio bu', 'chiong', 'chope',
  'gahmen',
  'heng', 'huat',
  'jialat', 'jio',
  'kena', 'kiasu',
  'la', 'lah', 'lao', 'leh', 'lepak', 'liao', 'liddat',
  'mafan', 'mah', 'meh',
  'paiseh', 'ps', 'paktor',
  'sabo', 'sia', 'sian', 'siao', 'simi',
  'tahan',  'ulu',
  'wa', 'walao', 'wayang',
  'ya', 'yah',
]
```