Slang detection and identification

Zhengqi Pei[1,2] **Zhewei Sun**[3] **Yang Xu**[3]

[1]Engineering Science Program, University of Toronto
[2]HAETEK Institute of Machine Intelligence, Shenzhen, China
[3]Department of Computer Science, University of Toronto
zhengqi.pei@mail.utoronto.ca
{zheweisun, yangxu}@cs.toronto.edu

Abstract

The prevalence of informal language such as slang presents challenges for natural language systems, particularly in the automatic discovery of flexible word usages. Previous work has explored slang in terms of dictionary construction, sentiment analysis, word formation, and interpretation, but scarce research has attempted the basic problem of slang detection and identification. We examine the extent to which deep learning methods support automatic detection and identification of slang from natural sentences using a combination of bidirectional recurrent neural networks, conditional random field, and multilayer perceptron. We test these models based on a comprehensive set of linguistic features in sentence-level detection and token-level identification of slang. We found that a prominent feature of slang is the surprising use of words across syntactic categories or syntactic shift (e.g., verb→noun). Our best models detect the presence of slang at the sentence level with an F1-score of 0.80 and identify its exact position at the token level with an F1-Score of 0.50.

1 Introduction

Slang, or 'the language of streets' (Green, 2015), is a type of informal language consisting of words and expressions shared within specific groups. A hallmark of slang is its expressivity, instantiated in the flexible use of words. For example, the word *sick* with the conventional sense of "ill" can also denote a positive slang sense of "awesome", such as "the band's album is sick". The expressive nature of slang exemplifies its social function, because it provides an effective way of communicating and knowledge-sharing within groups of distinct social identities, such as in the cases of vulgar tongue (Green, 2015) and online language. On the other hand, the flexible nature of slang use can be intriguing for language users, learners, and natural language systems. Here, we ask whether slang can be automatically detected in natural sentences, and what linguistic features might distinguish slang usage from conventional language use.

Our problem statement is simple: Given a natural sentence such as "the band's album is sick", can machines learn to 1) detect whether slang usage is present or not (i.e., sentence-level detection), and 2) identify the exact position of the slang term in the sentence (i.e., token-level identification). For each of these tasks, our systems should be able to learn to cope with two main categories of slang usage (Dhuliawala et al., 2016):

- **Newly extended senses**: existing words in the lexicon that develop novel slang senses distinct from their conventional senses, e.g., *clutch* refers to "an act of grasping" in its conventional usage, but is later extended to the slang sense of "tense critical situation".

- **Newly created words**: words that do not exist in the standard lexicon, e.g., blending of *friend* and *enemy* forms the slang word *frenemy* that describes a person who is simultaneously friend of and in conflict with someone.

Research on slang in the natural language processing community falls under several categories, but to our knowledge the current work is the first to tackle the basic problem of automatic slang detection and identification.

2 Related computational work

2.1 Slang dictionary construction and sentiment analysis

Existing approaches such as SlangNet (Dhuliawala et al., 2016), SlangSD (Wu et al., 2018), and SLANGZY (Gupta et al., 2019) have focused on efficiently maintaining and extending the construction of slang dictionaries to aid computational

Proceedings of the 23rd Conference on Computational Natural Language Learning, pages 881–889
Hong Kong, China, November 3-4, 2019.

sentiment analysis of slang content. Some popular systems from this line of research are based on modular representation (Pal and Saha, 2013) that treats slang in terms of various linguistic stages, each of which deals with slang word from different aspects, e.g., sound, concept, formation, etc. These dictionary-based methods rely on static lexical information and structure, which are typically not sufficient to capture the flexible semantics and lexical coinage in natural slang usages.

2.2 Slang word formation and interpretation

An independent line of work has explored generative models (Kulkarni and Wang, 2018) for slang word formation that captures processes such as blending, clipping, and reduplication. This work uses long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) models to generate slang words in terms of string components according to type-sensitive characteristics. Related work has also explored automatic interpretation of non-standard English words and phrases using sequence-to-sequence architecture with dual encoders (Ni and Wang, 2017). This method generates literal interpretations for queried non-standard expressions from source sentences, but the primary focus is on explanation as opposed to detection or identification, both of which are prerequisite tasks for slang interpretation.

Differing from these existing research, we present a methodological framework based on standard techniques in deep learning for automatic slang detection and identification that does not rely heavily on dictionary construction. We examine a comprehensive set of linguistic features that might be diagnostic of slang usage in natural settings, and we explore existing methods that leverage bidirectional LSTM with multilayer perceptron (MLP) (Rauber and Berns, 2011) and conditional random field (CRF) (Lafferty et al., 2001). Our framework is related to existing work that applies sequence-to-sequence models with attention mechanism (Luong et al., 2015) for the identification of dialectal varieties (Jurgens et al., 2017) and feature-based emotion detection from online media (Ileri and Karagoz, 2016). However, our emphasis here is on learning features that are relevant to the automatic discovery of slang usage.

To preview the framework, our models formulate slang detection and identification as a sequence-labelling task. In addition to typical word embedding inputs, we incorporate relevant linguistic features in the input via an efficient feature boosting procedure. Throughout our experiments, we found that the flexibility of Part-of-Speech (POS) feature is most diagnostic of slang usage: Slang often entails structured POS transformation of existing syntactic uses of words. We show how features related to POS confidence and POS shift in the input provide the improvement on model performance. We also demonstrate how novel tokens of slang can be discovered using a character level convolutional neural network (Zhang et al., 2015).

3 Computational methodology

We present the models and features we use for machine detection and identification of slang.

3.1 Specification of predictive tasks

In the *slang detection* task, our models determine whether a given sentence contains at least one slang usage, which can be an existing word with a novel slang meaning or a newly created word. We formulate this as a binary classification task.

In the *slang identification* task, our models identify each token within the input sentence as 'non-slang' or 'slang' by sequence labeling, which determines the exact positions of slang usage. Note that the models in the identification task encapsulate the detection task; an empty prediction that labels all tokens as 'non-slang' is equivalent to classifying the sentence as a non-slang sentence in the detection task, and vice versa.

3.2 Model architectures

We present a BiLSTM-MLP model that is capable of identifying slang words in a given sentence. The basic architecture is shown in Figure 1. Fully connected MLP layers are placed on top of both the forward and backward hidden states H_f and H_b of a biLSTM network encoding the input sentence. The output of the two MLP layers f_L and b_L are then concatenated as an input to the final MLP layer. In total, there are three components $W^{(f)}_{MLP}$, $W^{(b)}_{MLP}$, $W^{(con)}_{MLP}$ within the MLP block that are shared across all hidden states:

$$f_L = \sigma(W^{(f)}_{MLP} H_f + b^{(f)}) \quad (1)$$

$$b_L = \sigma(W^{(b)}_{MLP} H_b + b^{(b)}) \quad (2)$$

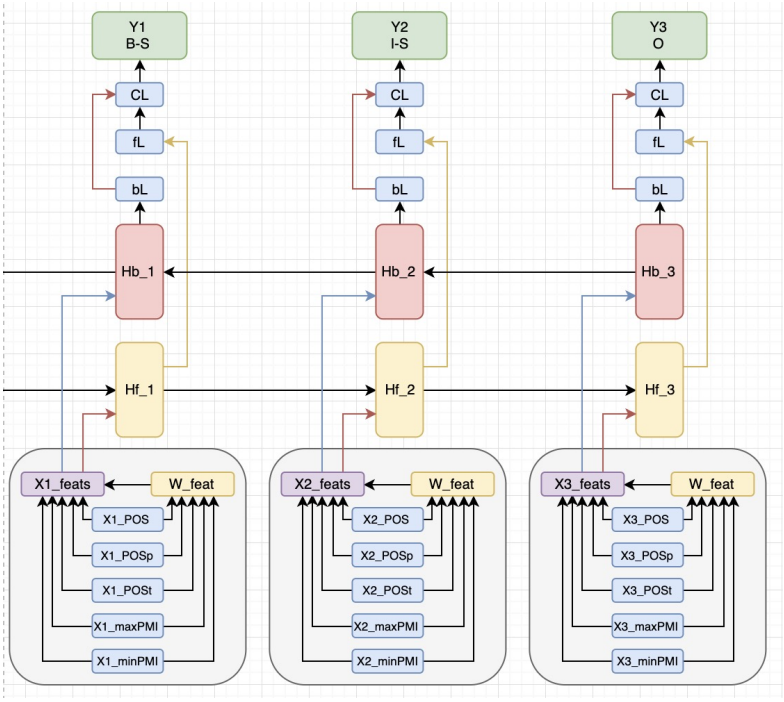

Figure 1: **BiLSTM-MLP model architecture with feature boosting.** The architecture for bidirectional LSTM with Multilayer Perceptron (MLP) as the output layer. The shared bL layer is the MLP unit for the current backward hidden state; the shared fL layer is the MLP unit for forward hidden state; the shared CL layer takes as input the concatenation of the outputs from fL and bL, and output the predictive tagging distribution.

$$C_L = \sigma(W_{MLP}^{(con)}[f_L; b_L] + b^{(con)}) \quad (3)$$

$$Y = softmax(W^{(Y)}C_L + b^{(Y)}) \quad (4)$$

The resulting output vectors are subsequently used to compute the predictive tag for input tokens. There are seven tags here: 'START', 'END', 'O', 'B-U', 'I-U', 'B-N', 'I-N'. These tags apply 'BIO' convention (Ramshaw and Marcus, 1999) that labels non-target token as 'O', initial token of the interested region (e.g., phrase) as 'B-', and the subsequent intermediate tokens of interest as 'I-', etc.

The MLP block in BiLSTM-MLP model can be swapped with an alternative conditional random field (CRF) (Lafferty et al., 2001) that better considers explicit sequential restrictions, e.g., the tag 'I-U' has to be placed after a 'B-U' tag. This sequential restriction can be captured via combination of an LSTM network and a CRF network. The CRF layer has a state transition matrix $A_{(i,j)}$ that models the transition score between the i-th tag and the j-th tag, and an emission matrix $f_{(i,k)}$ that models the output score for the i-th tag at the k-th word (Huang et al., 2015). The source sentence $X_{(i)}$ along with a sequence of tags $Y_{(i)}$ is evaluated via a CRF score:

$$S(X, Y) = \sum_{t=1}^{T}(A_{Y_{t-1},Y_t} + f_{Y_t,X_t}) \quad (5)$$

The CRF layer uses output states from BiLSTM layer to find tags in sequence with optimal CRF score to make prediction. (i.e. $X = [H_f; H_b]$) The CRF probability is easily computed in favor of the logarithmic predictive score as follows:

$$log(P(Y|X)) = S(X, Y) - log(\sum_{Y' \in Y_x} exp(S(X, Y'))) \quad (6)$$

Analogous to Huang et al. (2015), we apply dynamic programming during training to handle the intractable summation term. This BiLSTM-CRF model aims to identify the exact position of each slang word, in terms of sequential restrictions.

3.3 Linguistic features

We use a comprehensive set of linguistic features to facilitate interpretable learning from the models described. Carefully curated linguistic features can improve the training efficiency because linguistic knowledge helps to rectify the learning

process. Feature-based inputs support mapping between contextual concepts and domain-specific clues via distributed representations. The following linguistic features are stored with unique entries in the lookup tables, and they are encoded into embeddings via distributed representation. Figure 2 illustrates these features for the example token *fire*.

Unigram. We take each individual word as an input to the models. The words are represented by standard multi-dimensional word vectors obtained via embedding models such as word2vec (Mikolov et al., 2013).

Bigram. Similar to unigram embedding, a bigram embedding represents the word vector for a bigram. For instance, given an arbitrary word W_t at time-step t in its source sentence, the bigrams for W_t are $W_{t-1}W_t$ and W_tW_{t+1}, which correspond to forward bigram and backward bigram respectively. Whereas the unigram embedding X_t is identified as the word vector representation for W_t, the bigram embeddings are defined as their concatenations. The forward bigram embedding fB_t represents the vector $[W_{t-1};W_t]$, and the backward bigram embedding bB_t represents the vector $[W_t;W_{t+1}]$. Note that both forward and backward bigrams are implemented using the identical lookup table.

Pointwise Mutual Information. We consider measurement of discrepancy between two linguistic variables via PMI (Aji and Kaimal, 2012). Given two source words W_i and W_j, the PMI between them is computed as follows:

$$PMI(W_i, W_j) = log\frac{Pr(W_i, W_j)}{Pr(W_i)Pr(W_j)} \quad (7)$$

We estimate the probabilistic distributions from the Penn Treebank (Marcus et al., 1993), where the probabilities of the PMI can be computed based on co-occurrence statistics. In our models, we compute PMI of the current word with each of its neighboring words, and encode the resulting maximum and minimum PMIs as the features. For example, given $PMI(W_i, W_{i-1}) = 0$ and $PMI(W_i, W_{i+1}) = 2$, we would have the PMI features related to the current word W_i as $maxPMI = 2$ and $minPMI = 0$.

Part-of-Speech. We consider Part-of-Speech (POS) that represents a word's syntactic category. Common POSs include noun, verb, adjective, adverb, pronoun, preposition, conjunction, interjection, and numeral. Multiple POSs can be assigned to an identical word due to the possibility of a word having distinct grammatical properties in different sentences, e.g., *work* is a verb in "these models work", but it is a noun in "I don't like this work". We use Natural Language Toolkit (NLTK) (Loper and Bird, 2002) for POS tagging.

POSp. This linguistic feature is an accessory feature to the POS feature that only represents the grammatical property for the token in its current semantic context. Since each word token might have multiple POS tags, it is possible to count a word's POS distribution that represents the probabilities of a word attached with this specific POS tags as an additional linguistic feature. For example, given a well-formed text corpus C:

$$Pr(POS(like) \leftarrow verb|C) = 0.8$$
$$Pr(POS(like) \leftarrow noun|C) = 0.1$$
$$Pr(POS(like) \leftarrow numeral|C) = 0.0$$

POSt. Word class transfer is a common mechanism (e.g., in English) for extending word senses. We consider a novel feature that represents the transformation from the root-POS (the most commonly used POS for the current token) to the current-POS for the token, e.g., "IN-VB" is a POSt feature, where "IN" is the root-POS, "VB" is the current-POS of the token.

Bigram-Count. The Bigram-Count is similar to the $POSp$, except that the Bigram-Count represents the probability that the current word is collocated with its neighboring words. Given an arbitrary word token W_i in a sequence ϕ, the forward and backward Bigram-Counts are evaluated as

$$fBC = log(\frac{Pr(W_{i-1}W_i)}{Pr(W_i)}) \quad (8)$$

$$bBC = log(\frac{Pr(W_iW_{i+1})}{Pr(W_i)}) \quad (9)$$

The Bigram-Counts are also similar to PMI except that the Bigram-Counts focus more on the current word. In some cases, the Bigram-Count will leverage the zero PMI of low-frequent collocations.

3.4 Feature boosting: Feature-level learning

We present feature boosting for the models with limited features to learn feature-level knowledge. The linguistic features are assumed to be related in terms of the concatenated form of input vectors. For an input token, the related features can

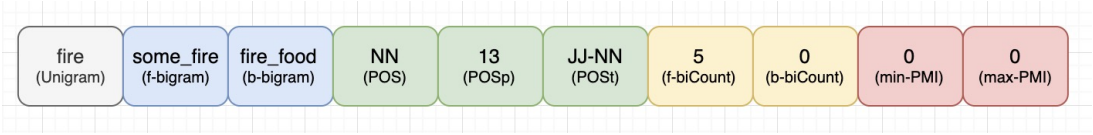

Figure 2: **Concatenation of linguistic features for a given token.** For a specific token *fire* in the source sentence "she can cook some fire food", the related linguistic features are represented as token vectors to concatenate the feature-based input for this token. Each randomly initialized vector is updated during training. The unigram features are represented as 32-dimensional word vectors, the bigram vectors are 20-dimensional, and all else are 16-dimensional.

be assigned with distinct weights that selectively focus on specific features. Suppose we have a token x represented by k distinct linguistic features $[f_1^{(x)}...f_k^{(x)}]$, where $f_i^{(x)} \in \mathbb{R}^{|V_i|\times 1}$, and a shared multi-layer perceptron W_{feat} across all the states. The local feature weights are defined as:

$$\alpha_{feat}{}^{(x)} = W_{feat}[f_1^{(x)}...f_k^{(x)}] + b_f \qquad (10)$$

Where b_f is the MLP bias, and $\alpha_{feat}{}^{(x)}$ is k-dimensional vector that assigns distinct weights to each feature. To provide feature-level information for the vectors fed as inputs to the BiLSTM layer, the feature vectors are weighted in terms of the computed $\alpha_{feat}{}^{(x)}$ for concatenation:

$$V_x = [f_1^{(x)} \cdot \alpha_{feat_1}{}^{(x)}...f_k^{(x)} \cdot \alpha_{feat_k}{}^{(x)}] \qquad (11)$$

The concatenation of weighted feature embeddings contains global feature-level information, allowing the inputs to selectively feed into the model in terms of the feature distribution. As an alternative, the last propagated hidden states from both forward and backward layers of the BiLSTM can be concatenated with the raw features $f_i^{(x)}$ to compute the local feature weights.

We demonstrate later the relative importance of different features in a feature ablation analysis where we remove less relevant input features and keep only the light-weighted but informative linguistic features such as Part-of-Speech and POSp.

3.5 Treatment of novel slang word forms

Our models are able to handle novel tokens by learning the contextual structure within a sentence. Although all the out-of-vocabulary tokens are consistently labelled as 'UKT' such that they are truly unseen by the models, the sequential relations can be captured by the hidden layers of LSTMs. In order to improve model predictability on unknown tokens, we apply character-based convolutional neural network to encode the spelling of words.

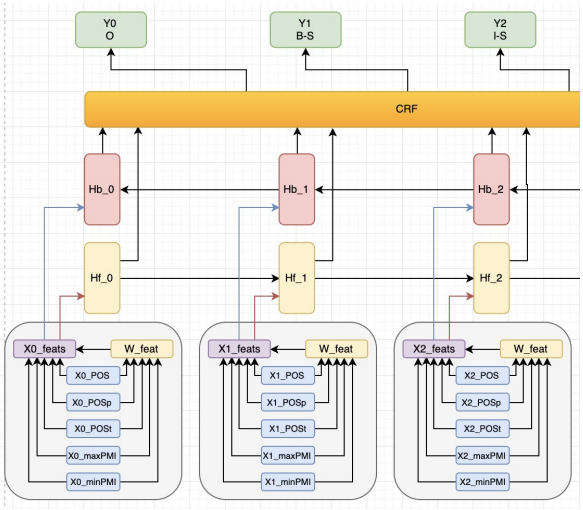

Figure 3: **BiLSTM-CRF model architecture with feature boosting.** The architecture for bidirectional LSTM with Conditional Random Field as the output layer. The shared CRF layer takes as input the BiLSTM's outputs from hidden states, then finds the optimal path in terms of sum of emission scores and transition scores. The optimal path results in the prediction.

Each character is represented by a fixed dimensional embedding (Zhang et al., 2015), similar to word embeddings, and forwarded into a convolutional neural network to obtain a character-level encoding of the word. The resulting Char-CNN embeddings are concatenated with the original input embeddings that feed into the models.

4 Experiments and results

4.1 Experimental setup

We consider datasets that are composed of sentences in two distinct categories, standard (slang-less) and slang-specific:

- **Slang-less sentence dataset**: 15-thousand non-slang sentences from Wall Street News (2011-2016) in Penn Treebank (Marcus et al., 1993) as the negative examples.

- **Slang-specific sentence dataset**: 15-thousand sentences that contain slang words from Online Slang Dictionary (http://onlineslangdictionary.com/) as the positive examples.

The sentences from Wall Street News are taken to be non-slang sentences since the news-based sentences were typically standard English conformed and reviewed before publication. In order to construct an even more trustworthy negative set for standard English, we filtered the sentences from Wall Street News based on the proportion of unknown tokens within the sentences. A News sentence will be considered as an eligible non-slang sentence if it has at most 20% words that have not been covered by the provided frequency-based vocabulary. The vocabulary (or lexicon) consists of top 25,000 most frequent English words from an authoritative text corpus, e.g., Penn Treebank. On average, each negative example sentence contains 12.11 (mean) tokens with standard deviation of 2.52.

We collect positive examples from lexical entries in the Online Slang Dictionary (OSD) where example usage sentences are available. We obtain the ground-truth slang usage positions from OSD and apply the BIO tagging scheme, which labels interested tokens with "B-" at the beginning token, and with "I-" at the subsequent tokens. All the tokens out of interest are labelled as "O". There are two kinds of slang types: UKT-slang and Normal word slang, labelled with "U" and "N", respectively. The UKT-slang refers to slang usages with novel word forms, while normal word slang refers to slang usages with existing words. Out of the 15,000 positive examples, 10,000 of which contain UKT slang words that are not covered by the lexicon. On average, each positive example sentence contains 13.79 (mean) tokens with standard deviation of 3.42.

All models are trained using the Adam optimizer (Kingma and Ba, 2015) with a learning rate 0.001 with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

4.2 Evaluation and results

We evaluate our models in terms of slang detection and identification. We also perform feature ablation to locate salient features of slang usage and show example cases of model success and failure.

4.2.1 Detection task

We evaluated our models to determine whether a given sentence contains at least one slang usage. The evaluation metrics are precision, recall, and F1-score. Table 1 summarizes the results from the models. Overall, all our proposed models performed substantially better than the baseline, with the CRF-based models yielding better performance than the MLP-based models. For instance, although all models tend to have high precision and relatively lower recall, the CRF-based models generally achieve better recall than the MLP-based models given the same level of precision. Importantly, the best overall model makes use of all linguistic features and yields an F1-score close to 0.80. This result suggests that the features we proposed contribute critically to both the precision and recall of slang detection. It is worth noting that models with only the POS related features achieve reasonable performance (although not as well as the full model), and we will return to this observation in the ablation analysis.

4.2.2 Identification task

We evaluated our models to identify each word in a given sentence at the word level. The evaluation metrics are precision, recall, and F1-score. Table 2 summarizes the results. Similar to the case of detection, our models performed substantially above the baseline in this task. In particular, both the MLP and CRF-based models yielded higher F1 scores (close to 0.50) when multiple features are taken into account. Tables 3 and 4 further summarize the results (i.e., number of correctly predicted cases) of these models in identifying the two main different types of slang: novel slang word and novel slang sense (of an existing word), and how the models fair with and without the incorporation of CNN character-based embeddings.

4.2.3 Feature ablation

We evaluated the contributions of the linguistic features on the test set via model performance degradation through ablation. We would like to evaluate the extent that the model performance would be degraded with respect to a single feature (e.g., POS), given a trained model with the complete featured set. In this case we would force all the POS embeddings to be zero-vectors, and we then compare the ablated model against the full model. Figure 4 summarizes the degradation of

Model (features)	Precision	Recall	F1-score
Random Guess (baseline)	0.5000	0.5000	0.5000
BiLSTM-MLP (POS+POSp)	**0.9893**	0.4806	0.6469
BiLSTM-MLP (POS+POSp+POSt+PMI)	0.9777	0.5651	0.7162
BiLSTM-MLP (POS+POSp+POSt+PMI boosting)	0.9771	0.6053	0.7475
BiLSTM-MLP (full features)	0.9433	0.6842	0.7931
BiLSTM-CRF (POS+POSp)	0.9873	0.5969	0.7440
BiLSTM-CRF (POS+POSp+POSt+PMI)	0.9749	0.6482	0.7787
BiLSTM-CRF (POS+POSp+POSt+PMI boosting)	0.9749	**0.6496**	**0.7797**
BiLSTM-CRF (full features)	0.9518	**0.6856**	**0.7971**

Table 1: Model comparisons in the slang detection task.

Model (features)	Precision	Recall	F1-score
Random Guess (baseline)	0.0263	0.4834	0.0498
BiLSTM-MLP (POS+POSp)	**0.6240**	0.3172	0.4206
BiLSTM-MLP (POS+POSp+POSt+PMI)	0.6172	0.3864	0.4753
BiLSTM-MLP (POS+POSp+POSt+PMI boosting)	0.5967	0.3975	0.4771
BiLSTM-MLP (full features)	0.5423	**0.4612**	**0.4985**
BiLSTM-CRF (POS+POSp)	0.5666	0.3712	0.4485
BiLSTM-CRF (POS+POSp+POSt+PMI)	0.5763	0.4183	0.4847
BiLSTM-CRF (POS+POSp+POSt+PMI boosting)	0.5954	**0.4280**	**0.4980**
BiLSTM-CRF (full features)	0.5499	0.4501	0.4950

Table 2: Model comparisons in the slang identification task.

Model (features)	New Word	New Sense
BiLSTM-MLP (POS+POSp)	194/523	35/199
BiLSTM-MLP (POS+POSp+POSt+PMI)	228/523	53/199
BiLSTM-MLP (POS+POSp+POSt+PMI boosting)	236/523	**53/199**
BiLSTM-MLP (full features)	**267/523**	66/199
BiLSTM-CRF (POS+POSp)	227/523	41/199
BiLSTM-CRF (POS+POSp+POSt+PMI)	251/523	50/199
BiLSTM-CRF (POS+POSp+POSt+PMI boosting)	**260/523**	50/199
BiLSTM-CRF (full features)	242/523	**83/199**

Table 3: Model comparisons on identified slang by type (either as new word or existing word with new sense).

Model	Identification F1-score	Detection F1-score
BiLSTM-MLP	0.5101	0.8649
BiLSTM-CRF	0.5024	0.8679
BiLSTM-MLP with Char-CNN	**0.5172**	**0.8693**
BiLSTM-CRF with Char-CNN	0.5146	0.8679

Table 4: Comparisons of model performance with and without character-based embedding.

model performance based on ablation of each individual feature in question.

Salience of Part-of-Speech transformation. Based on the feature ablation analysis, Part-of-Speech features are the most crucial to overall model performance. Bigram counts come next which suggests that syntacmatic relations also play a role in slang usage. We probed the most prominent features by dividing the POS transformations observed in the data into two kinds: homogeneous and heterogeneous transformations. POSt features such as "VV-VV" (i.e. the identified

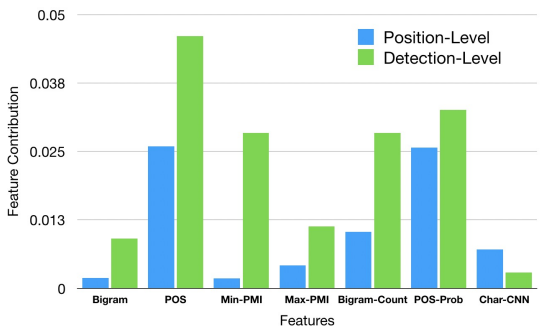

Figure 4: **Summary of feature ablation analysis.** The contributions of the individual linguistic features are shown. The degradation in performance is considered to be equivalent to a feature's contribution.

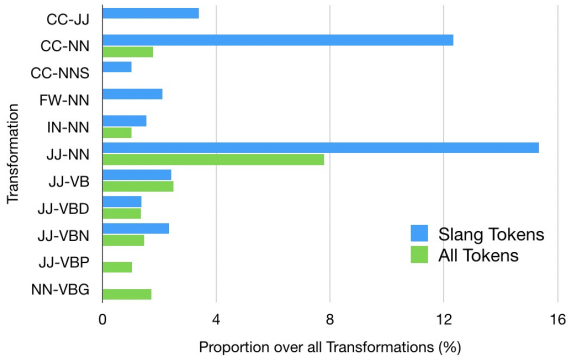

Figure 5: **Transformation of Part-of-Speech in slang usage.** Heterogeneous POS transformations (POSt) that have proportions higher than 1% are shown. There are 11 distinct POSt, 9 of which correspond to cases where the POSt proportion of slang word usage is higher than that of common word usage (i.e., control set). Both the slang tokens and normal tokens with JJ (adjective) tend to transfer to NN (noun); the slang words with CC (coordinating conjunction) are more likely to tranfer to NN (noun).

POS given the source sentence is identical to the root POS) are considered homogeneous transformations; the heterogeneous POSt refer to the case when the target POS differs from the root POS. The proportion of heterogeneous POSt over all the transformations is 25.74% among all the tokens, while the heterogeneous proportion surprisingly increases to 53.94% in slang-specific tokens. This indicates that a slang word is twice as likely to experience (heterogeneous) POS transformation in comparison to an arbitrary word, providing evidence that syntactic shift is a salient feature of slang usage. A comparison between POSt distributions of slang-specific and ordinary use cases is shown in Figure 5.

4.2.4 Examples of model success and failure

We provide examples of both successful and failed predictions to demonstrate the model capability in slang detection and identification:

- **Probe sentence**: "That money you sent me was clutch."
- **Model prediction**: ["clutch"]
- **Ground truth**: ["clutch"]

In the probe sentence, the token *clutch* refers to tense critical situation (noun) rather than its common sense "grasping" (verb). Our model successfully detected this slang component in the query.

- **Probe sentence**: "That's a real blower."
- **Model prediction**: ["–"] (no slang detected)
- **Ground truth**: ["blower"]

The token *blower* normally refers to a device that produces a current of air in common usage, while it refers to "surprise" in this probe sentence. Our model failed to detect this slang-component possibly due to insufficiency of contextual information.

5 Discussion

We take an initial step at automatic detection and identification of slang from natural sentences using established deep learning methods. We show how linguistic features combined with deep learning algorithms offer interpretability. We find that the bidirectional LSTM with feature-based inputs and character-based convolutional embeddings using multilayer perceptron yield the best performance in position identification, and the model with similar mechanisms except with conditional random field has better performance in detecting whether a source sentence contains a slang term. For unknown tokens, character-based convolutional embeddings improve the model in handling novel slang terms. We demonstrate that features combined with distributed word embeddings help machine detection of slang in general, and that Part-of-Speech among others is a prominent feature of slang usage. Our work provides a basis for locating slang from its flexible and unconventional syntactic word uses and offers opportunities for slang processing in downstream tasks in natural language processing.

References

Subhanpurno Aji and Ramachandra Kaimal. 2012. Document summarization using positive pointwise mutual information. *International Journal of Computer Science Information Technology*, 4:47–55.

Shehzaad Dhuliawala, Diptesh Kanojia, and Pushpak Bhattacharyya. 2016. SlangNet: A WordNet like resource for English slang. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4329–4332, Portorož, Slovenia. European Language Resources Association (ELRA).

Jonathon Green. 2015. *The vulgar tongue: Green's history of slang*. Oxford University Press, USA.

Anshita Gupta, Sanya Bathla Taneja, Garima Malik, Sonakshi Vij, Devendra K. Tayal, and Amita Jain. 2019. Slangzy: a fuzzy logic-based algorithm for english slang meaning selection. *Progress in Artificial Intelligence*, 8(1):111–121.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *Arxiv Preprint*.

Ibrahim Ileri and Pinar Karagoz. 2016. Detecting user emotions in twitter through collective classification. *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*.

David Jurgens, Yulia Tsvetkov, and Dan Jurafsky. 2017. Incorporating dialectal variability for socially equitable language identification. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Vivek Kulkarni and William Yang Wang. 2018. Simple models for word formation in slang. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, LA, USA. ACL.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Ke Ni and William Yang Wang. 2017. Learning to explain non-standard english words and phrases. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP 2017)*, Taipei, Taiwan. AFNLP and ACL.

Alok Ranjan Pal and Diganta Saha. 2013. Detection of slang words in e-data using semi-supervised learning. *International Journal of Artificial Intelligence and Applications*, 4(5):4961.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

T. Rauber and K. Berns. 2011. Kernel multilayer perceptron. In *2011 24th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 337–343.

Liang Wu, Fred Morstatter, and Huan Liu. 2018. Slangsd: Building, expanding and using a sentiment dictionary of slang words for short-text sentiment classification. *Lang. Resour. Eval.*, 52(3):839–852.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 649–657, Cambridge, MA, USA. MIT Press.