Simple Models for Word Formation in English Slang

Vivek Kulkarni
Department of Computer Science
University of California, Santa Barbara
vvkulkarni@cs.ucsb.edu

William Yang Wang
Department of Computer Science
University of California, Santa Barbara
william@cs.ucsb.edu

Abstract

We propose generative models for three types of extra-grammatical word formation phenomena abounding in English slang: *Blends, Clippings, and Reduplicatives*. Adopting a data-driven approach coupled with linguistic knowledge, we propose simple models with state of the art performance on human annotated gold standard datasets. Overall, our models reveal insights into the generative processes of word formation in slang – insights which are increasingly relevant in the context of the rising prevalence of slang and non-standard varieties on the Internet.

1 Introduction

Linguistic analysis of slang has traditionally received little attention with some arguing that research on slang be assigned to an *"extra-linguistic darkness"* (Labov, 1972). However, Eble (2012) argues that the emergence of social media has predominantly increased the usage of slang and non standard forms and *"slang is now worldwide the vocabulary of choice of young people"*[1]. This increasing pervasiveness has recently motivated research on slang and the linguistic phenomena it manifests. Most notable are the works of (Mattiello, 2005, 2008, 2013), who argues that slang exhibits extra-grammatical properties that distinguish it from the standard form.

Specifically, linguistic phenomena like *alphabetisms*, *blending*, *clippings*, and *reduplicatives* abound in slang (see Table 1)[2]. Note the rich and varied word formation patterns ranging from simple abbreviations like dink to more complex combinations like lambortini, a blend of lamborghini and martini. Note further, that

Word	Derived From	Type
dink	double income no kids	alphabetism
lambortini	lamborghini + martini	blend
diamat	dialectical + materialism	blend
tude	attitude	clipping (fore)
brill	brilliant	clipping (back)
yik-yak	yik	reduplicative

Table 1: Sample words depicting word formation in slang. Note the rich variation across different types.

even within a particular class like BLENDS there are variations in what portions of the components are retained.

These word formation mechanisms are not only attractive from a linguistic standpoint in deepening our understanding of slang but also have applications spanning the development of rich conversational agents and tools like brand name generators (Özbal and Strapparava, 2012). While such phenomena have been qualitatively studied by Mattiello (2008, 2013), computational models for their generation have not been proposed.

In this paper, we propose the first simple models for generating blends, clippings, and reduplicatives[3]. Our models incorporate linguistic insights coupled with data-driven analysis to model the above phenomena. In line with "Occam's razor", we strive for simplicity. The simplicity of our models not only implies better generalization, more robust estimation of parameters in the wake of small dataset sizes, and better interpret-ability but also yields state of the art performance.

Specifically, we show that by exploiting structural constraints, blend formation can be modeled as a simple *sequence labeling* problem as opposed to prior work which models it as a *general sequence to sequence* problem. This view enables the use of

---

[1]While the definition of slang is a controversial issue, we adopt a broad definition including non-standard expressions.

[2]While such phenomena are likely present in several languages, in this work we restrict ourselves to slang in English.

[3]While there has been some prior work on computationally modeling blends, generative models for clippings and reduplicatives have not been outlined.

a simple LSTM model to yield competitive performance. Similarly, we propose the first probabilistic generative models for clippings and reduplicatives effectively incorporating phonetic constraints. In a nutshell, our contributions are:

1. **Generative models.** We propose simple models for generating blends, clippings and, reduplicatives with state of the art performance.

2. **Linguistic Insights.** We reveal linguistic insights into these phenomena which we incorporate into the generative models.

3. **Resources.** We release all our models and the compiled datasets to aid further research [4].

2 Datasets and Definitions

Here, we define the extra-grammatical morphological phenomena modeled and describe the datasets used for our experiments and analysis. Mattiello (2013) argues that slang exhibits extra-grammatical morphological properties that distinguish them from the standard variety and identified four broad word formation phenomena[5] described below:

1. **Alphabetisms** are shortenings of a multi-word sequence. Examples include lol from laugh out loud or YOLO from you only live once. They can be further sub-categorized into two types based on their pronunciation although the distinction may not always be clear: (a) *Acronyms* are pronounced using the regular reading rules (for example. YOLO) (b) *Initialisms* are pronounced letter by letter (for example. BBC).

2. **Blends** or portmanteaus, are formed by merging parts of existing words. For example, edutainment is a blend of education and entertainment. Prior work notes that blend formation does not exhibit rigid rules but only demonstrates affinities towards certain patterns of formation (Mattiello, 2013) suggesting learning based approaches to modeling blends (Deri and Knight, 2015; Gangal et al., 2017).

3. **Clippings** are constructed by shortening words (lexemes). For example, berg is a clipping of iceberg, gym is a clipping of gymnasium and ammo is a clipping of ammunition. Based on the portion that is being clipped, clippings are sub-categorized into three types: (a) BACK clipping where the beginning of the word (lexeme) is retained (like brill from brilliant) (b) a FORE clipping, where the end of a word is retained (like choke from artichoke) and (c) A COMPOUND clipping (adman), a clipping of a compound word (advertisment man).

4. **Reduplicatives** are word pairs constructed by either repeating a word (boo boo) or by alternating certain vowels or consonants so that they are phonologically similar (clickety-clackety, teenie-weenie, itsy-bitsy).

In our work, we propose generative models using a data-driven approach towards generating blends, clippings ,and reduplicatives. We do not consider generative models for alphabetisms since a majority of them can be trivially generated by picking the first letter of each word making up the acronym (for example. laugh out loud → lol). We now outline the datasets considered:

1. **Blends.** We consider a gold standard dataset $\mathcal{D}_{knight}$ of 400 blends constructed by (Deri and Knight, 2015) from Wikipedia as well as a larger list of 1624 blends manually compiled by (Gangal et al., 2017) called $\mathcal{D}_{large}$, a superset of $\mathcal{D}_{knight}$. We define $\mathcal{D}_{blind} = \mathcal{D}_{large} - \mathcal{D}_{knight}$.

2. **Clippings.** We consider a list of 576 human curated clippings constructed by Mattiello (2013) for our analysis of clippings. These were manually collected from a variety of sources including prior work and dictionaries like the *Oxford English Dictionary* and the *Merriam-Webster online dictionary*.

3. **Reduplicatives.** We consider a dataset of a total of 337 reduplicatives manually constructed by Mattiello (2013) also collected from prior work and online dictionaries[6].

---

[4]Code and data with reproducible results is available at https://github.com/viveksck/simplicity

[5]These categories are not exhaustive and several slang words and non-standard expressions do not fall into any of the above categories.

[6]We note that the dataset released by (Mattiello, 2013) is the largest human curated dataset of slang word formation we are aware of. Furthermore, large crowd sourced data-sets like Urban Dictionary are a poor fit for such modeling since they do not explicitly contain annotations by linguistic experts.

3 Models and Methods

We now describe our proposed models to generate blends, clippings, and reduplicatives. We precisely formulate the problem, specify our models and comprehensively outline our evaluation.

3.1 Blends

Problem Formulation Given a string $C = C_1 \# C_2$, consisting of components C_1 and C_2, we seek to combine them to yield the blend B. For example, given $C_1 = \texttt{brad}$ and $C_2 = \texttt{angelina}$ such that $C = \texttt{brad\#angelina}$, we seek to generate the blend $\texttt{brangelina}$.

Existing Models (Deri and Knight, 2015) proposed a model to generate blends using multi-tape finite state transducers. Most recently, (Gangal et al., 2017) (the current state of the art) model this as a general sequence to sequence learning problem and propose a neural encoder-decoder architecture with attention to outperform the model by (Deri and Knight, 2015). However, this model fails to effectively exploit the inherent structure and linguistic constraints of blending. One implication is an exploration of an overly complex hypothesis space with a small amount of training data making it harder to generalize. A second implication is that the decoding phase uses exhaustive generation. In fact, the best model *exhaustively* generates all candidate strings, where the first part is a prefix of C_1 and the second part is a suffix of C_2 and scores them to pick the best candidate while using a backward model learned to generate the components given the blend. In contrast, we propose a more straightforward model that explicitly incorporates inherent linguistic constraints entirely obviating the need for decoding using exhaustive candidate generation yet yielding competitive performance.

3.1.1 Linguistic Insights into Blend Formation

While one can model the problem of learning to blend as a variable length sequence to sequence learning problem (akin to machine translation), we argue that incorporating structural constraints yields a different view of modeling the problem that can enable better generalization given the small amount of training data. We motivate this by observing the following constraints:

1. **Blend length and vocabulary constraints.** First, we observe that a majority of blends
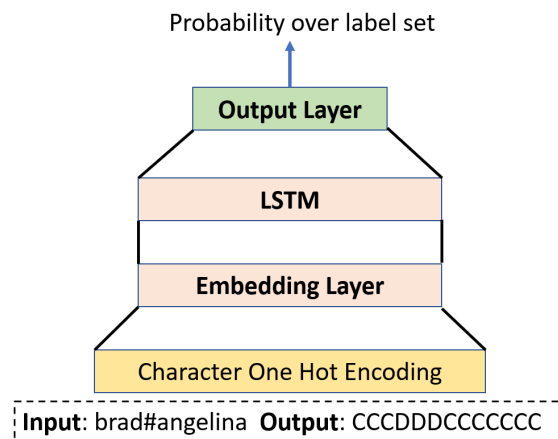

Figure 1: Our simple LSTM sequence labeling model for blends. Our model is considerably simpler both in complexity of the parameter space and in terms of implementation compared to the encoder-decoder model proposed by (Gangal et al., 2017) which modeled the problem as a machine translation problem.

(99.0% in D_{knight} and 92.4% in D_{blind}) are formed by using only characters present in the original components. Second, the length of the blends in these cases is at-most the length of the components.

2. **Fixed length input output representation.** The blend B can thus be encoded as a string of the exact same length as C by noting that B only contains characters copied from C or deleted from C. Specifically, we represent B by $E(B)$ denoting the sequence of *copy* (**C**) and *delete* (**D**) operations needed to transform C to B. $E(B)$ can easily be computed by the edit distance function between C and B. For example, $\texttt{brangelina}$ is encoded as $\texttt{CCCDDDCCCCCCC}$. Since $E(B)$ has the same length of C, we can now model this as a *fixed length sequence labeling* problem rather than a variable length sequence to sequence learning problem completely obviating the need for the *"encoder-decoder"* architecture for this large class of blends.

Equivalent Problem Definition Given a string $C = C_1 \# C_2$, consisting of components C_1 and C_2, learn $E(B)$, a labeling of each character in C from the label set $\{\mathbf{C}, \mathbf{D}\}$.

3.1.2 Proposed model: COPYCAT

Model Architecture Inline with work on neural sequence labeling (Wang et al., 2015; Plank et al., 2016), our model uses a single layer long

short term memory (LSTM) (Graves et al., 2012) and an embedding layer as depicted in Figure 1. Since the size of the training set is relatively small, we reduce the number of learn-able parameters by using pre-trained character embeddings frozen during training. In particular, we use 50 dimensional character embeddings from (Gangal et al., 2017) obtained by training an LSTM language model on an English Dictionary. The implementation of the LSTM layer is described by (Graves et al., 2012; Wang et al., 2015) and therefore omitted here. The output layer is a soft-max over the label set $\{\mathbf{C}, \mathbf{D}\}$.

Candidate Generation Our model outputs probability scores over the label set for each element in the sequence. As in previous works (Gangal et al., 2017), we use the output to generate an ordered candidate set $\mathcal{T}$. To construct $\mathcal{T}$, we use a simple top-K decoder which selects the k most probable label sequences. Finding the k most probable tag sequences from the soft-max outputs can be cast as finding the top k shortest simple paths in a directed acyclic graph which can be efficiently solved using (Yen, 1971; Eppstein, 1998). Note that the greedy decoder which just picks the most likely label at each position is a special case with $k = 1$. While the number of candidates generated by (Gangal et al., 2017) depends on the size of C, our model generates a constant number of candidates ($k = 5$) regardless of the input C.

Candidate Ranking and Selection While the list of candidates in $\mathcal{T}$, can be used to make a prediction we note that re-ranking these candidates can result in better performance and thus consider multiple ranking strategies:

1. **LSTM.** We consider only the scores as obtained by the LSTM with no re-ranking.

2. **LSTM + LM.** We augment the score of each candidate to include both the score of the LSTM as well as its score according to a character level language model where the language model is trained on a large amount of unsupervised text [7].

3. **LSTM + LM + LEN.** Figure 2 shows a least squares fit to the length of the blends versus the length of its component, suggesting a strong correlation between these two variables. We capture this notion through a

---

[7]We use words from the CMU pronouncing dictionary.

probabilistic model. Specifically, we model $\mathbf{Pr}(Blend_{len}|Component_{len})$ by fitting a Bayesian Ridge Regression model to the training data and score each candidate on this model as well. Finally, we combine the scores obtained for the LSTM, the language model and the length model uniformly to yield the final score for each candidate.

In each of the above cases, we pick the topmost candidate as our prediction.
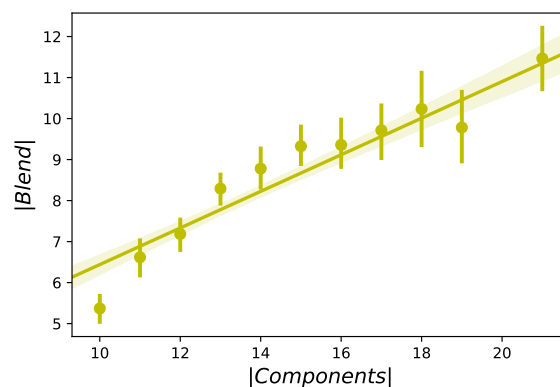

Figure 2: Best fit between blend length and the length of the input string, implying that one can infer the length of the blend from the length of the components using a regression model.

Table 2 illustrates the effects of each of these ranking strategies on an exemplary set of strings. While ranking using the raw scores of the LSTM yields blends which are close, observe that the LSTM alone does not capture ease of pronunciation effectively. For example, the top ranking candidate `brngelina` is relatively unlikely under both a character language model and a phoneme language model. Incorporating scores from the language model results in much more natural blends like `brangelina`. To observe the effect of the length model, note that for `kentucky#indiana` the blends obtained by LSTM and LSTM + LM are too short (`keiana` and `keana`). Incorporating the length model boosts scores for candidates closer to the target length yielding `keniana` which is closer to the target `kentuckiana`.

3.1.3 Evaluation

We compare our model to previous methods namely (Deri and Knight, 2015) and (Gangal et al., 2017). Inline with (Gangal et al., 2017), we evaluate our model on $\mathcal{D}_{knight}$ and $\mathcal{D}_{blind}$ (consisting of 1078

Input	LSTM	LSTM + LM	LSTM + LM + LEN	Gold standard
brad#angelina	brngelina	brangelina	brangelina	brangelina
animated#matrix	animtrix	animatrix	animatrix	animatrix
merkel#sarkozy	merkrkozy	merkkozy	merkkozy	merkozy
dramatic#drastic	draastic	drastic	dramastic	dramastic
kentucky#indiana	keiana	keana	keniana	kentuckiana
employability#agility	emplglity	emplgility	employgility	employagility

Table 2: Set of examples demonstrating the effect of incorporating the language model and length model when ranking candidates revealing insights into blend formation. Incorporating the language model generally improves the fluency of the blend while the length model helps generate blends with lengths close to the target.
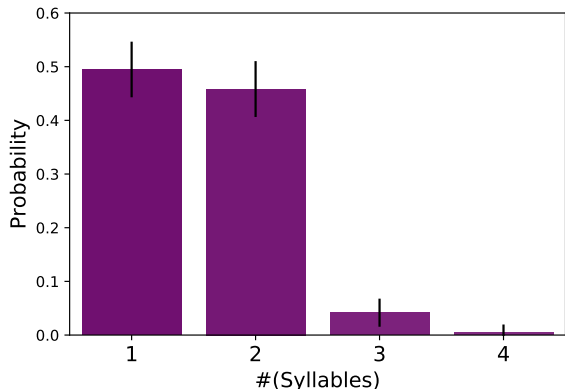

Figure 3: Distribution of the number of syllables in clippings. Most clippings have at-most 2 syllables but it is a challenge to infer whether a given word has a one or two syllable clipping.

instances)[8]. For evaluating on $\mathcal{D}_{knight}$, we use 10-fold cross-validation. For evaluating on $\mathcal{D}_{blind}$ we train our model on $\mathcal{D}_{knight}$ and report the mean score on the test-set obtained using 10 random splits of the training data. As in previous work, our metric is the edit distance between the predicted blend and the true blend.

3.2 Clippings

Problem Formulation Given a word w, learn a model that can generate its clipping c. For example, given the word `administration` we would like the model to output the clipping `admin`.

Proposed Models We motivate our models by presenting two insights into linguistic properties of clippings first noted by (Mattiello, 2013). First, most clippings are back clippings while fore clippings are relatively rare. Second, most clippings tend to have at most two syllables (see Figure 3). The first insight guides our model in determining whether to retain a prefix or a suffix of the original word. The second insight guides our model in

determining how much to retain (or clip). In particular, we can capture this in two ways: (a) Working in the phoneme space and (b) Learn a function to predict the length of the clipping (which encodes the number of syllables implicitly).

3.2.1 CLIPPHONE

CLIPPHONE operates by mapping the word w to a sequence of phonemes, explicitly clipping in the phoneme space and mapping the phoneme sequence back to the grapheme space. In particular, the model can be described as follows: **(1)** Let θ and π represent multinomial distributions over clipping types and the number of syllables respectively. **(2)** Represent the word w as a sequence of phonemes $\mathcal{P}$ and identify each syllable. **(3)** Draw a sample l from π to represent the number of syllables in the clipping and the type t by drawing a sample from θ. **(4)** If $t \in \{\text{BACK}, \text{FORE}\}$, clip $\mathcal{P}$ to have exactly l syllables by selecting the appropriate length prefix or suffix depending on clipping type t to make exactly l syllables represented by $\mathcal{P}_{clip}$. If t is a COMPOUND, clip each word recursively and concatenate the outputs. **(5)** Map P_{clip} back to grapheme space to yield the clipping. **(6)** The parameters of θ and π, both multinomial distributions can be estimated from observed data via maximum likelihood estimators.

Phoneme-based representation Given w, we obtain its phoneme representation using the pre-trained state-of-art neural model G2PSEQ2SEQ (Yao and Zweig, 2015)[9]. For example, the word `captain` is mapped to the following sequence of phonemes $\mathcal{P}$ given by `K AE P T AH N`.

Clipping in the phoneme space We identify the syllable boundaries in $\mathcal{P}$ by looking for vowel phonemes and clip $\mathcal{P}$ until it contains the desired number of syllables. For example, a one syllable clipping of `K AE P T AH N` is `K AE P T`

---

[8]We discard relatively rare instances with insertions.

[9]G2PSeq2Seq has a PER of 5.45% on CMUDict.

since AH is the beginning of the second syllable ($\mathcal{P}_{clip}$ = K AE P T).

Mapping clipping back to graphemes Finally, given the clipped phoneme sequence $\mathcal{P}_{clip}$, we map it back to the grapheme space by learning a sequence to sequence model to map a phoneme sequence to its grapheme sequence. We follow the same model architecture as used in G2PSEQ2SEQ (Yao and Zweig, 2015) but just flip the input and output sequences, thus learning a model to map phonemes back to graphemes. Finally, we quantify the effectiveness of this model explicitly in our evaluation by establishing an upper-bound on the expected performance using this model.

3.2.2 CLIPGRAPH

As we will demonstrate empirically, CLIPPHONE poses the following challenges: Since a given phoneme sequence can map to multiple grapheme sequences, explicitly mapping to phoneme space and back introduces errors and loss of fidelity [10]. This is compounded by noting that whether w's clipping should have one or two syllables is hard to predict (since both occur in almost equal proportions empirically, see Figure 3). Furthermore, deciding whether the clipping should end in a vowel or a consonant and determining the length is yet another challenge. As an alternative, we propose CLIPGRAPH where we seek to directly learn a model to predict the length of the clipping directly given w, obviating the need to work in phoneme space. CLIPGRAPH works as follows: **(1)** Let θ represent multinomial distributions over clipping types and let π be a model that predicts the length l of the clipping w_{clip} given w. **(2)** Draw a sample from θ to get the type of clipping t. **(3)** Retain the appropriate l-length prefix or suffix of w based on the type t handling compound words recursively. **(4)** We use Ridge regression to learn π from the training data and estimate θ using its maximum likelihood estimator.

Evaluation To evaluate our models, we consider a naive baseline NAIVE which clips w to one of its prefixes or suffixes randomly. For CLIPPHONE, we also consider one (CLIPPHONE-1SYL) and two (CLIPPHONE-2SYL) syllable clippings. Since (Jamet, 2009) notes that determining whether a word has a one or two syllable clipping is extremely challenging, we also consider a model with an ora-

---

[10]K AE P T can be mapped to capped or capt.

cle (CLIPPHONE(O)) on the number of syllables in the clipping to quantify this. Finally, we establish an upper bound (G2P-GOLD) on the performance of any method using our learned P2G model. G2P-GOLD maps the gold standard G to phoneme space and maps the resulting phoneme sequence back to yield $\hat{G}$ as the predicted clipping. We use the edit distance between the predicted clipping and the gold standard as the evaluation metric.

3.3 Reduplicatives

Problem Formulation Given a word v, we seek to generate a word w such that $v.w$ is a reduplicative. For example, given the word flip, we would like to generate flop or flap to yield the reduplicatives flip-flop or flip-flap.

Proposed Model We motivate our model from a linguistic standpoint by referring to observations by (Mattiello, 2013) that a large fraction of reduplicatives can be formed by either (a) Duplicating the word DUPLICATE (boo-boo) (b) Exchanging the initial vowel VOWELEX (bing-bong) and (c) Exchanging the initial consonant CONEX (teenie-weenie). Other patterns include adding a consonant (artsy-fartsy) or adding schm/shm (moodle-schmoodle). In our work, we propose a generative model for the three dominant forms of reduplication mentioned above. Broadly, our model captures the notion that vowels and consonants display strong replacement preferences. For example, the vowel i is much more likely to be replaced with a than u (instances like clip-clap, wishy-washy). Similarly the consonant t is much more likely to be replaced by w (as in teenie-weenie, tinky-winky). We incorporate these insights into our generative model as follows: **(1)** Let θ be a distribution over the three different types of reduplicatives. Let ϕ_v, ψ_c be distributions over letters that replace vowel v and consonant c respectively. **(2)** Sample the type of reduplicative t generated from θ. **(3)** If t is DUPLICATE, set w to v and return w as the reduplicative component. **(4)** If t is VOWELEX, find the first vowel x with non-zero replacement probability. Sample the replacement z from ϕ_x and replace x with z. If t is CONEX, find the first consonant c with non-zero replacement probability and sample the replacement z from ψ_c. Replace c with z. **(5)** Return the edited string as the second component of the reduplicative. **(6)** The parame-

ters of multinomial distributions θ, ϕ_v, and ψ_c are estimated via MLE estimates from the data.

Evaluation We consider two baseline models (a) LET Uniformly replace a letter with another letter in v to return w. (b) **LET(COND)** Uniformly replace a letter (vowel or consonant) with a letter from its class. Since reduplicatives are characterized by phonologically similar sounds, merely using edit distance as a metric for evaluation would be ineffective. For example, even the ill-rhyming `flip-flsp` has the same edit distance as the correct reduplicative `flip-flop`. Thus, we use a distance measure (MIR) defined over the phoneme space (Hixon et al., 2011) which effectively captures the similarity of two phonetic sequences by modeling the affinities between pairs of phonemes using a point access mutation matrix and is superior over metrics like PER (phoneme error rate).

4 Experiments

Having described our models and evaluation metrics in the previous section, we proceed to evaluate our models empirically and describe our results.

Blends We train our model on $\mathcal{D}_{knight}$ and evaluate on the $\mathcal{D}_{blind}$ dataset as in (Gangal et al., 2017) comparing against previous methods (Deri and Knight, 2015; Gangal et al., 2017). We set the number of hidden units to 50 with a dropout probability of 0.5[11]. We use ADAM (Kingma and Ba, 2014) optimizer with an initial learning rate of 0.001 to train the model for 500 epochs with early stopping over a validation set.

Tables 3 and 4 show the mean edit distance of our predictions from the target blend. First, the evaluation on the $\mathcal{D}_{knight}$ dataset compares the performance of our models against previous baselines. Note that just using our basic model COPYCAT -(LSTM + LM) outperforms the baseline proposed by (Deri and Knight, 2015) (1.59 vs 1.49). Furthermore, observe that even our vanilla model (LSTM) significantly outperforms the equivalent "FORWARD" models by (Gangal et al., 2017)[12] that use greedy and beam-search decoding (1.90 and 2.37 vs 1.75). Moreover, observe that our model even achieves almost equivalent performance to the "FORWARD" state-of-art model which uses

---

11 All hyper-parameters were chosen using a validation set.
[11] All hyper-parameters were chosen using a validation set.
[12] We obtain the numbers for these baselines from the latest released data and predictions at https://github.com/vgtomahawk/Charmanteau-CamReady.

exhaustive decoding (1.37 vs 1.33). Furthermore, our model achieves competitive performance with the more complex models proposed by GANGAL-BACKWARD which use exhaustive decoding. We emphasize that we can achieve competitive performance using a simpler model without using exhaustive decoding. Similar observations can also be made for the evaluation of the $\mathcal{D}_{blind}$ data set (our best model yields a score of 1.91 vs 1.77). Altogether these observations suggest that even simple models with effective modeling of linguistic structure can perform competitively and even outperform overly complex models (see Table 6 for a few example predictions).

Model	Distance
KNIGHT	1.59
GANGAL-FORWARD (GREEDY)	1.90
GANGAL-FORWARD (BEAM)	2.37
GANGAL-FORWARD (SOTA)†	1.33
GANGAL-BACKWARD (SOTA)†	1.12
COPYCAT-(LSTM)	1.75
COPYCAT-(LSTM + LM)	1.49
COPYCAT-(LSTM + LM + LEN)	**1.40**
COPYCAT-(LSTM + LM + LEN)†	**1.37**

Table 3: 10-fold cross validation performance of our blending model COPYCAT in terms of edit distance (lower is better) on $\mathcal{D}_{knight}$ dataset. † indicates ensemble approach using sub-samples of training data consistent with previous work. Our simpler model yields competitive performance (especially compared to the state of art forward model) without the need for exhaustive decoding (which the state of art uses), uses a smaller learn-able parameter set while effectively using linguistic insights into the blending process.

Clippings We consider the dataset of clippings introduced by (Mattiello, 2013) and report the mean edit distance (μ) on a set of 173 clippings in Figure 4 (see Table 5 as well). First, both CLIP-PHONE ($\mu = 3.39$) and CLIPGRAPH ($\mu = 2.65$) outperform the naive baseline ($\mu = 4.6$). Second, based on the empirical distribution, it is almost impossible to predict whether a word has a one or two syllables (0.49 vs 0.46 significant only at $\alpha > 0.2$) clipping and incorrect guesses critically affect the downstream performance. In the absence of such information, just clipping to one syllable yields better performance. However, when this information is exactly known (CLIPPHONE(O)), we note an improvement as expected ($\mu = 2.79$). Finally, CLIPGRAPH shows a small but not signif-

Model	Distance
KNIGHT	2.10
GANGAL-FORWARD (SOTA)[†]	1.76
GANGAL-BACKWARD (SOTA)[†]	1.77
COPYCAT-(LSTM)	2.27
COPYCAT-(LSTM + LM)	2.13
COPYCAT-(LSTM + LM + LEN)	**1.98**
COPYCAT-(LSTM + LM + LEN)[†]	**1.91**

Table 4: Performance of our blending model COPY-CAT in terms of edit distance (lower is better) on $\mathcal{D}_{blind}$ dataset. † indicates ensemble approach using sub-samples of training data consistent with previous work. Our simpler model yields competitive performance without the need for exhaustive decoding, uses a smaller learn-able parameter set while effectively using linguistic insights into the blending process. To ensure fair comparison, numbers for the baselines were obtained by filtering the released predictions for these models to the same set of words our models were evaluated on.

icant advantage over methods working explicitly in the phoneme space which are prone to errors by imperfect conversion. Finally, the upper bound on the performance is substantially better than our models ($\mu = 0.79$) suggesting scope for future improvements.

| Input | Our models | | Gold |
	CLIPPHONE	CLIPGRAPH	Clipping
cocaine	coke	coca	coke
juvenile	juve	juve	juvey
amelia	umm	amel	mel
alfred	fred	fred	fred
kid video	kidvid	kivid	kidvid

Table 5: Exemplary predictions from clippings models. We can generate one/two syllable clippings, as well as compound clippings. Note the effect of incorrect P2G conversion for `amelia` as `umm` which is pronounced similar to `ame`. The current G2P model does not incorporate stress. It is possible that incorporating stress into the model can address this scenario.
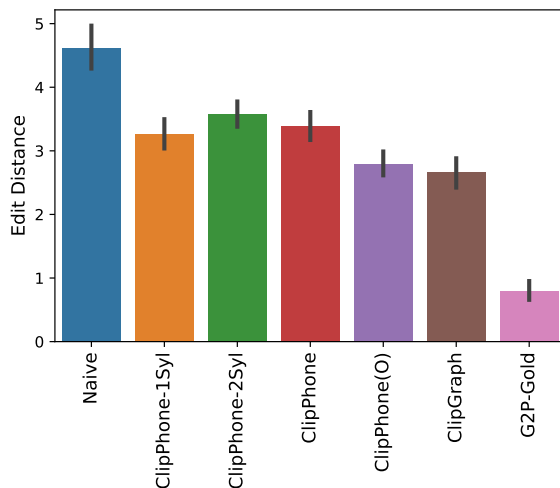

Figure 4: **Evaluation of clipping models** (lower is better). The NAIVE model is the performance lower bound. Both CLIPPHONE and CLIPGRAPH substantially outperform the baseline. Estimating whether a given word has one or two syllable clipping is a major challenge hindering CLIPPHONE since both cases are equally likely from empirical estimation. Using an oracle on the number of syllables (CLIPPHONE(O)) improves performance. CLIPGRAPH which operates purely in grapheme space performs as well as CLIP-PHONE(O). G2PGOLD denotes a upper bound when using our P2G model.

Input	Prediction	Gold	Dist
jizz#disney	jizzney	jizzney	0
scum#fuzz	scuzz	scuzz	0
phone#neck	phoneck	phoneck	0
woman#romance	womance	womance	0
piss#mishap	pisshap	pisshap	0
pass#asshole	passhole	passhole	0
spike#angel	spangel	spangel	0
man#amazon	mamazon	manazon	1
awkward#sauce	awkwauce	awksauce	1
bed#orgasm	bergasm	bedgasm	1

Table 6: Exemplary predictions from our simple blends model which suggests our model effectively captures blending phenomena by incorporating linguistic constraints.

Reduplicatives We evaluate our model on a held-out test set of 50 reduplicatives obtained using the manually compiled dataset by (Mattiello, 2013). We evaluate two flavors of our model: (a) OUR(NODUP) where we disallow generating duplicates (which are trivial to generate) and (b) OUR, the full-fledged model where duplicate reduplicatives are allowed. We report the mean MIR for each model over 10 independent runs in Figure 5. Our model consistently outperforms the baselines (LET,

and LET(COND)) by at-least 8 percentage points suggesting it adeptly captures patterns in reduplicative formation. Finally, we examine the inferred probability distributions to gain insights into the linguistic phenomena in reduplicative formation few of which we outline: (a) The most common reduplicative types are DUPLICATE, VOWELEX followed by CONEX. (b) Vowel i is more likely to be replaced by a and o and (c) Consonant t is much more likely to be replaced by w and l (like

in `teenie-weenie`). We leave a comprehensive analysis of these patterns to future work.
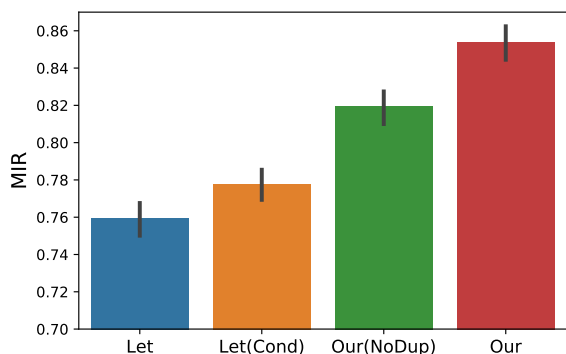

Figure 5: Performance of our reduplicative models based on the MIR metric (higher is better). Our models OUR(NODUP) and OUR consistently outperform baselines by at-least 8 percentage points (0.86 vs. 0.78).

5 Related Work

Blends, clippings and reduplicatives have been studied from a linguistic standpoint (Thun, 1963; Murata, 1990; Merlini Barbaresi, 2008; Hladky, 1998; Hamans, 1997; Fandrych, 2008; Moehkardi, 2016; Ungerer, 2007; Beal, 1991; Algeo, 1977; Smith et al., 2014; Shaw et al., 2014; Beliaeva, 2014; Broad et al., 2016; Renner et al., 2013; Gries, 2004a,b). Most relevant is the work of (Mattiello, 2005, 2008, 2013) who argues that slang is pervasive on the Internet, suggests its extra-grammatical nature and outlines some phonological and morphological properties. Specifically, these phenomena are discussed followed by a qualitative analysis on a manually compiled dataset of 1580 words from various sources. Recently (Deri and Knight, 2015) and (Gangal et al., 2017) study the problem of learning to blend and derive data-driven computational models for the task. (Deri and Knight, 2015) propose a model based on finite state transducers. (Gangal et al., 2017) outperform (Deri and Knight, 2015) by modeling the problem of generating blends as a variable length sequence to sequence learning problem and propose a neural encoder-decoder based model. We differ from all of these works in several ways. In contrast to the blending model proposed by Gangal et al. (2017), our model is simpler with lesser parameters, does not require using an encoder-decoder framework, or exhaustive decoding and yet yields competitive performance on a large class of blends. We also propose the first computational generative data-driven

models for clippings and reduplicatives while capturing phonetic similarity as well and evaluate our models quantitatively.

6 Conclusion

We proposed generative models for blends, clippings, and reduplicatives, three dominant word-formation phenomena in slang. Our models are distinguished by their simplicity, adept use of linguistic and structural constraints, easy to implement and yield state of the art performance.

Our work suggests several directions for future research. First, our blending model can be extended to handle relatively rare insertions, incorporate the language model and the length model in a unified reinforcement learning framework optimizing a joint reward. Second, we do not investigate the complementary problem of de-blending. Moreover, we note that our evaluation of blends is based on an assumed gold standard. It would be useful to also characterize our blending model based on a human evaluation. Third, the gap between the performance of our clipping model and the upper bound (by the oracle) opens up the question of developing more nuanced models for clipping perhaps using deeper linguistic cues like stress patterns. Fourth, our models do not incorporate relatively rare types of reduplicative formation (like *schm* reduplicatives) suggesting yet another direction for research. Yet another open question is whether a global model can effectively model all the above phenomena. Finally, in our work, we focus on only developing models for word formation in English slang. However, such word formation patterns are also evident in other languages (Štekauer et al., 2012) and it is an open question as to whether similar models generalize to other languages as well.

Finally, our work potentially enables the development of several applications some of which include brand name generators, and rich conversational agents that are not only passive agents but can actively contribute to the evolution of language varieties.

Altogether our work has implications for the broader fields of Internet Linguistics and natural language understanding especially in the context of slang formation.

References

John Algeo. 1977. Blends, a structural and systemic view. *American speech* 52(1/2):47–64.

Joan Beal. 1991. Toy boys and lager louts: Motivation by linguistic form. In *Language usage and description: Studies presented to NE Osselton on the occasion of his retirement*. pages 139–148.

Natalia Beliaeva. 2014. Unpacking contemporary english blends: Morphological structure, meaning, processing .

Rachel Broad, Brandon Prickett, Elliott Moreton, Katya Pertsova, and Jennifer L Smith. 2016. Emergent faithfulness to proper nouns in novel english blends. In *Proceedings of WCCFL*. volume 33, pages 77–87.

Aliya Deri and Kevin Knight. 2015. How to make a frenemy: Multitape fsts for portmanteau generation.

Connie Eble. 2012. *Slang and sociability: In-group language among college students*. UNC Press Books.

David Eppstein. 1998. Finding the k shortest paths. *SIAM Journal on computing* 28(2):652–673.

Ingrid Fandrych. 2008. Submorphemic elements in the formation of acronyms, blends and clippings. *Lexis. Journal in English Lexicology* (2).

Varun Gangal, Harsh Jhamtani, Graham Neubig, Eduard Hovy, and Eric Nyberg. 2017. Charmanteau: Character embedding models for portmanteau creation. *arXiv preprint arXiv:1707.01176* .

Alex Graves et al. 2012. *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer.

Stefan Th Gries. 2004a. Isnt that fantabulous? how similarity motivates intentional morphological blends in english. *Language, culture, and mind* pages 415–428.

Stefan Th Gries. 2004b. Shouldn't it be breakfunch? a quantitative analysis of blend structure in english. *Linguistics* pages 639–668.

Camiel Hamans. 1997. Clippings in modern french, english, german and dutch. *Trends in Linguistic Studies and Monographs* 101:1733–1742.

Ben Hixon, Eric Schneider, and Susan L Epstein. 2011. Phonemic similarity metrics to compare pronunciation methods. In *Twelfth Annual Conference of the International Speech Communication Association*.

Josef Hladky. 1998. Notes on reduplicative words in english. *Brno studies in English* 24:33–78.

Denis Jamet. 2009. A morphophonological approach to clipping in english. can the study of clipping be formalized? *Lexis. Journal in English Lexicology* (HS 1).

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

William Labov. 1972. Some principles of linguistic methodology. *Language in society* 1(1):97–120.

Elisa Mattiello. 2005. The pervasiveness of slang in standard and non-standard english .

Elisa Mattiello. 2008. *An introduction to English slang: A description of its morphology, semantics and sociology*, volume 2. Polimetrica sas.

Elisa Mattiello. 2013. *Extra-grammatical morphology in English: abbreviations, blends, reduplicatives, and related phenomena*, volume 82. Walter de Gruyter.

L Merlini Barbaresi. 2008. Extra-grammatical morphology: English reduplicatives. *Words in action. Diachronic and synchronic approaches to English discourse. Studies in honour of Ermanno Barisone* pages 228–241.

Rio Rini Diah Moehkardi. 2016. Patterns and meanings of english words through word formation processes of acronyms, clipping, compound and blending found in internet-based media. *Humaniora* 28(3):207–221.

Tadao Murata. 1990. Ab type onomatopes and reduplicatives in english and japanese. *Linguistic fiesta: festschrift for Professor Hisao Kakehis sixtieth birthday* pages 257–272.

Gözde Özbal and Carlo Strapparava. 2012. A computational approach to the automation of creative naming. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, pages 703–711.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. *arXiv preprint arXiv:1604.05529* .

Vincent Renner, François Maniez, and Pierre Arnaud. 2013. *Cross-disciplinary perspectives on lexical blending*, volume 252. Walter De Gruyter.

Katherine E Shaw, Andrew M White, Elliott Moreton, and Fabian Monrose. 2014. Emergent faithfulness to morphological and semantic heads in lexical blends. In *Proceedings of the Annual Meetings on Phonology*. volume 1.

Michael R Smith, Ryan S Hintze, and Dan Ventura. 2014. Nehovah: A neologism creator nomen ipsum.

Pavol Štekauer, Salvador Valera, and Lívia Kőrtvélyessy. 2012. *Word-formation in the world's languages: A typological survey*. Cambridge University Press.

Nils Thun. 1963. *Reduplicative words in English: A study of formations of the types tick-tick, hurly-burly, and shilly-shally*. Uppsala.

1433

Friedrich Ungerer. 2007. Word-formation. In *The Oxford handbook of cognitive linguistics*.

Peilu Wang, Yao Qian, Frank K Soong, Lei He, and Hai Zhao. 2015. Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. *arXiv preprint arXiv:1510.06168* .

Kaisheng Yao and Geoffrey Zweig. 2015. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. *arXiv preprint arXiv:1506.00196* .

Jin Y Yen. 1971. Finding the k shortest loopless paths in a network. *management Science* 17(11):712–716.