Neuramanteau: A Neural Network Ensemble Model for Lexical Blends

Kollol Das
kolloldas@gmail.com
Independent Researcher
Bangalore, India

Shaona Ghosh
sg811@cam.ac.uk
University of Cambridge
Department of Engineering,
Cambridge, UK

Abstract

The problem of blend formation in generative linguistics is interesting in the context of neologism, their quick adoption in modern life and the creative generative process guiding their formation. Blend quality depends on multitude of factors with high degrees of uncertainty. In this work, we investigate if the modern neural network models can sufficiently capture and recognize the creative blend composition process. We propose recurrent neural network sequence-to-sequence models, that are evaluated on multiple blend datasets available in the literature. We propose an ensemble neural and hybrid model that outperforms most of the baselines and heuristic models upon evaluation on test data.

1 Introduction

Blending is the formation and intentional coinage of new words from existing two or more words (Gries, 2004). These are called neologisms. Neologisms effectively trace changing cultures and addition of new technologies. Blending is one way to create a neologism. A lexical blend is formed by combining parts of two or more words. Predicting a high quality lexical blend is often unpredictable due to the uncertainty in the formal structure of blends (Beliaeva, 2014). Merging source words `digital` and `camera`, the common expectation is the blend `digamera`, although, in practice, `digicam` is more common (Beliaeva, 2014). There are multiple unknown factors such as phonology, semantics, familiarity, recognizability and lexical creativity that contribute to blend formation. Some downstream applications that can leverage such blending systems include generating names of products, brands, businesses and advertisements to name a few especially if coupled with contextual information about the business or sector.

Word$_1$	Word$_2$	Blend	Structure	Category	Coverage
aviation	electronics	avionics	avi-onics	Prefix + Suffix	16.74%
communicate	fake	communifake	communi-fake	Prefix + Word	10.78%
speak	typo	speako	speak-o	Word + Letter	0.14%
west	indiea	windies	w-indies	Letter + Word	0.89%
point	broadcast	pointcast	point-cast	Word + Suffix	22.56%
scientific	fiction	scientifiction	scienti-fic-tion	Word + Word overlap	22.56%
affluence	influenza	affluenza	af-fluen-za	Prefix + Suffix overlap	13.98%
brad	angelina	brangelina	br-a-ngelina	Prefix + Word overlap	11.39%
subvert	advertising	subvertising	sub-vert-ising	Word + Suffix overlap	16.31%

Table 1: Sample blends in our dataset along with the type and coverage. There are other types of rare blends that is beyond the scope of this work.

1.1 Related Work

Blends are compositional words consisting of whole word and a splinter (part of morpheme) or two splinters (Lehrer, 2007). The creative neoglogism of blend formation has been studied by linguists in an attempt to recognize patterns in the process that model human lexical creativity or to identify source words and blend meaning, context and influence. With the popularity of deep neural networks (DNN)s (LeCun et al., 2015), we are interested in the question if neural network models of learning can be leveraged in a generative capacity, that can sufficiently explore or formalize the process of blend formation. Blends can be formed in several closely related morphologically productive ways as shown in Table 1. Our work targets blends that are ordered combinations of prefixes of first word and suffixes of the second word. Examples include, `avionics` (prefix + suffix), `vaporware` (word + suffix), `robocop` (prefix + word), `carmageddon` (overlap). Several theories have been forwarded as to the structure and mechanism of blending (Gries, 2012), without much consensus (Shaw et al., 2014). Most

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 576–583,
Taipei, Taiwan, November 27 – December 1, 2017 ©2017 AFNLP

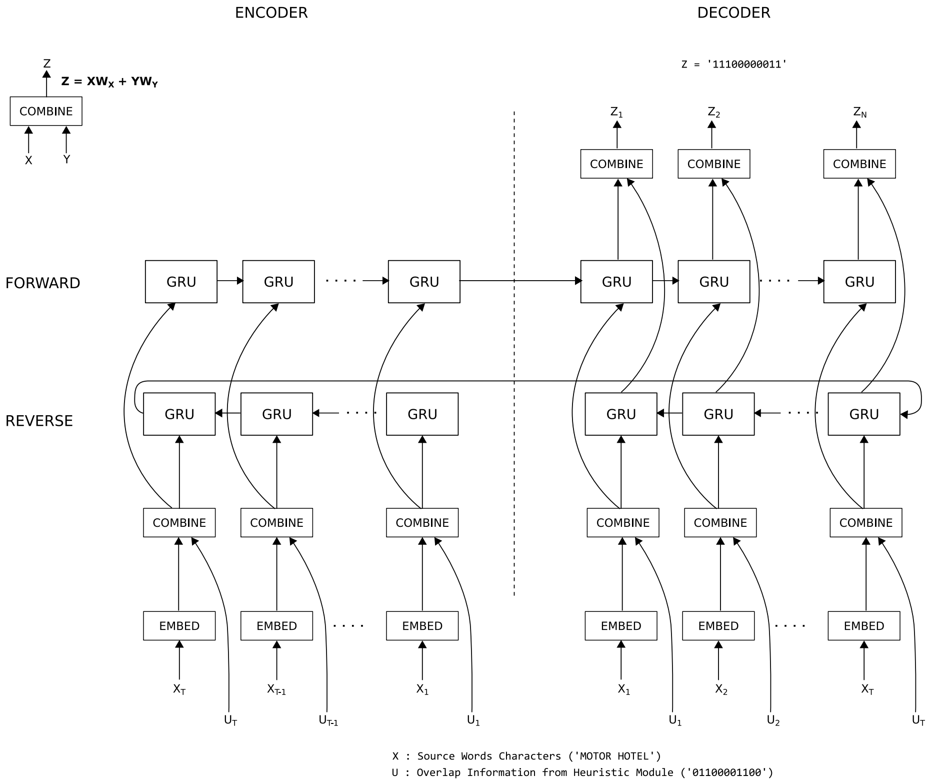

Figure 1: Architectural diagram of the bidirectional hybrid encoder decoder model.

implementation work in blenders and generators is quite sparse in the literature and mainly concern explicit blending of vector embeddings.

Naive implementations of generators exist on the internet [123], that are simply lists of all possible blend combinations. Some other work is based on greedy heuristics for better results [45]. Andrew et. Al (2014) built a statistical word blender as a part of a password generator. Ozbal (2012) uses a combination of edit distances and phonetic metrics and Pilichowski (2013) also uses a similar technique. Our work empirically captures some of these mechanisms from blend datasets used in the literature using neural networks. Numerous studies on deterministic rules for blend formation (Kelly, 1998; Gries, 2004, 2006, 2012; Bauer, 2012) do not find consensus on the blending process mainly due to the ’human factor’ involved in designing rules.

The closest work of using novel multitape Finite State Transducers (FST) for blend creation is in work by Deri et. al. (2015). Our model is a neural model, different from the multitape FST model. The multitape FST model is similar to our baseline heuristic model with which we compare our neural model proposed in this paper. The primary benefit of our model is that it attempts to arrive at a consensus among various neural experts in the generative process of new blend creation.

1.2 Contributions

We summarize the main contributions of our work as follows:

1. We propose an ensemble neural network model to learn the compositional lexical blends from two given source words.

2. We generalize the problem of lexical blending by leveraging the character based sequence-to-sequence hybrid bidirectional models in order to predict the blends.

3. We release a blending dataset and demo of our neural blend generator along with open

[1] http://www.dcode.fr/word-contraction-generator

[2] http://werdmerge.com/

[3] http://www.portmanteaur.com/

[4] https://www.namerobot.com/namerobot/name-generators/name-factory/merger.html

[5] http://www.namemesh.com/company-name-generator

source software

2 Neural Lexical Blending Model

2.1 Sequence-to-sequence Models

Sequence-to-Sequence (Seq2Seq) neural network models (Sutskever et al., 2014), are a generalization of the recurrent neural network (RNN) sequence learning (Cho et al., 2014) paradigm, where a source sequence $x_1, \ldots, x_n$ is mapped to a fixed sized vector using a RNN often serving as the encoder, and another RNN is used to map the vector to the target sequence $y_1, \ldots, y_n$, functioning as the decoder. However, RNNs struggle to train on long term dependencies sufficiently; and therefore, Long Short Term Memory Models (LSTM) and Gated Recurrent Units (GRUs)(Cho et al., 2014) are more common for such sequence to sequence learning.

2.2 Bidirectional Seq2Seq Model

We propose a bidirectional forward and reverse GRU encoder-decoder model for our lexical blends. To this end, both the encoder and the decoder are bidirectional, i.e. they see the ordered input source words both in the forward direction and the backward direction. The motivation here is that in order to sufficiently capture the splinter point, there should be dependency on the neighbouring characters in both directions. Figure 1 shows the bidirectional Seq2Seq model that we propose. Since we have two source words for every blend, that is a sequence of characters, the input to our model has the source words concatenated with a space and padding to align to the longest concatenated example. For example in Figure 1, the source words `work` and `alcoholic` are concatenated as `cilohocla krow` and `work alcoholic` for the encoder and decoder respectively, which subsequently gets reversed to `work alcoholic` and `cilohocla krow` for the reverse encoder and decoder units. The representation of a sequence of characters in the input pair of source words is the concatenation of the fixed dimensional character to vector representations. The model's prediction corresponding to the two source words in the input, is the order preserving binary output $\{\hat{y}_t = \{0,1\} | x_t \in y_t\}$ for model prediction $\hat{y}$, ground truth target y and concatenated source input pair $x = (x_1, ..., x_T)$. For example, if x is `work alcoholic` without padding, the prediction on the blended word is 11110000111111 for the target `workoholic`. The order is enforced implicitly in the concatenated inputs. The indices that are predicted 1 are for the characters in the concatenated input that are included in the blend.

The forward and reverse hidden states h_t^f and h_t^r of the encoder at time t is given by:

$$h_t^f = GRU_{enc}^f \left(h_{t-1}^f, x_{t_r} \right) \tag{1}$$

$$h_t^r = GRU_{enc}^r \left(h_{t+1}^r, x_{t_r} \right) \tag{2}$$

where GRU_{enc}^f and GRU_{enc}^r stand for forward and reverse GRU encoder units described by (Cho et al., 2014) and $t_r = T - t + 1$. Similarly, the forward y^f and reverse y^r intermediate outputs of the decoder are given by:

$$y_t^f = GRU_{dec}^f \left(y_{t-1}^f, x_t \right) \tag{3}$$

$$y_t^r = GRU_{dec}^r \left(y_{t+1}^r, x_t \right) \tag{4}$$

where $y_0^f = h_T^f$ and $y_{T+1}^r = h_1^r$. The final decoder output is the result of applying smooth non-linear sigmoid transformation to the GRU outputs

$$z_t^j = \sigma \left(W_f y_t^f + U_r y_t^r \right) \tag{5}$$

z is character wise probability of inclusion in the blend. We apply sigmoid loss on z for training.

2.3 Ensemble Hybrid Bidirectional Seq2Seq Model

We propose an ensemble model of our vanilla bidirectional encoder decoder model discussed in Section 2.2 in order to capture different representations of the source word, hidden state and a variety of blends. Blends can be inherently varying in structure of the blend formation resulting in multiple possible blend candidates. Each member or an expert in our ensemble model has the same underlying architecture as described in Section 2.2 and in Figure 1. However, the experts in the ensemble do not share the model parameters, the motivation being able to capture a wider range of parameter values thus tackling the variations and multiplicity in the blending process adequately. Further, there is a subjective and qualitative nature of the blend formation process that leads to naturally consider ensemble predictions with the notion that

Dataset	**No. of Examples**	**No. of Overlaps**
Wiktionary	2854	1379
Train	2140 (75%)	
Validation	428 (15%)	
Test	286 (10%)	
Test + Validation with overlaps	319	319
Test + Validation no overlaps	395	0
Cook	230	90
Thurner	482	320
Believa	335	168

Table 2: Datasets for experiments. Blends need not be overlapping even if the source words share common substrings. E.g. puny + unicode = punycode.

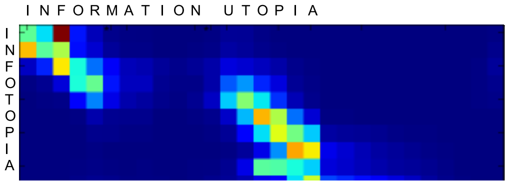

Figure 2: Attention Mask while predicting in attention baseline model.

Model	**Character Accuracy**			**Word Accuracy**		
	Test	*With Overlaps*	*No Overlaps*	*Test*	*With Overlaps*	*No Overlaps*
Baselines						
Conditional	-	-	-	0.192	0.185	0.203
Heuristic Conditional	-	-	-	0.420	**0.806**	0.063
char-to-char						
Vanilla	0.865	0.882	0.850	0.110	0.123	0.077
Split Encoder	0.856	0.871	0.832	0.085	0.120	0.053
Attention	0.912	0.925	0.886	0.280	0.367	0.133
Index based						
Pointer feedforward	-	-	-	0.185	0.197	0.195
Pointer decoder	-	-	-	0.245	0.220	0.250
char-to-binary						
Binary decoder	0.951	0.951	0.949	0.320	0.367	0.330
Bidirectional binary decoder	0.951	0.949	0.948	0.360	0.350	**0.353**
Hybrid binary decoder	0.954	0.964	0.940	**0.430**	0.520	0.313

Table 3: Single instance model results on the Wiktionary Test set. Accuracies are reported at the character and word level. To show fine grained performances the models are additionally tested on two mutually exclusive datasets, one with overlapping blends and the other without. We have run paired t-tests with the baseline heuristic model with p-value of 0.012.

the collective consensus should be more smooth than capturing individual preferences or specialization in the experts where each expert predicts differently. Each expert is independently initialized and parameterized by a GRU based encoder decoder neural network and trained end-to-end simultaneously.

The model makes the final prediction by using a method of combining the expert predictions known as *confidence measure*. Each expert predicts the blended target $z_t^i = [0, 1]$ using Equation 5, for the ith expert such that $i \in K$, where there are K experts. *Confidence* of each expert i with respect to a pre-defined threshold γ is given by:

$$Cf^i = \frac{\sum_{t=1}^{n} \left| z_t^i - \gamma \right|}{n} \tag{6}$$

For the purposes of evaluating the quality of the ensemble model prediction, we perform a confidence weighted voting on the blended words (at a word level instead of character level) predicted by the individual expert and report the top voted blend predictions by the ensemble by selecting the prediction of the expert(s) that has the highest weighted confidence. The accuracy with respect to the ground truth is then evaluated on the test set and reported in the Section on experiments.

We would like to introduce the two baseline models here, as they naturally lead to the hybrid aspect of our model which we discuss later. The conditional baseline model builds a conditional probability distribution of the splice points from source word lengths, i.e. $P(i_1|l_1)$ and $P(i_2|l_2)$, where i_1, i_2 are the splice indices and l_1, l_2 are the lengths of source word 1 and 2 respectively. The model predicts using argmax from the distributions and combines the prefix and suffix. The heuristic model greedily looks for common substrings in the two source words and joins them with the overlap e.g. `group` + `coupon` = `groupon` and `group` + `coupon` = `gron`. Then from the set of combinations, it picks the one that has the longest overlap. If no common substring is found, it reverts to plain conditional model.

The bidirectional binary encoder-decoder that we discussed before does poorly on overlapping blends because it struggles to determine the blends with overlaps. We propose a further enhancement to the ensemble model whereby we introduce extra information about the overlap and common substrings between the source words. This is a hybrid between the heuristic and neural model. Consider the overlap type blend `group` + `coupon` = `groupon`. A mapping is induced from the source words that indicates to the neural network that `oup` is the overlapping segment. This mapping is provided to the hybrid model as additional information in the form of a binary sequence indicating overlaps between the 2 source words so `group` + `coupon` gets mapped to 001110011100 and is fed into the encoder just after the embedding layer as shown in Figure 1. The motivation is that the extra information should take the burden off the model in finding common overlaps.

Effect of dominance To study the effect of dominance of either of the source words we add binary tags to the characters of each source word proportional to the portion of source word in the blend. If the proportion is greater than or equal to 50 percent, then the dominance is set to 1 else 0. Note that both words can have dominance set to 1. Additionally, we set dominance of both the words to 1 if the proportions differ by less than 0.1 percent.

3 Experiments

3.1 Dataset Details

We use modern collection of english blend words curated by Wiktionary[6] with a total of 3250 blends. Each example consists of two source words and a target blend word. We restrict the dataset to only prefixes of first source word and suffixes of second source word leading to a total of 2854 blends. As held out datasets we use the dataset created by Cook et al. (Cook and Stevenson, 2010) from wordspy.com, and collected in 2008. It has 179 samples distinct from our training set and 230 unique blends in total with a good balance among different types of blends.

[7] The Thurner dataset that we use has 482 unique blends which are a subset of the full Thurner dictionary (1993) of around 2300 words collected from 85 sources. The structure of the blends is skewed towards overlapping blends constituting roughly about 66 percentage of all the blends. The Believa dataset (2014) has 235 unique blends collected from multiple sources from year 2000 until recent. The dataset has upto 4 source words but we pick only ones with 2. The blends are well balanced with distributions very similar to Cook (2010).

3.2 Network and Training Details

In this section, we discuss the network structure and training details. All the code is written using Tensorflow (Abadi et al., 2016).

Network Layout: As discussed before we use GRU RNN network as our encoder decoder Seq2Seq model. Our proposed model has single layer GRU units with 128 neurons. The character vocabulary has a size of 40. Our dataset contains blends with numbers, hyphens and also capitals. Larger, multiple layers or more number of experts in the ensemble networks did not improve performance drastically.

Model	Top-1 Word Accuracy					
	Test	*OL*	*No OL*	*Cook*	*Thurner*	*Believa*
Bidirectional Binary (K=60)	0.470	0.495	**0.435**	0.487	0.320	0.373
Hybrid Bidirectional Binary (K=30)	**0.540**	0.677	0.400	**0.578**	0.465	**0.430**
Heuristic Conditional	0.420	**0.806**	0.063	0.375	**0.571**	0.409
Mixture of Experts (K=6)	0.365	0.347	0.363	0.370	0.228	0.277

Table 4: Results of the ensemble models (top-1 accuracy) on various datasets along with the heuristic model. The accuracies reported are based on weighted voting of the experts. K indicates number of experts. OL indicates overlap.

Model	Top-2 Words Accuracy					
	Test	*OL*	*No OL*	*Cook*	*Thurner*	*Believa*
Bidirectional Binary (K=60)	0.635	0.677	**0.570**	0.613	0.504	0.575
Hybrid Bidirectional Binary (K=30)	**0.677**	0.784	0.527	**0.674**	0.612	**0.597**
Heuristic Conditional	0.434	**0.909**	0.068	0.401	**0.641**	0.481

Table 5: Results of the ensemble models (top-2 accuracy) on various datasets along with the heuristic model.

Training The ensemble model of experts is trained end-to-end on mini-batches using Adam Optimization (Kinga and Adam, 2015) over the Wiktionary dataset. The mini batch size is 100 and maximum length of the concatenated source word pair is 29. The learning rate is initialized to 0.006. The fixed sized vector embedding representation of the inputs to our model is set at 128 as well. We allow a dropout rate of 0.25 to prevent over-fitting. The parameters of the model are randomly initialized from a normal disribution with zero mean and standard deviation of 0.1. Training is run for 16 epochs. We tuned hyper-parameters based on our models performance over a range on the validation sets.

3.3 Baseline Heuristic, Neural and Hybrid Models

We compare the performance of our proposed model with other models. The baselines are the conditional probability distribution based model and the heuristic model described previously.

[6] https://en.wiktionary.org/wiki/Category:English_blends Note: Last accessed $14^{th} March, 2017$

[7] We are grateful to the authors for sharing with us their dataset.

We also compared our model with several types of Seq2Seq models. These include the vanilla char-to-char where the encoder takes source words concatenated and the decoder outputs target blend as characters, Attention char-to-char with added attention mechanism (Bahdanau et al., 2014) to the vanilla char-to-char and Split encoder where two encoders each takes one source word and the decoder takes concatenated encoder states as input. Index based target prediction models that we compared against are the Pointer feedforward model that has an encoder like the vanilla Seq2Seq but the decoder is replaced with feed forward network that outputs a distribution over two indices or pointers that indicate splice points. Pointer decoder is another variant where the decoder is unrolled for two time steps always, each providing a distribution over indices of the concatenated source words. Binary target prediction models including our proposed ensemble model outputs a probability of including the character from the source concatenation at each time step. Attention mechanisms were attempted for the other models but showed no meaningful improvement in performance; the attention mask results were blurred in most cases. We suspect this is due to the mismatch in the encoder and decoder representations (characters versus binary or indices).

3.4 Results and Analysis

The demo of our model is available online[8]. Dataset and source code[9] is also provided. In all the results reported, character level accuracy indicates the generalized accuracy over number of characters that were predicted correctly in the blend word. Word level accuracy indicates the generalized accuracy over the number of correct blend word predictions over all the source word pair instances.

Table 3 shows the results of all our single instance models as discussed in the previous section. The heuristic conditional model achieves high accuracy in correctly predicting overlapping blends. Its outperforms all the other models in the overlapping blends subset dataset. Overlapping blends account to about fifty percent of the total blends in general. The heuristic conditional model however does poorly in the non overlapping blends as it defaults to a greedy search in order to find the single character overlaps, such that its performance is worse than the plain conditional model in this subset.

In the char-to-char models, the attention based model performs best as it is able to observe the input words in their entirety at every time step. Figure 2 shows the attention mask for this model while predicting the blend `infotopia`. We can clearly see where it jumps to the second word. The two other char-to-char models perform poorly as they have to additionally track the current character to output. The index based models perform slightly better as the problem is now about predicting the correct splice points. Their performance is relatively unaffected by the blend types - overlapping or non-overlapping. The char-to-binary models outperforms the rest as the solution space is restricted to predicting the binary vector indicating the characters that are present in the blend. The bidirectional model is further able to improve its performance due to its ability to scan the character sequence in both directions for exploiting neighbourhood structure. It can make a better decision on the splinters with this extra information. The vanilla binary model is able to perform better on the overlapping blends data than the bidirectional model but loses out on the non-overlapping case. The bidirectional model essentially is able to generalize to both overlapping and non-overlapping subsets. Finally the hybrid model between heuristic and neural performs the best by including extra information on the overlaps when available. Although it is unable to beat the heuristic model in the overlapping blends subset, it does better overall.

In Tables 4 and 5, we report the performance of our proposed bidirectional ensemble model. The ensemble models are able to bring in significant gains as compared to the single instance models. The hybrid model is able to get the best of both world: heuristic and neural models to get the best overall accuracy. However, its performance is skewed towards the overlapping blends resulting in lower performance on the no overlap data subset in comparison to the plain bidirectional model. The heuristic model performs well on the Thurner (1993) dataset due to the majority of the words in the Thurner data comprising overlapping blends. The Cook (2010) and Believa datasets (2014) are more balanced in which the hybrid model outperforms the rest. The Mix-

[8] https://neuroblender.herokuapp.com

[9] https://github.com/FreeEnergy/neuramanteau

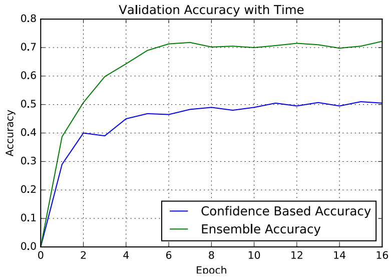

Figure 3: Performance Plot of the Bidirectional Encoder Decoder SeqToSeq model.

$\textbf{Word}_1$	$\textbf{Word}_2$	**Hybrid Prediction**	**Heuristic Prediction**	**Ground Truth**
work	alcoholic	workoholic, workholic	woholic, wolic	workoholic
snow	apocalypse	snowocalpyse, snowpocalypse	snocalypse	snowpocalypse
book	bootlegger	booklegger	boootlegger, bookbootlegger	booklegger
family	honeymoon	famimoon, familmoon	famoon, familymoon	familymoon
edge	pixel	edgixel, edxel	edgel	edgel

Table 6: Comparison of predictions from heuristic and ensemble hybrid models on sample inputs.

ture of Experts (MoE) ensemble causes experts to specialize individually to the examples early on, most often converging to a subset of experts. This led to reduced accuracies for MoE throughout. It was difficult to train the model as it preferred to converge (specialize) to a few experts even when enforcing a variance loss to encourage diversity. In contrast, the bidirectional ensemble model did not specialize as each expert is trained separately. That independence helps the ensemble to generalize in capturing the wider variations in the blends. In Table 6, we compare the predictions from the hybrid ensemble model and the heuristic single instance models. On observing the failure modes, due to the qualitative and subjective nature of blends, we find that some of the predicted blends are still quite plausible. We would hazard to say that some of these predictions actually look more natural than the dataset values.

Model	**Top-1 Word Accuracy**					
	Test	*OL*	*No OL*	*Cook*	*Thurner*	*Believa*
Bidirectional Binary with Dominance(K=15)	0.491	0.505	0.491	0.517	0.378	0.415
Hybrid Bidirectional Binary with Dominance (K=15)	0.572	0.721	0.453	0.643	0.525	0.487

Table 7: Results of dominance based models. K indicates number of experts. OL indicates overlaps.

$\textbf{Word}_1$	$\textbf{Word}_2$	**Blend Predictions**		
		$\textbf{Word}_1$ Dominant	$\textbf{Word}_2$ Dominant	Equal Dominannce
breakfast	lunch	breaklunch	brunch	breaklunch
phone	tablet	phonelet	phablet	phonlet
aviation	electronics	aviationics	avionics	aviatonics
bombay	hollywood	bombaywood	bollywood	bombwood
republican	democrat	republicrat	repumocrat	repubocrat

Table 8: Sample predictions from hybrid ensemble with variation in dominance.

Figure 3 shows the performance plot of the validation accuracy of any expert predicting correctly in comparison to the weighted voting prediction. The weaker metric of evaluating the accuracy based on any expert, outperforms the weighted voting prediction over the training duration. Tables 8 and 7 shows some sample predictions and accuracy by the dominance enhanced hybrid model on sample inputs. With the famous `brunch` blend, our model predicts that the blend would have been `breaklunch` if `breakfast` had been given more importance by the coiner. Similarly our model predicts `phone` and `tablet` gives `phonlet` when dominance of the source words are same. But when the word `tablet` is set to be dominant the model predicts `phablet`, the form of the blend which is in popular use now. We can hypothesize that the creator of this blend perhaps wanted more emphasis on tablets when marketing `phablet` form factor mobile devices.

Comparison with Deri et al. (2015) The main working example in their published paper: stay + vacation predicted `stacation` on their demo website, instead of `staycation` as claimed, whereas our model demo predicted `staycation`. Further, on their dataset (with common words with the Wiktionary dataset that we used removed), our baseline heuristic beat their model with accuracy of 47.7% in comparison to their models 45.3%, while our primary model achieved 48.3%. Their dataset consists of 400 examples in contrast to our dataset of 2854 examples. Their system is unable to generalize to non-overlapping blends like `staycation`, `workholic` correctly which our system can.

4 Future Work and Conclusions

In this work, we show that neural networks are well suited to modelling uncertainties in the blending process. The ensemble RNN neural and ensemble RNN neural-hybrid encoder-decoder systems that we propose generalized very well to overlapping and non-overlapping blended English words from two source words. They outperform statistical, heuristic, neural single instance and mixture of experts ensemble models over multiple datasets. However, these ensemble models are unable to capture the stringent rules and restrictions that disallow certain character combinations like `bxy`, `ii`, `gls`. An attempt to tag inputs with phonetic or articulatory information failed to correct these mistakes. One possibility is to use reinforcement learning (Sutton and Barto, 1998) to apply specific rules of word formation. Other types of errors are recognizability errors which causes loss of recognition of one or both source words and over-representation errors which adds extra (and unnecessary) characters from the source words. Some of these examples are provided in the Appendix. We believe these errors occur due to the sparsity in the training data.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Laurie Bauer. 2012. Blends: Core and periphery. *Cross-disciplinary perspectives on lexical blending*, pages 11–22.

Natalia Beliaeva. 2014. A study of english blends: From structure to meaning and back again. *Word Structure*, 7(1):29–54.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Paul Cook and Suzanne Stevenson. 2010. Automatically identifying the source words of lexical blends in english. *Computational Linguistics*, 36(1):129–149.

Aliya Deri and Kevin Knight. 2015. How to make a frenemy: Multitape fsts for portmanteau generation. In *HLT-NAACL*, pages 206–210.

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

Stefan Th Gries. 2006. Cognitive determinants of subtractive word formation: A corpus-based perspective.

Stefan Th Gries. 2012. Quantitative corpus data on blend formation: psycho-and cognitive-linguistic perspectives. *Cross-disciplinary perspectives on lexical blending*, pages 145–167.

Michael H Kelly. 1998. To brunch or to brench: Some aspects of blend structure.

D Kinga and J Ba Adam. 2015. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.

Adrienne Lehrer. 2007. Blendalicious. *Lexical creativity, texts and contexts*, pages 115–133.

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*, pages 703–711. Association for Computational Linguistics.

Maciej Pilichowski and Włodzisław Duch. 2013. Braingene: computational creativity algorithm that invents novel interesting names. In *Computational Intelligence for Human-like Intelligence (CIHLI), 2013 IEEE Symposium on*, pages 92–99. IEEE.

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.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Dick Thurner. 1993. *Portmanteau dictionary: blend words in the English language, including trademarks and brand es*. McFarland & Company.

Andrew M White, Katherine Shaw, Fabian Monrose, and Elliott Moreton. 2014. Isn't that fantabulous: Security, linguistic and usability challenges of pronounceable tokens. In *Proceedings of the 2014 workshop on New Security Paradigms Workshop*, pages 25–38. ACM.