Phonotactic Modeling of Extremely Low Resource Languages

Andrei Shcherbakov[1] and Ekaterina Vylomova[2] and Nick Thieberger[1]
[1]School of Languages and Linguistics
[2]Department of Computing and Information Systems
The University of Melbourne
`ultrasparc@yandex.ru evylomova@gmail.com thien@unimelb.edu.au`

Abstract

This paper presents a novel approach to low resource language modeling. Here we propose a model for word prediction which is based on multi-variant ngram abstraction with weighted confidence level. We demonstrate a significant improvement in word recall over "traditional" Kneser-Ney back-off model for most of the examined low resource languages.

1 Introduction

For a dictionary in the course of being created, a problem for the linguist/lexicographer is how to find words that are not already recorded in the dictionary. Linguists working on small languages may extract as many lexemes as they can from texts they record, and then add to that with elicitation of items unlikely to appear in texts, especially paradigmatic information (like verb inflections, pronouns, kin terms and so on), place names, and biological names. They are now also using experimental stimuli to get at more nuanced meanings. However, once they have exhausted these sources, they need some way to discover other possible forms in the language. In this paper we discuss a method for creating possible word forms that can be confirmed by speakers (or not) as being words in the language. A general approach that has been used before is to apply the phonotactics to generate other possible forms (Prince and Tesar, 2004; Dell et al., 2000; Goldrick, 2004; Heinz, 2007). This was done with flip charts of possible sequences of phonemes, and then computationally, generating forms on the basis of known permissible combinations.[1]

This task could also be viewed from the perspective of machine learning, representing a particular case of language modeling. We can state

---

[1]See e.g. `http://billposer.org/Software/WordGenerator.html`

the task as follows. The model gets an initially collected vocabulary of a given language as an input and has to predict likely undiscovered words as accurately as possible. There are many approaches to this task. The most powerful ones known today are based on neural networks, including deep learning techniques (Sundermeyer et al., 2012; Kim et al., 2015; Sundermeyer et al., 2015; Hwang and Sung, 2016; Oparin et al., 2012). Unfortunately, the ability of such algorithms to capture knowledge depends heavily on the amount of training data, and generally they aren't usable for a real language unless one provides at least several thousand training instances to them. Although, theoretically, such algorithms are able to learn many sophisticated rules not obvious even for a human analyst, the rules are hard to validate.

Typically, for low resource languages we are not able to obtain a training vocabulary of sufficient size. In the current research, we consider a practical task where an initial vocabulary of 300+ basic words has to be extended with more or less proper candidate words. For an optimal likelihood of hitting new words, we need to produce more diverse word forms while keeping compliance to phonotactical rules of word formation. Many researchers (Onishi et al., 2002; Blevins, 2003; Warker and Dell, 2006; Edwards et al., 2004; Chambers et al., 2003; Luce and Large, 2001; Vitevitch and Luce, 2005) have identified the importance of phonotactics in the process of word prediction.

Fortunately, pattern-based language modeling is a well explored topic in NLP. A pattern, e.g. an ngram, here may mean a sequence of some language units (phrases, words, characters, phonemes); these approaches mainly rely on probabilistic evaluation of various sequence likelihood based on training set statistics. A text corpus or vocabulary may be used for pattern distribution evaluation. However, as we will see in our experiments, such algorithms are too simplistic for really

small vocabularies.

A recurrent back-off to an $(n-1)$-gram suffix of a given n-gram is the most common strategy. In the context of smoothing for language models, back-off n-gram models have been well studied (Chen and Goodman, 1999). Among the myriad of proposed approaches, the Kneser-Ney approach is widely considered to be the best approach. Therefore, we consider Kneser-Ney smoothing as a baseline for evaluating the proposed algorithm.

We demonstrate that suffix gram-based back-off approaches may lack predictive power for small vocabularies, and go on to propose two major new methods: (1) multi-variant abstraction in n-grams, and (2) a distributional confidence metric for abstracted n-gram likelihood estimation. The algorithm based on these methods has been successfully realized in the Word Generator application[2] that is being currently used in field linguistics studies of endangered languages.

The paper is structured as follows. In Section 2 we provide a description of the system's architecture. Then in Section 3 we describe the languages we used and the corresponding vocabularies. In Section 4 we provide our experiments settings. And, finally, we discuss the results and observations in Section 5.

2 Architecture

2.1 Multi-Variant abstraction

In the context of our work, a word is considered as a sequence of phonemes that is expected to obey latent phonotactical rules. The rules are estimated based on the training data. The choice of phonemes (vs. characters) is primarily based on the fact that many low resource languages simply didn't have any script system during active phases of their evolution, and modern scripts merely follow their phonetic representation. We also note that our experiments with English have shown that phonetic representation enables better prediction results than orthographic one.

We now describe the process of generating abstracted forms in more detail. First, we add beginning (ˆ) and finishing ($) quasi-phonemes to each word. Then the system parses the source vocabulary and records frequency of each observed ngram. The length N of a gram is limited by a parameter $MaxNG$. In our experiments $MaxNG$ is chosen to be 5 since it yields the best results for the majority of tested languages.

We then augment the list of *concrete* ngrams with k-*abstracted* ones, where k is set of abstracted positions.[3] Each position $p \in k$ should satisfy the following two conditions. First, the pth phoneme in the original *concrete* ngram should be ***abstractable***. *Abstractable* here means that it's known either as a vowel or a consonant phoneme; quasi-phonemes and specials, such as pauses or phoneme modifiers, are not abstractable. Second, the tail phoneme in a ngram is exempt of abstraction always being treated as concrete; thus, the following inequality should be true: $1 \leq p \leq N - 1$. We will further refer to the final list of *concrete* and *abstracted* ngrams as V^+. As an example, suppose we observed a 5-gram `ˆbats` in the training vocabulary. The system generates the following abstracted 5-grams: `ˆⒸats`, `ˆbⓋts`, `ˆbaⒸs`, `ˆⒸⓋts`, `ˆbⓋⒸs`, `ˆⒸaⒸs`, and `ˆⒸⓋⒸs` where Ⓥ and Ⓒ are abstract vowel and consonant, respectively. Again, note the first pseudo-phoneme is not abstracted here as it's neither vowel nor a consonant.

At the next step ngrams that differ by the last phoneme only, are grouped into a structure referred to as ***selector*** over some prefix $(n-1)$-gram.

Then the algorithm builds a candidate word starting with a single `ˆ` and adds phonemes one by one. A word is complete once a `$` is appended. This assumption is based on the observation that word boundaries are tied to some phonotactical patterns as well (McQueen, 1998; Brent and Cartwright, 1996; Friederici and Wessels, 1993). We now take a closer look at the process of a word formation. Let w be a word prefix that has been generated up to this point, and L be its length. The algorithm takes into consideration a $(n-1)$-gram consisting of $N-1$ trailing phonemes of w, where $N = \max(L, MaxNG)$. The next phoneme is actually produced in two steps. Firstly, we randomly choose a single *selector* from V^+ which prefix $(n-1)$-gram matches w. The probability of each candidate is proportional to its abstraction confidence level (see below) as well as a direct function of its vocabu-

---
[3]Adriaans and Kager (2010) also noted the utility of adding abstraction of phonemes
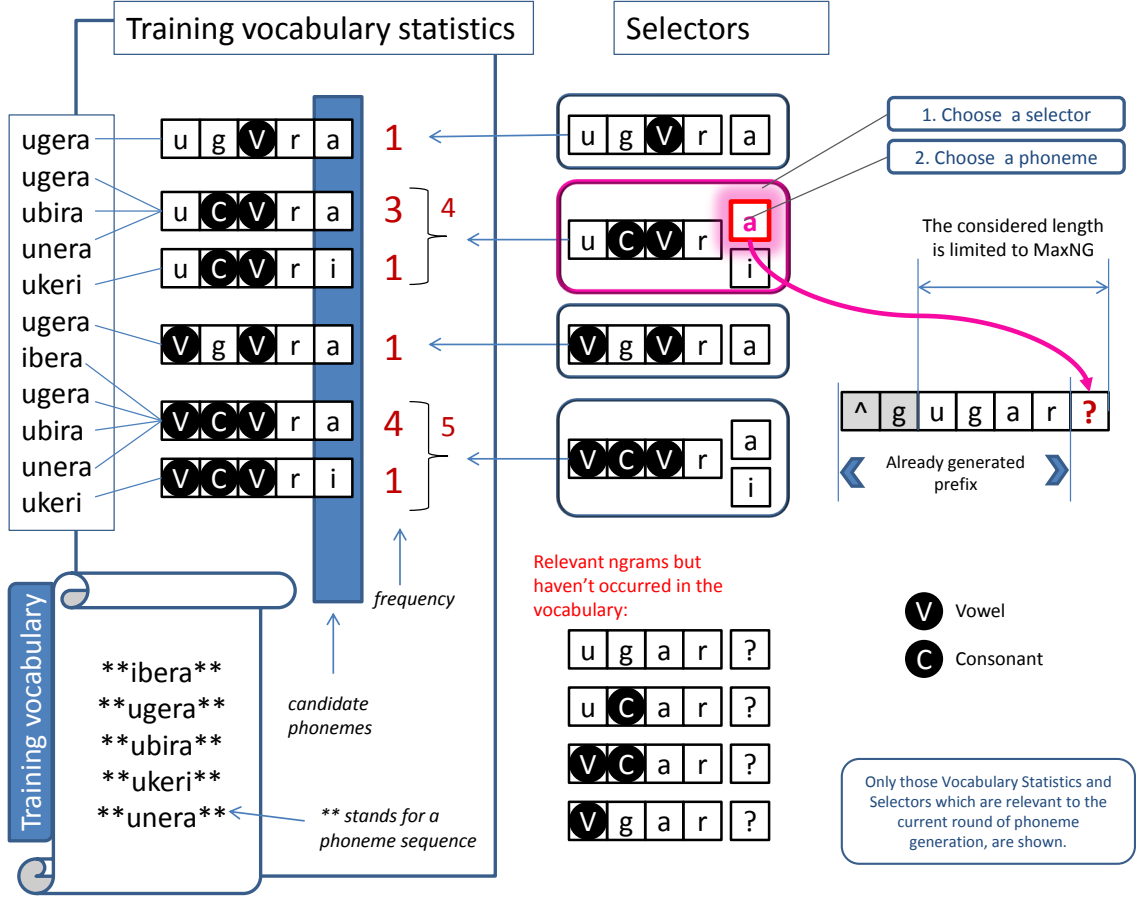
Figure 1: An example of phoneme generation. A prefix '^gugar' is assumed to be already generated at this point. The next phoneme choice is based on a fraction of our toy vocabulary relevant to the 'ugar' trailing $(n-1)$-gram, $n = 5$

lary frequency. Secondly, we choose an n-gram from the given *selector* content with the probability proportional to its frequency. The trailing phoneme of chosen n-gram is nominated to be the next phoneme of the generated word. Indeed, this procedure assigns smoothed conditional probability of y phoneme addition to a given $(n-1)$-gram $\tilde{w}$ as follows:

$$p(y|\tilde{w}) = \sum_{g \in \tilde{w}^+} \left(\frac{c(g,y)}{c(g)} \cdot \frac{F(g)f_s(c(g))}{S} \right) \quad (1)$$

where $c(x)$ is a count of x gram occurrences in the training vocabulary; S is a normalizing constant; f_s is a frequency squashing function; in our experiments, $f_s(n) = \log_2(1+n)$; F(g) is the confidence level of g gram which is discussed in more detail in 2.2; x^+ is abstract closure of x gram, i.e. set of concrete or/and abstract grams known in training vocabulary statistics (V^+) that satisfy

x gram:

$$x^+ = \{x' \in V^+ : x' \models x\} \quad (2)$$

Equation 1 generally looks like a "traditional" back-off probability but with more potentially contributing terms and with confidence levels introduced. Figure 1 illustrates candidate phoneme selection process in a toy vocabulary example.

2.2 Confidence level

As shown above, we apply various abstracted ngrams in order to predict the next phoneme. Although such an approach has an advantage of multiple pattern usage, it's pretty clear that soundness of each abstraction may differ. For example, consider 'xxoxx', 'xxaxx', 'xxexx', 'xxixx' grams that occur uniformly in the vocabulary. Here it's likely that any vowel is suitable to appear in the middle position of these grams. However, if we observe just a single pattern 'xxaxx' then we consider 'a'

as an immutable phoneme at its position. Taking into account such considerations, we attempt to evaluate the 'appropriateness' of each abstracted ngram and avoid over-generalization. We used an entropy-like metric representing the uniformity of more concrete sequences distribution over an embracing abstracted ngram. For an n-gram g containing $a(g) > 0$ abstracted phonemes, the following formula denotes a confidence level:

$$F(g) = \alpha^{a(g)-1} \cdot z(F'(g)) \qquad (3)$$

Here α is a constant found to have the optimum value $\alpha \approx 0.9$; z is a simple piece-wise function which setting is considered in 4.2 ; $F'(g)$ is a confidence metric itself calculated as follows.

$$F'(g) = \left\langle F'(h_i) \log_2 \left(1 + \frac{c(h_i)F'(h_i)}{\langle c(h_j)F'(h_j)\rangle} \right) \right\rangle \qquad (4)$$

Averages $\langle \ldots \rangle$ here are taken over all satisfying ngrams h_k having one less abstracted phonemes, i.e over

$$\forall h_k : g \models h_k \wedge a(h_k) = a(g) - 1$$

For concrete ngrams, we assume that confidence levels equal to one:

$$a(g) = 0 \implies F(g) = F'(g) = 1 \qquad (5)$$

3   Data set

In our study we consider two language families: Oceanic (Austronesian) and Pama-Nyungan. For the first group we take three languages spoken in Vanuatu: South Efate (Central Vanuatu), Vurës (Northern Vanuatu), and Tamambo (Northern Vanuatu). Phonologically they are quite different to those spoken in Australia. Their sound system consists of around 15 consonants and 5 vowels ([a], [i], [u], [o], [e]).

Languages of the second group are spoken in Australia. Pama-Nyungan presents the most widely spread language group in Australia and covers 7/8 of its territory. Most Australian languages present quite similar phonology comprising 15-17 consonants and 3 vowels with variative length ([i], [i:], [u], [u:], [a], [a:]) (Baker, 2014; Hamilton, 1996; Busby, 1980). Gamilaraay (northern New South Wales), Kukatja (Western Desert), and Wik-Mungkan (Cape York Peninsula) belongs to Pama-Nyungan family.

The number of speakers varies from as low as 35 (Gamilaraay, 2006) up to 6,000 (South Efate, 2005).

Table 1 summarizes the basic sizes of language vocabularies used in our experiments.

Language	#Words	#Vowels	#Cons.
Gamilaraay	2423	5	15
Kukatja	8632	5	12
South Efate	2575	5	15
Tamambo	2067	5	15
Vurës	2166	9	16
Wambaya	1195	3	17
Wik-Mungkan	3884	10	17

Table 1: Explored low resource languages vocabulary summary

4   Experiments

4.1   Method and Measure

In our experiments we randomly split the data into the training and the test parts. We run cross-validation several times, each time shuffling the data. In each run the model generates a fixed size set of samples (words). To evaluate the model's quality, we measure the recall over the test vocabulary, i.e. the ratio of generated word hits out of the test vocabulary. We suppose that such an evaluation procedure would model adequately an exploration of unknown words as well, projecting a train set and a test set into a today's known part of a language vocabulary and its yet undiscovered part, respectively. In order to emphasize the dictionary incompleteness and the extrapolation intent, we'll use the *Recall rate* term in place of mere *Recall*. Note that the maximum theoretically reachable recall value is below one in most of the reported experiments, due to a relatively low number of generated words; nevertheless, we do use such a convenient metric unless that limitation really affects the result.

4.2   A study on confidence level

We use a simplex optimization method to find the best approximation of z function in Equation (3) that maps confidence into the likelihood. We approximate it as a piece-wise linear function. The objective is to achieve the best mean recall rate for low resources languages we examined; we repeated measurements with various sizes of training vocabularies taken in a uniform proportion.
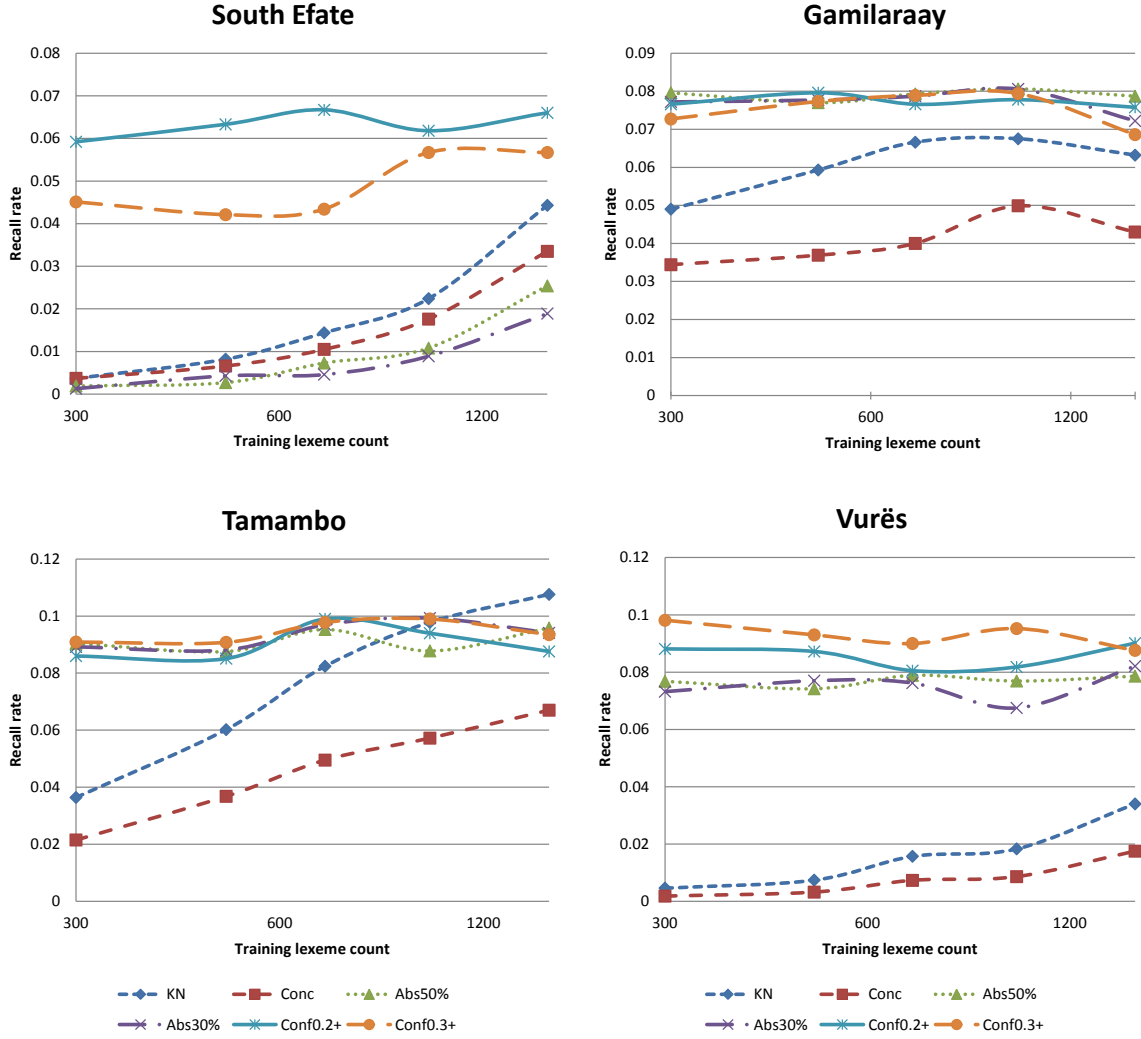
Figure 2: Recall rate vs. training vocabulary size ($300 \ldots 1500$ words) for low resource languages

Surprisingly, the optimal function was (almost) linear at the middle range of its domain; after validating a series of solving results and some non-affecting rounding of digits we have chosen the following two options for further experiments. The first one, referred to as $\mathbf{CONF}_{0.3+}$, yields the best mean result as well as best results for most of languages explored:

$$z(x) = \left\{ \begin{array}{ll} 0, & \text{iff } x < 0.3 \\ 0.6, & \text{iff } 0.3 \leq x < 0.4 \\ 0.8, & \text{iff } 0.4 \leq x < 0.5 \\ 1, & \text{iff } x \geq 0.5 \end{array} \right\} \quad (6)$$

The second one, $\mathbf{CONF}_{0.2+}$, was preferable in some cases:

$$z(x) = \left\{ \begin{array}{ll} 0, & \text{iff } x < 0.2 \\ \min(\frac{100x-2}{45}, 1), & \text{iff } x \geq 0.2 \end{array} \right\} \quad (7)$$

4.3 Options

In our contrastive experiments we use the following algorithms and options (the identifiers correspond to those found in graph legends).

$\mathbf{CONC}$ - concrete ngrams only used for word generation, i.e. no vowel or consonant abstraction is allowed at all.

$\mathbf{KN}$ - Kneser-Ney back-off, $\delta = 1$

$\mathbf{ABS50\%}$, $\mathbf{ABS30\%}$ denote multi-variant abstractions with fixed uniform confidence ($F' \equiv 0.5$ and $F' \equiv 0.3$ in Equation 3, respectively.)

$\mathbf{CONF0.3+}$, $\mathbf{CONF0.2+}$ represent multi-variant abstraction with variable confidence computed according to Equation 6 or Equation 7, respectively.
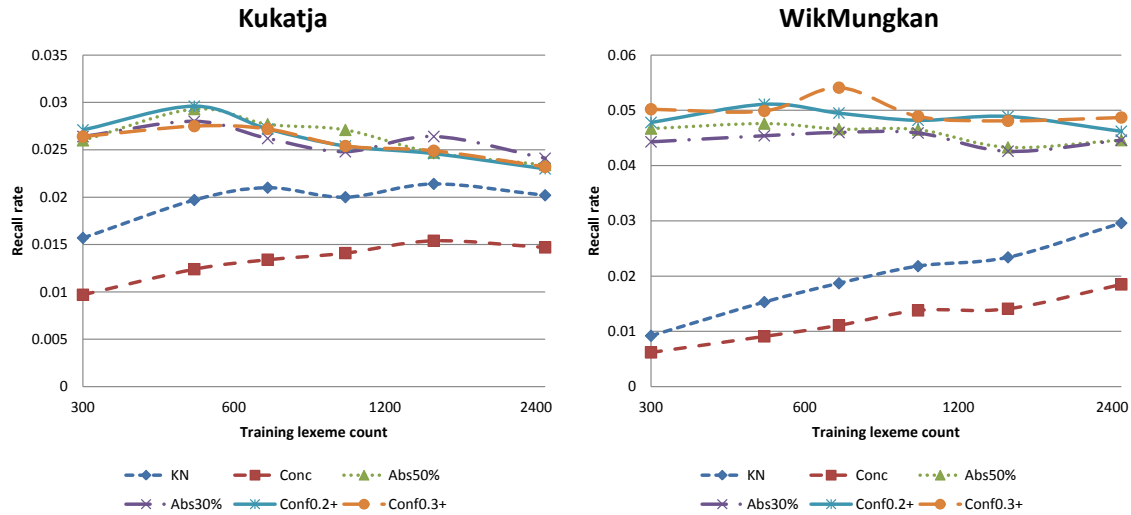
88

Figure 3: Recall rate vs. training vocabulary size (300 . . . 2500 words) for low resource languages
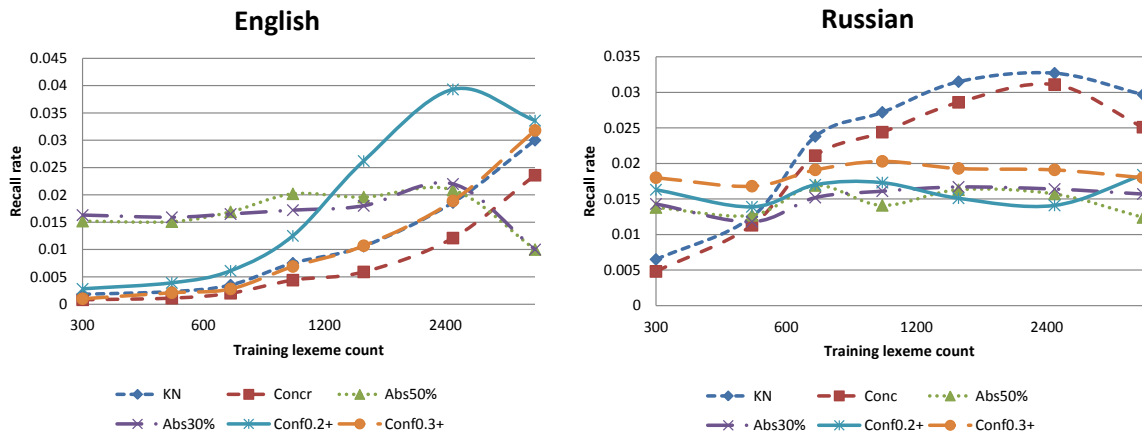

Figure 4: Recall rate vs. training vocabulary size (300 . . . 4000 words) for high resource languages

4.4 Recall & Precision vs. Training word count

In each trial we generate 1000 words for each language and measure the recall rate as described in 4.1. Figure 2 shows how the recall rate depends on the training vocabulary size at various options for South Efate, Gamilaraay, Tamambo and Vurës languages. We tried training vocabularies of 300, 500, 700 and 1500 words. For Kukatja and Wik-Mungan, we also examined a larger training word list, of 2500 words (see Figure 3)

For the comparison, we built similar graphs for two very high resource languages, Russian and English[4] (Tabain et al., 2004; Kipyatkova and Kar-

pov, 2015), restricting the total vocabulary used in experiments to 5000 most frequently used words of each languages. In such a way we were simulating a low resource environment. The results are presented at Figure 4. Also, in order to catch a picture of models' predictive power in a large resource vocabulary context, we attempted predicting words of a large vocabulary having the models trained on a given number of most frequently used words. We generated 1000 non-learned words each time and measured precision[5] of vocable pre-

---

[4]English words were broken into phoneme-representing character sequences before the training.

[5]At fixed counts of generated and test words, the precision is proportional to the recall. Thus, these two metrics are of similar sense for comparing the algorithms quality. We prefer the precision here just because the test vocabulary is vastly larger in volume than a generated word set, which fact lowers the maximum recall value terribly.
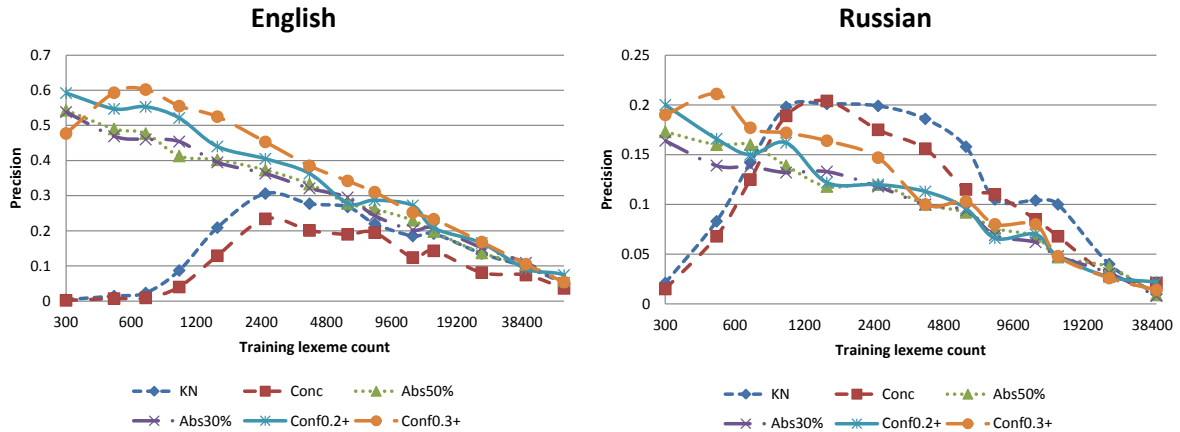
Figure 5: Precision against a large vocabulary vs. count of (most frequent) training words for high resource languages

diction (see Figure 5). We used Wiktionary top 100,000 most frequently-used English words and the frequency dictionary of modern Russian language (Lyashevskaya and Sharov, 2009) as large dictionaries.

4.5 Recall vs. Generated word count

In this experiment we evaluate the recall value achieved upon k words has been generated, $k \in [1 \dots 10,000]$. We use training vocabularies of 1500 words each. The idea here is to check the ability of each method to hit less easily derivable 'fractions' of vocabulary content.

Some of the graphs are shown on Figure 7. As we see, the proposed multi-variant abstraction methods keep hitting new words pretty well even when thousands of test dictionary words are nailed and excluded of further targeting; in contrast, concrete ngram approach and even Kneser-Ney backoff tend to decline yielding an equally high hit rate at some point (and sometimes nearly stop hitting at all), despite any possible advantage they may possess at the start.

4.6 Inflected words Recall

We roughly estimated the recall of inflected word forms for South Efate language. To get a collection of inflected words, we extracted all words of stories collected in (Thieberger, 2011). Then we filtered out known lemmas as well as proper nouns. The rest has been used a test set for the inflected word hit detection. We used random 1500 vocable samples to train the generator and varied the number of generated words, exactly as we did
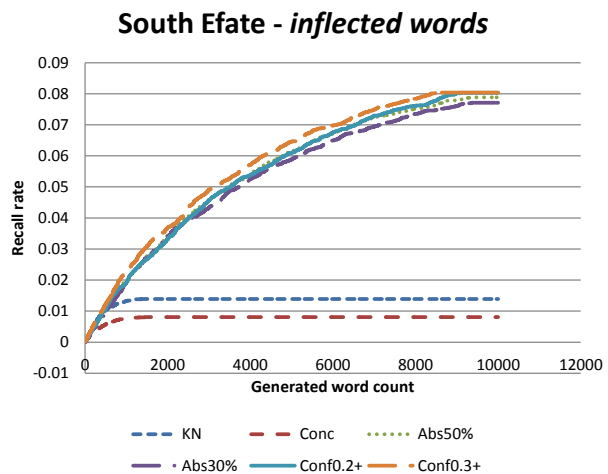

Figure 6: Recall rate of inflected words found in "Stories from South Efate" vs. generated word count

in 4.5. The recall curves displayed at Figure 6 appear to be similar to ones at Figure 7 (for South Efate), demonstrating an approximately constant ratio of about 3.6 between lemmas and inflected forms over generated words.[6]

5 Discussion

Unsurprisingly, the increase of the training vocabulary yields more chances to produce meaningful words merely attempting to reuse concrete sequences, and power of abstraction gradually decreases with training vocabulary size increase.

---

[6]This should be respected as an overestimation since not all possible inflected forms are present in the text corpus
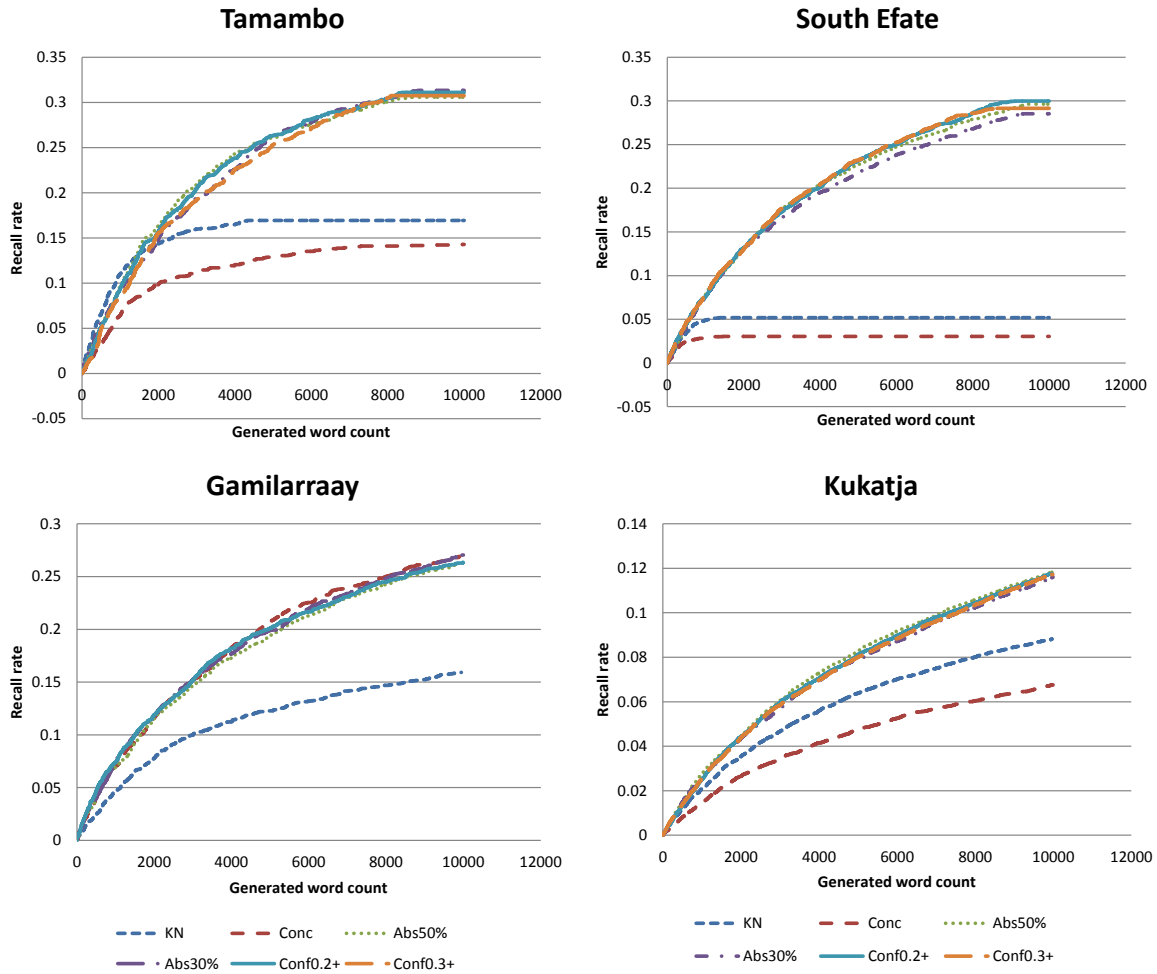
Figure 7: Recall rate as a function of generated word count for training vocabularies of 1500 words

The level of necessary concreteness may essentially vary over languages. Most languages demonstrate strong preference towards the abstracted approach. This feature doesn't quite correlate to a language family, it's more related to each language's own phonotactics instead. For example, two of three examined Austronesian languages follow that rule, but the third one doesn't. For such languages, the proposed multi-variant abstraction algorithm works fine, and with `Conf0.3+` option it outperforms the Kneser-Ney approach significantly, especially for very small vocabularies: its recall is higher in about 1.3 to 8 times.

However, some language vocabularies still tend to be much more predictable by following concrete ngrams, except for cases of really tiny training sets. At those "phonotactically concrete" languages the proposed technique may play either around the the same or somewhat less efficient than Kneser-Ney smoothing (which one indeed reproduces continues concrete grams); still, in such cases increasing the weight of concrete ngrams may more directly address the issue than merely adjusting the smoothing and abstraction mode.

In hunting more unusual words (at the cost of precision), when one generates large number of candidate words, the proposed algorithm demonstrates its advantages almost regardless of language and training set size.

During the experiments with word generator it was reported that the generator rather frequently produces inflected forms of known words. In this view, usage of existing text corpora for filtering out such forms, as well as learning of inflection rule patterns may effectively increase the precision of word prediction algorithm w.r.t. vocabularies.

6 Conclusions

We proposed a novel abstraction technique and confidence metric in probabilistic language modeling and demonstrated its advantage for different families of low resource languages. We also outlined two primary areas for further research. They include finding a self-adjustable balance between concrete and abstracted consideration, and improving vocable prediction by filtering out inflection patterns (Tesar and Prince, 2003).

References

Frans Adriaans and René Kager. 2010. Adding generalization to statistical learning: The induction of phonotactics from continuous speech. *Journal of Memory and Language*, 62(3):311–331.

Brett Baker. 2014. 4 word structure in australian languages. *The Languages and Linguistics of Australia: A comprehensive guide*, 3:139.

Juliette Blevins. 2003. The independent nature of phonotactic constraints: an alternative to syllable-based approaches. *The syllable in optimality theory*, pages 375–403.

Michael R Brent and Timothy A Cartwright. 1996. Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, 61(1):93–125.

Peter A Busby. 1980. The distribution of phonemes in australian aboriginal languages. *Pacific Linguistics. Series A. Occasional Papers*, (60):73.

Kyle E Chambers, Kristine H Onishi, and Cynthia Fisher. 2003. Infants learn phonotactic regularities from brief auditory experience. *Cognition*, 87(2):B69–B77.

Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394.

Gary S Dell, Kristopher D Reed, David R Adams, and Antje S Meyer. 2000. Speech errors, phonotactic constraints, and implicit learning: a study of the role of experience in language production. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 26(6):1355.

Jan Edwards, Mary E Beckman, and Benjamin Munson. 2004. The interaction between vocabulary size and phonotactic probability effects on children's production accuracy and fluency in nonword repetition. *Journal of speech, language, and hearing research*, 47(2):421–436.

Angela D Friederici and Jeanine MI Wessels. 1993. Phonotactic knowledge of word boundaries and its use in infant speech perception. *Perception & psychophysics*, 54(3):287–295.

Matthew Goldrick. 2004. Phonological features and phonotactic constraints in speech production. *Journal of Memory and Language*, 51(4):586–603.

Philip James Hamilton. 1996. Phonetic constraints and markedness in the phonotactics of australian aboriginal languages (1996). *Toronto Working Papers in Linguistics*.

Jeffrey Nicholas Heinz. 2007. *Inductive learning of phonotactic patterns*. ProQuest.

Kyuyeon Hwang and Wonyong Sung. 2016. Character-level language modeling with hierarchical recurrent neural networks. *arXiv preprint arXiv:1609.03777*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2015. Character-aware neural language models. *arXiv preprint arXiv:1508.06615*.

Irina Kipyatkova and Alexey Karpov. 2015. A comparison of rnn lm and flm for russian speech recognition. In *International Conference on Speech and Computer*, pages 42–50. Springer.

Paul A Luce and Nathan R Large. 2001. Phonotactics, density, and entropy in spoken word recognition. *Language and Cognitive Processes*, 16(5-6):565–581.

O Lyashevskaya and S Sharov. 2009. The frequency dictionary of modern russian language. *Azbukovnik, Moscow*.

James M McQueen. 1998. Segmentation of continuous speech using phonotactics. *Journal of memory and language*, 39(1):21–46.

Kristine H Onishi, Kyle E Chambers, and Cynthia Fisher. 2002. Learning phonotactic constraints from brief auditory experience. *Cognition*, 83(1):B13–B23.

Ilya Oparin, Martin Sundermeyer, Hermann Ney, and Jean-Luc Gauvain. 2012. Performance analysis of neural networks in combination with n-gram language models. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5005–5008. IEEE.

Alan Prince and Bruce Tesar. 2004. Learning phonotactic distributions. *Constraints in phonological acquisition*, pages 245–291.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Interspeech*, pages 194–197.

Martin Sundermeyer, Hermann Ney, and Ralf Schlüter. 2015. From feedforward to recurrent lstm neural networks for language modeling. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(3):517–529.

Marija Tabain, Gavan Breen, and Andrew Butcher. 2004. Vc vs. cv syllables: a comparison of aboriginal languages with english. *Journal of the International Phonetic Association*, 34(02):175–200.

Bruce Tesar and Alan Prince. 2003. Using phonotactics to learn phonological alternations. In *Proceedings from the annual meeting of the Chicago Linguistic Society*, volume 39, pages 241–269. Chicago Linguistic Society.

Nick Thieberger. 2011. Natrauswen nig efat: stories from south efate.

Michael S Vitevitch and Paul A Luce. 2005. Increases in phonotactic probability facilitate spoken nonword repetition. *Journal of memory and language*, 52(2):193–204.

Jill A Warker and Gary S Dell. 2006. Speech errors reflect newly learned phonotactic constraints. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32(2):387.