Open-lexicon Language Modeling Combining Word and Character Levels

Michał Kozielski, Martin Matysiak, Patrick Doetsch, Ralf Schlüter, Hermann Ney
Human Language Technology and Pattern Recognition Group
Chair of Computer Science 6
RWTH Aachen University, D-52056 Aachen, Germany
{*kozielski,matysiak,doetsch,schluter,ney*}*@cs.rwth-aachen.de*

***Abstract*—In this paper we investigate different n-gram language models that are defined over an open lexicon. We introduce a character-level language model and combine it with a standard word-level language model in a backoff fashion. The character-level language model is redefined and renormalized to assign zero probability to words from a fixed vocabulary. Furthermore we present a way to interpolate language models created at the word and character levels. The computation of character-level probabilities incorporates the across-word context. We compare perplexities on all words from the test set and on in-lexicon and OOV words separately on corpora of English and Arabic text.**

I. Introduction

The problem of out-of-vocabulary (OOV) words has been a focus of an extensive study in the domain of language modeling and speech recognition. The goal of this study is to define a language model (LM) that assigns non-zero probabilities to words that are not included in a fixed vocabulary. Furthermore, the LM must fulfill normalization constraints to allow for correct perplexity measurements and comparisons.

Brakensiek [1] explores the usage of character n-grams in the domain of postal address recognition. In that domain the OOV rate tends to be very high because of the large amount of street and person names for which it is hard to obtain a sufficient dictionary. The performance in a recognition experiment did not however outperform the closed-vocabulary baseline.

Creutz [2] decomposes words into sub-word units called morphemes. The experiments were run for highly-inflecting languages, i.e. languages that rely strongly on modification of words based on tense, person, etc. In speech recognition experiments a morpheme-based LM performed better than a standard word n-gram model by 20% in relative word error rate.

Bisani [3] approaches the OOV problem by using a hybrid lexicon consisting of graphones and full words. After decoding in a speech recognition experiment, the fragments are concatenated greedily to form words. This hybrid approach performed better than the corresponding baseline. Vertanen [4] uses a hybrid LM similar to the one described in [3]. Additionally, a confusion network is introduced after the decoding process to further improve the recognition results. Shaik [5] uses a hybrid LM that contains full words, morphemes, and graphones. The intention is to use full words for the most frequently appearing words, morphemes for less frequent words, and graphones for OOV recognition.

Hazen [6] introduces a generic OOV model that allows for any sequence of characters during recognition. The dependencies between characters are captured using an additional character-level LM. Similarly, Kozielski [7] uses the character-level LM to explicitly recognize and hypothesize OOV words. In those LMs, the word-level and character-level contexts are clearly separated from each other as opposed to the hybrid approaches.

In this paper we describe and compare different approaches to define an n-gram LM over an open lexicon. In Section II we introduce a standard backoff combination between the word-level and character-level language models and describe approaches to redefine the character-level LM to fulfill the normalization constraint. We also show how to implement the model using a lexical prefix tree. In Section III we introduce a character-level LM that makes use of the across-word context and interpolate it with a standard word-level LM. In Section IV we report perplexity measurements on text sources in English and Arabic.

II. Backoff language model

Let W be the total word space, $V \subseteq W$ the fixed lexicon (vocabulary), and $\mathcal{C}$ the character inventory which includes the word end symbol $\#$. We denote the function that maps a word to a corresponding sequence of characters as $\hat{c} : W \to \mathcal{C}^*$, and the opposite function $\hat{w} : \mathcal{C}^* \to W$.

We define an LM that assigns a prior probability to a word sequence $w_1^N := w_1, \ldots, w_n$:

$$q(w_1^N) = \prod_{i=1}^{N} q(w_i | w_{i-n+1}^{i-1}) = \prod_{i=1}^{N} q(w_i | h_i) \qquad (1)$$

where $w_i \in W$ is any word from the word space and $h_i = w_{i-n+1}, \ldots, w_{i-1}$ is the context of $n-1$ words.

The typical n-gram $p(w_i|h_i)$ assigns zero probability to OOV words $w_i \notin V$. The smoothing procedure lets us reserve some of the probability mass for unseen words. Here however we want to smooth over the set of all possible OOV words which is of infinite size.

We therefore introduce a generic token w_{oov} that represents all OOV words and that accumulates the whole OOV probability mass. We then represent an OOV word as a sequence of characters. We can now capture the a priori knowledge of dependencies between characters by constructing a second m-gram model of order m on character level. The probability of a sequence of characters $c_1^M \in \mathcal{C}^*$ is computed as:

$$p(c_1^M) = \prod_{j=1}^{M} p(c_j | c_{j-m+1}^{j-1}) \tag{2}$$

The character-level LM is normalized over all sequences of all lengths M, because we include the word-end symbol $\# \in \mathcal{C}$ at the end of every character sequence.

The final probability of a word is defined by combining the word-level and character-level LMs in a backoff fashion:

$$q(w_i|h_i) = \begin{cases} p(w_i|h_i) & \text{if } w_i \in V \\ p(w_{\text{oov}}|h_i)p(\hat{c}(w_i)) & \text{if } w_i \notin V \end{cases} \tag{3}$$

Whenever we encounter an OOV word, it is retained in the context h_i as the w_{oov} token. In fact both models can hypothesize any word from the word space, but the probability for an OOV word assigned by the word-level LM is zero.

The probability mass of the character-level LM is distributed over all words from the word space and not only over OOV words. Instead we want to use a character-level LM $\bar{p}(c_1^M)$ that assigns zero probability to in-lexicon words. Based on the model in Eq. (2), different approaches to redefine the character-level LM can be proposed, as described in the following.

A. Character-level LM with early subtraction

For a given character prefix c_1^j we define a set of suffixes that together with the prefix constitute in-lexicon words:

$$S(c_1^j) = \{c_{j+1}^M : \hat{w}(c_1^M) \in V\} \tag{4}$$

We define a function that for a given character context isolates the probability mass of the character-level LM which is contributed by the suffixes of in-lexicon words:

$$\beta(c_1^j) = \sum_{M, c_{j+1}^M \in S(c_1^j)} p(c_{j+1}^M | c_1^j) \tag{5}$$

Then we exclude this probability mass from every m-gram of the character-level LM and renormalize:

$$\bar{p}(c_j|c_1^{j-1}) = p(c_j|c_{j-m+1}^{j-1}) \frac{1-\beta(c_1^j)}{1-\beta(c_1^{j-1})} \tag{6}$$

The context is not limited to the last $m-1$ characters but depends on the complete within-word history c_1^{j-1}.

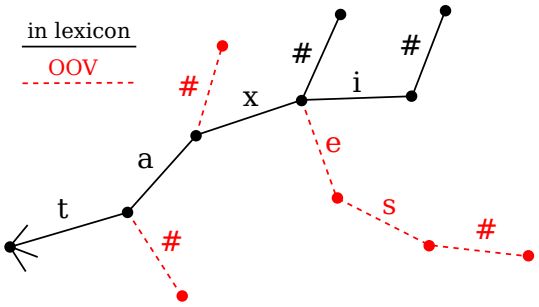

Figure 1: Illustration of the lexical prefix tree. Solid, black nodes and arcs demonstrate common prefixes for both in-lexicon and OOV words. Dashed, red nodes and arcs illustrate OOV words, outside of the common part of the tree. Once we traverse a red arc, it is impossible to arrive at a black arc again.

$$\bar{p}(c_1^M) = \prod_{j=1}^{M} \bar{p}(c_j|c_1^{j-1}) \tag{7}$$

Such a formulation explicitly excludes in-lexicon words. Note the following for all in-lexicon words $\hat{w}(c_1^{M-1}\#) \in V$:

$$\beta(c_1^{M-1}\#) = 1 \tag{8}$$

Additionally this LM performs an effective look ahead, because the probability mass coming from in-lexicon words is isolated and subtracted as early as possible. The values of β can be efficiently precomputed by using the following recurrent equation:

$$\beta(c_1^j) = \sum_{\tilde{c} \in \mathcal{C}} p(\tilde{c}|c_1^j) \beta(c_1^j \tilde{c}) \tag{9}$$

B. Character-level LM without early subtraction

In the renormalized character-level LM defined in the previous section the probability mass is shifted away from in-lexicon words. Alternatively we can define the model $\bar{p}(c_1^M)$ by just excluding the word-end symbols of in-lexicon words and without shifting the probability mass:

$$\beta(c_1^j) = \begin{cases} 1 & \text{if } c_j = \# \wedge \hat{w}(c_1^j) \in V \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$\bar{p}(c_j|c_1^{j-1}) = \frac{p(c_j|c_{j-m+1}^{j-1})\left[1-\beta(c_1^j)\right]}{1-p(\#|c_{j-m+1}^{j-1})\beta(c_1^{j-1}\#)} \tag{11}$$

Note that the probabilities do not change for contexts that do not form an in-lexicon word.

$$\forall c_1^{j-1} : \hat{w}(c_1^{j-1}\#) \notin V \quad \bar{p}(c_j|c_1^{j-1}) = p(c_j|c_{j-m+1}^{j-1}) \tag{12}$$

C. Normalization constraints

The word-level and character-level LMs are normalized over all words from the word space by definition:

$$\sum_{w \in W} p(w|h) = 1 \quad \sum_{w \in W} p(\hat{c}(w)) = 1 \tag{13}$$

The backoff combination defined in Eq. (3) is not properly normalized because the probability mass assigned to in-lexicon words by the character-level LM is lost. The character-level LMs defined in sections II-A and II-B do not suffer from this limitation but is normalized over all out-of-vocabulary words:

$$\sum_{w \in W} \bar{p}(\hat{c}(w)) = \sum_{w \notin V} \bar{p}(\hat{c}(w)) = 1 \tag{14}$$

which leads to the desired:

$$\sum_{w \in W} q(w|h) = 1 \tag{15}$$

Because the normalization constraints are fulfilled, it is possible to compute perplexities over the whole test set (both in-lexicon and OOV words) using the proposed model.

D. Implementation using lexical prefix tree

The renormalized character-level LM $\bar{p}$ can be represented as a big, infinite lexical prefix tree, which is illustrated in Figure 1. The nodes of this tree represent character contexts; The arcs are weighted with the renormalized probabilities $\bar{p}$; And the leafs denote word ends.

We construct the lexical prefix tree by first adding all words from the fixed lexicon. We call this part a common one, because every node in this part of the tree represents a prefix that can constitute either an in-lexicon or an OOV word. Furthermore every node in the common part contains outgoing arcs labelled with characters that further lead to the completion of an in-lexicon word.

To make this tree complete we add missing arcs such that from every node we can traverse an arc that is labeled with any character from the character inventory. Those new arcs lead to nodes of the tree that represent prefixes of only OOV words. Once we leave the common part of the tree we reach a prefix that can constitute solely an OOV word.

The character-level LM does not need to be renormalized for contexts outside of the common part of the lexical prefix tree and the computation of probabilities is then restrained to the context of $m-1$ previous characters:

$$\forall c_1^j : S(c_1^j) = \varnothing \quad \bar{p}(c_j|c_1^{j-1}) = p(c_j|c_{j-m+1}^{j-1}) \tag{16}$$

The common part of the lexical prefix tree is finite and can be generated statically. All arcs beyond the common part are expanded dynamically using the original m-gram character-level LM which can be implemented using a finite table. That is why the implementation of such an infinite lexical prefix tree is feasible.

E. Max and sum backoff

In [7] we proposed the following way of combining the LMs:

$$q(w_i|h_i) = \max\{p(w_i|h_i),\, p(w_{\text{oov}}|h_i)p(\hat{c}(w_i))\} \tag{17}$$

This approach uses the original character-level LM as defined in Eq. (2). Because the character-level LM assigns some probability also to in-lexicon words, it is possible that this probability will be actually higher than the corresponding word-level probability. The use of the maximum instead of the condition as in Eq. (3) corresponds to the implementation details of a decoder based on the weighted finite state transducers (WFST) [8].

Furthermore we can rewrite Eq. (17) to use the sum instead of the maximum:

$$q(w_i|h_i) = p(w_i|h_i) + p(w_{\text{oov}}|h_i)p(\hat{c}(w_i)) \tag{18}$$

By definition the use of the sum yields higher probabilities than the use of the maximum, which in turn yields higher probabilities than the use of the condition as defined in Eq. (3). Because Eq. (18) is normalized, which can be easily shown, then Eq. (17) loses some probability mass and does not fulfill the normalization constraint. We include those methods here for a comparison.

III. Interpolated language model

In the backoff approach the weighting of the word-level and character-level LMs is imposed by the backoff weights of the word-level LM (which are different for every n-gram). However, the usual smoothing procedure treats the w_{oov} token as an ordinary word and does not take into account that in our scenario it is used to represent a class of words. Instead we investigate a combination that is based only on a single, global parameter.

The probability of a word is defined by interpolating the word-level and character-level LMs:

$$q(w_i|h_i) = \lambda \cdot p(w_i|h_i) + (1-\lambda) \cdot p(\hat{c}(w_i)) \tag{19}$$

where $\lambda \in [0, 1]$ is the interpolation weight. Here the word-level LM does not include the w_{oov} token. If it encounters an OOV word in the context h_i it uses the lower order distribution directly (because the backoff weight for an unseen context is 1). The contribution for an OOV word assigned by the word-level LM is zero.

So far we have excluded the preceding words for the computation of the character-level probabilities. Now we want to reformulate the character-level LM introduced in Eq. (2) to include the across-word context. The probability of a sequence of characters $c_m^M \in \mathcal{C}^*$ given a preceding sequence $c_1^{m-1} \in \mathcal{C}^*$ is computed as:

$$p(c_m^M|c_1^{m-1}) = \prod_{j=m}^{M} p(c_j|c_{j-m+1}^{j-1}) \tag{20}$$

We denote the function that maps a sequence of words to a sequence of characters and truncates it to the last $m-1$ characters as $\hat{c}_m : W \rightarrow \mathcal{C}^*$. The sequence of characters includes the word-end symbol $\# \in \mathcal{C}$ after each word. The final interpolated probability of a word is defined as:

$$q(w_i|h_i) = \lambda \cdot p(w_i|h_i) + (1-\lambda) \cdot p(\hat{c}(w_i)|\hat{c}_m(w_1^{i-1})) \quad (21)$$

The word-level and character-level LMs are normalized over all words from the word space – recall Eq. (13) – so is the interpolated LM by definition. This is an advantage over the backoff approach in which we have to redefine the character-level LM to fully fulfill the normalization constraint. The parameter λ should be always optimized on a development set.

IV. Experiments

We evaluate our model by comparing the perplexities (PPL) on word and character level on all words from the test set and on in-lexicon and OOV words separately. The lower the perplexity the better the language model. The computation of the character-level perplexities includes the word-end symbol after every word (even the last one). We revisit the definition of the character-level perplexity:

$$\mathrm{PPL} = -\frac{1}{N_c} \sum_{i=1}^{N} \log q(w_i|h_i) \quad (22)$$

where N_c is the length of the sentence w_1^N measured as the number of characters. Please note that the word-level perplexities can be always recomputed to give the character-level perplexities and vice-versa.

A. Datasets

We used text sources in English and Arabic separately. The English corpus consists of around 3M running words and has been built upon the combined LOB [9], Brown [10], and Wellington [11] corpora. The test set contains 8k running words and the OOV rate is 2.6% The Arabic corpus consists of around 20M running words taken from Arabic newspapers like: Addustour, Alahram, Albayan, Alittihad, Alwatan, Alraya; in addition to audio transcriptions for GALE project's BN and BC data, along with some web text. The test set contains 20k running words and the OOV rate is 1%.

B. Language models

As the word-level LM we use a 3-gram model with modified Kneser-Ney discounting built upon the training text source containing one sentence per line. We always use the largest possible vocabulary for training of the word-level LM. As the character-level LM we use a 10-gram model with Witten-Bell discounting. We cannot use the standard modified Kneser-Ney method because of lack of singletons in the training data. We take different approaches to creating the training text source for the character-level LM. For all backoff approaches we use a list of words, one word per line, split into separate characters. The word list includes only OOV words extracted from the training source with their frequency counts. We use different vocabulary sizes for that purpose; a vocabulary of size zero means that all words were used. For the interpolated approach with across-word context we use a list of sentences, one sentence per line, split into separate characters and with the word-end symbol after each word. Because of the across-word context the character-level LM has to be trained on whole sentences (similarly to the word-level LM) and not on separate words.

Table I: English - Comparison of the perplexities between different methods.

language model	char PPL			word PPL
	in-lex	OOV	total	total
word-level only	3.403	–	–	–
char-level only	4.015	20.231	4.320	1084.4
backoff				
- condition	3.438	32.020	3.811	595.6
- maximum	3.438	32.020	3.811	595.6
- sum	3.437	32.020	3.811	595.3
- w/o early sub.	3.438	31.524	3.808	593.5
- with early sub.	3.438	32.319	3.813	596.9
interpolated				
- w/o context	3.442	21.342	3.742	545.9
- with context	3.406	23.965	3.726	534.5

Table II: Arabic - Comparison of the perplexities between different methods.

language model	char PPL			word PPL
	in-lex	OOV	total	total
word-level only	3.378	–	–	–
char-level only	3.680	19.302	3.722	1438.9
backoff				
- condition	3.394	18.860	3.438	927.5
- maximum	3.394	18.860	3.437	926.7
- sum	3.387	18.860	3.431	917.6
- w/o early sub.	3.394	18.569	3.437	926.4
- with early sub.	3.394	18.880	3.438	927.6
interpolated				
- w/o context	3.393	19.488	3.438	928.1
- with context	3.349	23.846	3.404	878.1

C. Experimental results

Tables I and II show the comparison of perplexities between different methods on the corpora for English and Arabic. The condition and maximum backoffs are not prop-

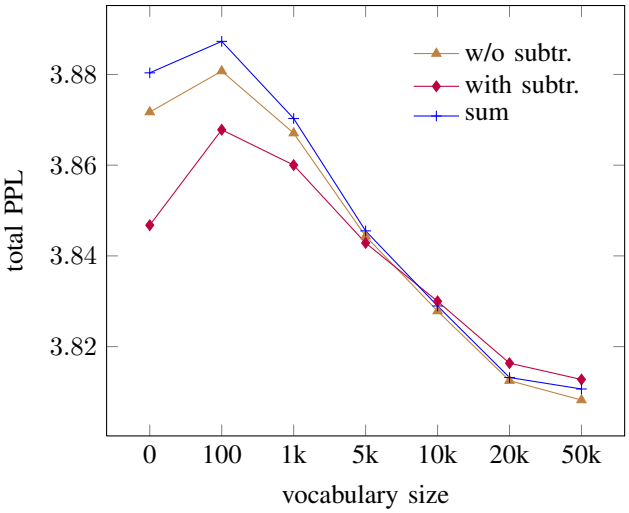

Figure 2: English - The total character-level perplexities obtained by using the backoff approaches with respect to the size of the vocabulary.

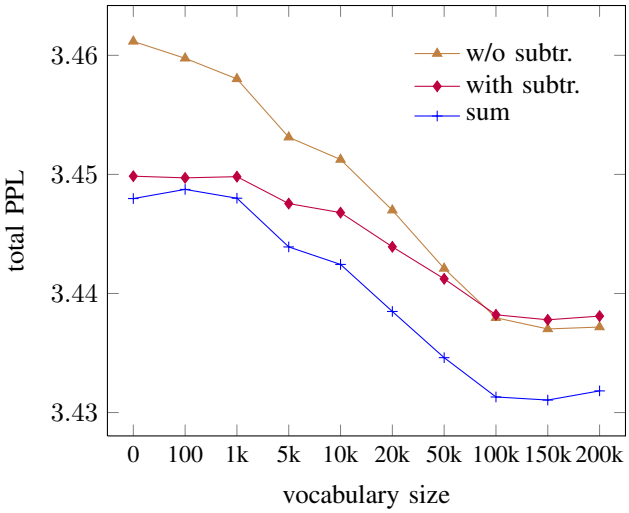

Figure 3: Arabic - The total character-level perplexities obtained by using the backoff approaches with respect to the size of the vocabulary.

erly normalized and thus cannot be properly compared, because some probability mass is being lost, as explained in Section II-C. The presentation was broken down to in-lexicon words, OOV words, and all words in total. We also include results that were obtained using solely the word-level or a character-level LM. The interpolated LM with across-word context clearly outperforms other approaches on both corpora. On the text in English even the interpolated LM without across-word context is better than all backoff approaches. The perplexities computed over OOV words (even when using only the character-level LM) where much higher than those computed over in-lexicon words, because the OOV words are in general much less regular in structure than the in-lexicon words.

Figures 2 and 3 show the perplexities obtained by using the backoff approaches with respect to the size of the vocabulary for English and Arabic. For the clarity of the presentation we included only the sum backoff and the backoff approaches with the redefined character-level LM. The larger the vocabulary, the smaller the number of OOV words in the word list used for training of the character-level LM. The results show that the sum backoff outperforms other backoff approaches on the text in Arabic, but the backoff with character-level LM without early subtraction was slightly better on the text in English. We also found out that using only the less frequent words from the vocabulary for training of the character-level LM improves the performance, which confirms the findings of [7]. The combination with the character-level LM with early subtraction was in general the worst approach, but this approach improves if more in-lexicon words are used in training. The word-level LM did not change throughout those experiments as it was created using the largest vocabulary possible. Using a smaller vocabulary for training of the word-level LM always had a negative impact on the performance.

Figures 4, 5 and 6 show the perplexities obtained using the interpolated LM with respect to interpolation weight λ. The presentation was broken down to in-lexicon words, OOV words, and all words in total. The inclusion of across-word context shifts the optimal interpolation weight towards the character-level LM, from $\lambda = 0.97$ to $\lambda = 0.77$; as this model becomes more confident. It is important to note that the interpolation with the character-level LM with across-word context improves also the perplexity computed over exclusively in-lexicon words, which means that also tasks with zero OOV rate can benefit from the use of this method. The curve in Figure 5 is strictly increasing as there is no contribution from the word-level LM for OOV words.

V. Conclusions

We have shown that the interpolation of word-level and character-level LMs gives better perplexities than the combination of them in a backoff fashion. The incorporation of the across-word context on character-level significantly improves the results of the interpolated LM. Moreover we have demonstrated that the original backoff approach can be improved by redefining the character-level LM to assign zero probability to in-lexicon words. Finally we have shown that the inclusion of the character-level LM improves the performance even on tasks with zero OOV rate.

Acknowledgments. This work was partially supported by a Google Research Award and by the Quaero Programme, funded by OSEO, French State agency for innovation.

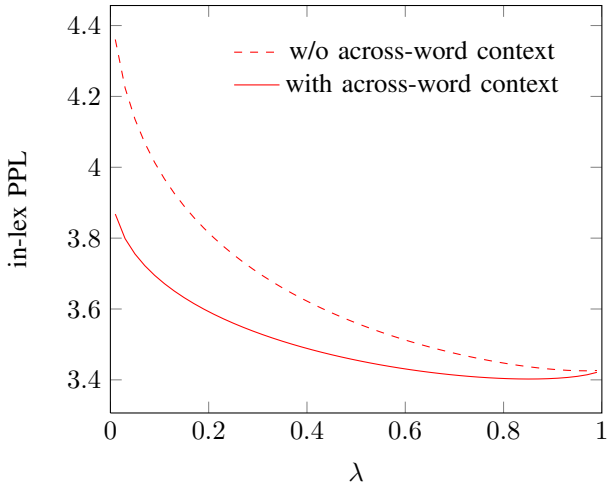

Figure 4: English - The in-lexicon character-level perplexities of the interpolated LM with respect to the interpolation weight.

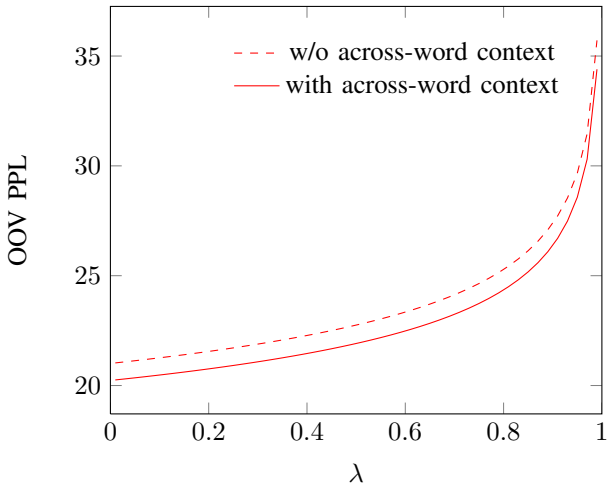

Figure 5: English - The out-of-vocabulary character-level perplexities of the interpolated LM with respect to the interpolation weight.

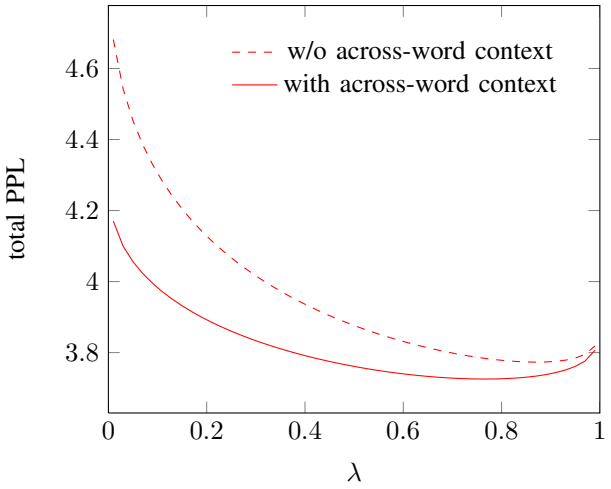

Figure 6: English - The total character-level perplexities of the interpolated LM with respect to the interpolation weight.

References

[1] A. Brakensiek, J. Rottland, and G. Rigoll, "Handwritten address recognition with open vocabulary using character n-grams," in *Frontiers in Handwriting Recognition (IWFHR). Proceedings. Eighth International Workshop on*, 2002, pp. 357–362.

[2] M. Creutz, T. Hirsimäki, M. Kurimo, A. Puurula, J. Pylkkönen, V. Siivola, M. Varjokallio, E. Arisoy, M. Saraçlar, and A. Stolcke, "Morph-based speech recognition and modeling of out-of-vocabulary words across languages," *ACM Trans. Speech Lang. Process.*, vol. 5, no. 1, pp. 3:1–3:29, Dec. 2007.

[3] M. Bisani and H. Ney, "Open vocabulary speech recognition with flat hybrid models," in *Interspeech*, Lisbon, Portugal, Sep. 2005, pp. 725–728.

[4] K. Vertanen, "Combining open vocabulary recognition and word confusion networks," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, Apr. 2008, pp. 4325–4328.

[5] M. A. Basha Shaik, A. El-Desoky Mousa, R. Schlüter, and H. Ney, "Hybrid language models using mixed types of sub-lexical units for open vocabulary german LVCSR," in *Interspeech*, Florence, Italy, Aug. 2011, pp. 1441–1444.

[6] T. Hazen and I. Bazzi, "A comparison and combination of methods for OOV word detection and word confidence scoring," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 1, 2001, pp. 397–400.

[7] M. Kozielski, D. Rybach, S. Hahn, R. Schlüter, and H. Ney, "Open vocabulary handwriting recognition using combined word-level and character-level language models," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2013.

[8] D. Rybach, H. Ney, and R. Schlüter, "Lexical prefix tree and WFST: A comparison of two dynamic search concepts for LVCSR," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 6, pp. 1295 –1307, Jun. 2013.

[9] S. Johansson, E. Atwell, R. Garside, and G. Leech, *The Tagged LOB Corpus: User's Manual*, Norwegian Computing Centre for the Humanities, 1986.

[10] W. Francis and H. Kucera, "Brown corpus manual, manual of information to accompany a standard corpus of present-day edited American English," Tech. Rep., 1979.

[11] L. Bauer, "Manual of information to accompany the Wellington corpus of written New Zealand English," Tech. Rep., 1993.