PERFORMANCE ANALYSIS OF NEURAL NETWORKS IN COMBINATION WITH N-GRAM LANGUAGE MODELS

Ilya Oparin[1], *Martin Sundermeyer*[2], *Hermann Ney*[2], *Jean-Luc Gauvain*[1]

[1]LIMSI CNRS, Spoken Language Processing Group
[2]RWTH Aachen, Computer Science Department, Human Language Technology and Pattern Recognition
{oparin,gauvain}@limsi.fr, {sundermeyer,ney}@informatik.rwth-aachen.de

ABSTRACT

Neural Network language models (NNLMs) have recently become an important complement to conventional n-gram language models (LMs) in speech-to-text systems. However, little is known about the behavior of NNLMs. The analysis presented in this paper aims to understand which types of events are better modeled by NNLMs as compared to n-gram LMs, in what cases improvements are most substantial and why this is the case. Such an analysis is important to take further benefit from NNLMs used in combination with conventional n-gram models. The analysis is carried out for different types of neural network (feed-forward and recurrent) LMs. The results showing for which type of events NNLMs provide better probability estimates are validated on two setups that are different in their size and the degree of data homogeneity.

Index Terms— Neural network, language model, STT.

1. INTRODUCTION

Having been used for several decades, n-gram language models (LMs) still form the basis of modern language modeling for speech-to-text (STT). There are very few approaches that were shown to systematically bring additional improvements over n-gram baselines and are thus used in large-scale STT systems. Standard n-gram LMs rely on a discrete space representation of the vocabulary, each word being associated with a discrete index, while Neural Network (NN) LMs are based on the idea of word representation in a continuous space.

Recent advances with NNLMs deal either with feed-forward (FFNN) or recurrent neural networks (RNNs). The use of FFNNs for language modeling was introduced in [1] and successfully applied to speech recognition in [2]. The complexity of inference and training remains a major difficulty, which depends mainly on the size of training data and the output vocabulary. Thus one of the current research directions deals with the factored representation of the vocabulary and better use of training data [3]. Another direction is the possibility to combine different information within NNLMs, such as morphology or character information [4, 5]. Significant and systematic improvements (in interpolation with baseline Kneser-Ney (KN) n-gram LMs) were reported for state-of-the-art STT systems with the FFNNs cited above.

Elman networks ([6]) can be viewed as NNs with the unlimited context. While in FFNNs the current state of the hidden layer depends on the layer formed by projecting a fixed n-gram context into a low-dimensional space, in Elman NNs it depends on the last observed word and the state of the hidden layer before this observation. This way, there is no explicitly defined context, and history is captured implicitly by the recurrent nature of the model. Elman NNs have recently been introduced in language modeling for STT under the name of recurrent networks [7], that is used in this paper. RNNs showed excellent performance on different tasks. Currently the lowest perplexity on the widely-used Penn Treebank corpus is reported with RNNs interpolated with a KN LM [8].

Results obtained with neural networks in STT systems gave birth to a whole series of publications and currently made NNs one of the most promising directions of research in language modeling. At the same time the question why NNLMs appear to improve over n-gram baselines is still open. Contrary to n-gram models, it is not easy to make predictions of the performance of NNLMs for individual observations. The most common explanations do not go further mentioning that "similar" words, when projected into a low-dimensional continuous space, get similar representations that result in similar probability estimates. This explanation looks rather general and does not give real clues to the question how the combination of NNLMs and n-gram LMs can be improved. That is why the more detailed analysis of NNLM performance, that might be beneficial for research in this direction, is presented in this paper.

2. UNDERLYING IDEAS

The goal of this work is to give an insight on how the combination of NN and n-gram LMs may be improved. The NNLM performance is analyzed jointly with the baseline n-gram LMs. The model supposed to do better for a given event (a word or sentence boundary token in test data) is the one that provides a higher probability.

Our most basic intuition was that NNLMs should do better in cases a KN LM backs off for probability estimation. It is known what backoff level is used by the n-gram LM for each word in test data. Thus, it is possible to analyze separately the way NNLMs performs for the events which probabilities are estimated with an n-gram LM without backoff, with a backoff

to trigrams, bigrams or unigrams. We call it *backoff levels*. It is important to notice that the backoff level is the order used by an n-gram LM to estimate the probability for each given event in test. The statistics investigated are

- percentage of events for which NNLMs provide higher probabilities;
- per-level perplexities;
- sum of absolute differences in probabilities provided by NN and KN LMs for each event (showing how important the gain is).

The per-level perplexities are computed in such a way that

$$\exp\left(\sum_k \frac{N_k}{N}\ln(PPL_k)\right) = PPL \quad (1)$$

holds, where N_k is the number of events at the given backoff level k, N is the total number of events at all backoff levels, PPL_k is the per-level perplexity and PPL is the general perplexity calculated in the usual way. It is easy to see that per-level perplexities are obtained as

$$PPL_k = \exp\left(-\frac{1}{N_k}\sum_{w_k}\ln P(w_k|h_k)\right) \quad (2)$$

where w_k are the events in test data estimated with an n-gram LM at the k^{th} backoff level, h_k being corresponding histories.

It seems necessary to perform a more detailed count-based analysis. This is motivated by the way probabilities are estimated in n-gram LMs. For example, the value of the discount D used in the Kneser-Ney scheme (see [9]) may in practice be close to 1 which means that all singleton events obtain very low probabilities. In this case NNLMs may provide better predictions. Another special case is the one when the n-gram counts are equal to the history counts. Thus, count-based parameters are introduced in our analysis, such as

- n-gram count $c(hw)$;
- history count $c(h)$;
- number of different words following n-gram history $N_{1+}(h\cdot)$, where $N_{1+}(h\cdot) = |\{w_i : c(hw_i) > 0\}|$.

3. EXPERIMENTAL RESULTS

3.1. Experimental Setup

Two different setups are used to perform the analysis presented in this paper. English *Penn* Treebank portion of the Wall Street Journal corpus is a small homogeneous corpus that is commonly used to compare results across research sites. However, it is also important to verify the results on a much larger and much more heterogeneous corpus. The French transcriptions of broadcast conversations were chosen as this larger test corpus. Different in their nature fast transcriptions of broadcast news were chosen for training the models for this setup. These data were preprocessed and normalized by Vocapia Research for LIMSI/Vocapia submission to the *Quaero* 2011 STT evaluation. Corpora characteristics are presented in Table 1. More information on STT systems for the Quaero 2010 evaluation can be found in [10].

The FFNNs are trained with the context length of 3 words, projection and hidden layer sizes equal to 300 and 500. A shortlist of 12k most frequent words is used on the *Quaero* corpus with resampling of training data with the coefficient 0.25 at each training epoch. The resampling is necessary due to the large size of the training data. No shortlists and resampling are used for the *Penn* NNs. The recurrent NNs are trained with the BUT RNN toolkit with 500 hidden nodes and back-propagation through time (see [7] for details).

corpus	*train size*	*test size*	*vocabulary size*	*OOV*
Penn	930k	82k	10k	6.1%
Quaero	92M	2.3M	200k	0.1%

Table 1. *Characteristics of the corpora.*

The baseline n-gram LMs are KN-discounted (backoff and interpolated versions) 4-gram models trained on all the training data without pruning and cut-offs. Use of the backoff version of Kneser-Ney discounting, as compared to the interpolated one, is motivated by the possibility of having a clearer picture for different backoff levels and counts.

3.2. General Analysis

The general statistics for different setups and models are presented in Table 2 where, *LM* stands for a KN n-gram model with indices *b* and *i* corresponding to its backoff and modified interpolated versions. The number of events at each level of backoff (as described in Section 2) is presented in the column *#e* (excluding OOVs). The percentage of the events for which a NN provides higher probabilities than a KN LM is shown in the column *%NN* in general and at each backoff level separately. General and per-level perplexities of NN and KN LMs are presented in the columns *pp KN* and *pp NN*. The last column corresponds to the sum of differences in probabilities assigned to each event by NN and KN LMs. Positive number means an NNLM provides higher probabilities than an n-gram model for given type of events, and vice versa.

Different NNs provide higher probabilities for about half of the n-grams for both setups (see columns *%NN*). The percentage grows for the n-grams for which the KN LM backs off. On the *Penn* corpus NNLMs do steadily better if a deeper backoff is used by the KN LM. On the *Quaero* corpus the tendency is the same, with the only difference for the backoff version of KN LM, for which a NNLM provides higher probabilities in similar number of cases at trigram and bigram levels. If one looks at the perplexity results at different backoff levels, it is also seen that n-gram LMs are characterized by lower perplexities for the events estimated without backing off (*4gr* row) and NNLMs, in turn, give better perplexities for the events for which a KN LM is backing off.

We also repeated the experiments in combination with the 3-gram KN LMs and obtained similar results, i.e. n-gram LMs perform better for the events for which no backoff is used and the NNLMs tend to do better as backoff gets deeper.

3.3. Count-Based Analysis

Different count-based combinations of parameters mentioned in Section 2 were tried with the focus on the tendencies that

	LMs	level	#e	%NN	pp KN	pp NN	ΔP
Penn	FFNN-iLM	all	77628	51.5	164	163	-52
		4gr	12706	33.7	6	10	-717
		3gr	17823	42.2	29	43	-46
		2gr	31720	55.2	227	210	637
		1gr	15379	69.2	9821	4522	73
	RNN-iLM	all	77628	57.8	164	143	1251
		4gr	12706	46.0	6	8	-70
		3gr	17823	51.8	29	35	369
		2gr	31720	60.6	227	189	864
		1gr	15379	68.8	9821	4443	88
Quaero	FFNN-bLM	all	2310325	55.9	166	148	-3344
		4gr	791460	39.4	18	23	-22619
		3gr	720661	63.5	124	93	10627
		2gr	625211	63.0	718	539	7351
		1gr	172993	73.6	74495	47114	1298
	FFNN-iLM	all	2310325	53.6	156	148	-12037
		4gr	791460	33.0	16	23	-28981
		3gr	720661	58.6	99	93	7671
		2gr	625211	66.2	766	539	7963
		1gr	172993	81.2	115533	47114	1308

Table 2. *Statistics on the combination of NN and KN LMs.*

hold for all the NNLMs on both setups, in comparison with the interpolated version of the KN LMs. Due to the drastic lack of space in this short paper only the most pronounced ones are presented in Table 3. During the analysis these results are compared to the general statistics in Table 2.

First, contrary to our expectations, there is no clear evidence that NNLMs do always better for the events that are estimated with KN LM n-grams based on singleton counts (see column "$c(hw) = 1$"). At the same time, n-gram LMs give better estimates for the n-grams with the count equal to their history counts (see column "$c(w, h) = c(h)$"). The number of events that get higher probabilities with NNLM (*%NN*) and probability difference sum (ΔP) is lower that in general (compare to Table 2).

As a generalization of this case, the same tendency holds for the events with a small number of unique words that can follow its n-gram history (see column "$N_{1+}(h\cdot) < \theta$" where $\theta = 10$). The other way round, NNLMs tend to do better if many different words were seen for an event's n-gram history in the training data, as seen from the "$N_{1+}(h\cdot) > \theta$" column. It is interesting to note that most of this improvement comes from the highest order for which KN LMs are not expected to provide very reliable estimates in such cases. The values of thresholds θ are corpus-dependent and should be tuned for different tasks. Here we illustrate conclusions with θs equal to 10 for the *Penn* and 100 for the *Quaero* corpus due to their different sizes. KN LMs provide much higher (than in the general case without count-based constraints) probabilities for the n-grams with high history counts and only one possible word following the history, according to the training data, as can be seen from the column "$c(h) > 1, N_{1+}(h\cdot) = 1$". KN n-gram LMs also perform better for the events that are not singleton and have a limited history count (see "$c(hw) > 1, N(h) < \theta$").

The NNLM performance for the events that obtain (with a KN LM) probabilities higher or lower predefined thresholds was also investigated. The NNLMs do well for the n-grams with a high number of unique words that can follow the history and with low-probabilities assigned by an n-gram model, as shown in the column "$P < \eta, N_{1+}(h\cdot) > \theta$" (as illustrated with η equal to 0.1), both on the level of percentage of the events that obtain higher probabilities with a NNLM (*%NN*), and the probability difference sum (ΔP). The other way round, n-gram LMs do better for high-probability non-singleton n-grams with limited history counts, as illustrated in column "$P > \eta, c(hw) > 1, c(h) < \theta$".

	Stand-alone			Interpolated			
	KN	*FF*	*R*	*KN+FF*	*KN+R*	*R+FF*	*KN+FF+R*
all	143	142	126	115	106	115	103
-oov	164	163	143	132	120	130	117

Table 4. *Perplexities on the Penn Treebank corpus.*

During the analysis the regularities depending only on event history were of a particular interest. For example, if a NNLM performs better for the events with a particular history count $c(h)$, we could use it to implement context-dependent interpolation weights $\lambda(h)$ instead of one single λ for each LM, similar as proposed in [11] for the combination of multiple n-gram models built from different sources. However, most cases observed so far when an NNLM does better or worse than in general, as compared to an n-gram model, are conditioned both on the predicted word and its n-gram history. It should also be noted that further count-based analysis tightly coupled with the peculiarities of the KN smoothing scheme may be promising.

3.4. Recurrent vs. Feed-Forward NNLMs

The recurrent NNLM performs better than the 4-gram feed-forward NNLM. Perplexities of stand-alone and linearly interpolated models are presented in Table 4 (*KN* stands for the 4-gram interpolated KN LM, *FF* and *R* for the feed-forward and recurrent NNLMs) for all n-grams in test (*all*) and without n-grams ending with an OOV word (*-oov*). It can be seen that while the interpolation of the FFNN with the RNN helps to improve the stand-alone RNN perplexity, only minor improvements are attained after adding the FFNN to the combination of the KN and RNN LMs (this goes in line with the results reported in [8]). At the same time, as seen from Tables 2 and 3, the FFNN and RNN follow similar patterns to improve KN n-gram estimates for different types of events, that points out to the fact that these models are not likely to complement each other when interpolated with an n-gram LM.

4. CONCLUSIONS AND FUTURE WORK

In this paper we compared the performance of state-of-the-art feed-forward and recurrent NNLMs to that of conventional n-gram LMs. Detailed quantitative analysis on two different corpora was performed and, as a result, the types of cases for which NNLMs systematically provide higher or lower probability estimates as compared to a KN LM were defined. We believe these results are important to better understand the relations between NN and n-gram LMs and implement better interpolation schemes. The most general conclusions are:

			$c(hw)=1$			$c(hw)=c(h)$			$N_{1+}(h\cdot)<\theta$			$N_{1+}(h\cdot)>\theta$		
	LM	level	#e	%NN	ΔP	#e	%NN	ΔP	#e	%NN	ΔP	#e	%NN	ΔP
Penn	FFNN	all	14500	46.6	38	4881	24.6	-612	14442	33.4	-892	47181	51.3	780
		4gr	3945	35.6	-88	3478	20.8	-500	7753	27.1	-765	4740	44.3	57
		3gr	5616	45.7	79	1343	32.8	-108	5608	38.3	-152	11941	44.0	107
		2gr	4920	56.2	47	60	65.0	-5	1081	53.2	16	30500	55.3	618
		1gr	19	89.5	0	-	-	-	-	-	-	-	-	-
	RNN	all	14500	52.0	220	4881	38.2	-338	14442	43.6	-233	47181	58.7	1373
		4gr	3945	44.2	16	3478	35.7	-273	7753	40.4	-309	4740	54.9	234
		3gr	5616	51.7	151	1343	44.2	-56	5608	47.4	82	11941	53.9	274
		2gr	4920	58.5	53	60	51.7	-8	1081	46.1	-6	30500	61.2	865
		1gr	19	36.8	0	-	-	-	-	-	-	-	-	-
Quaero	FFNN	all	250532	50.1	-2202	37718	16.3	-7217	225624	29.7	-17973	1335358	58.5	10769
		4gr	122476	39.7	-3106	31618	14.2	-6557	173076	24.9	-17357	275312	37.7	-1577
		3gr	91613	56.3	751	5803	27.3	-590	48380	45.8	-513	479332	61.0	5493
		2gr	33242	66.6	154	297	16.8	-69	4168	42.0	-103	580714	66.3	6852
		1gr	3201	96.8	0	-	-	-	-	-	-	-	-	-
			$c(h)>1, N_{1+}(h\cdot)=1$			$c(hw)>1, c(h)<\theta$			$P<\eta, N_{1+}(h\cdot)>\theta$			$P>\eta, c(hw)>1, c(h)<\theta$		
	LM	level	#e	%NN	ΔP	#e	%NN	ΔP	#e	%NN	ΔP	#e	%NN	ΔP
Penn	FFNN	all	2733	19.2	-494	4089	29.2	-514	39174	51.8	503	3969	29.1	-516
		4gr	1999	14.5	-396	2301	24.0	-383	2444	43.3	35	2272	24.2	-381
		3gr	674	29.2	-94	1571	34.9	-124	9314	43.6	68	1501	34.8	-127
		2gr	60	65.0	-5	217	42.4	-8	27416	55.3	400	196	42.3	-8
	RNN	all	2733	37.1	-300	4089	38.8	-305	39174	58.4	737	3969	38.8	-310
		4gr	1999	33.6	-230	2301	36.3	-238	2444	52.7	83	2272	36.6	-237
		3gr	674	46.4	-62	1571	44.3	-46	9314	53.9	132	1501	44.2	-51
		2gr	60	51.7	-8	217	25.3	-22	27416	60.4	522	196	23.5	-22
Quaero	FFNN	all	20150	16.0	-4212	291607	33.2	-17639	1206403	59.0	7510	190865	26.6	-18583
		4gr	17408	14.6	-3788	203976	26.1	-17603	204650	38.1	-487	150510	22.2	-17482
		3gr	2610	25.4	-381	76658	48.5	-199	432699	60.0	2886	37103	43.2	-996
		2gr	132	22.7	-41	10973	58.4	163	569054	65.9	5111	3252	42.8	-104

Table 3. *Count-based example statistics on the combination of NN and n-gram LMs.*

- NNLMs improve over n-gram LMs for the cases when the latter backs off to lower orders. The gain tends to increase with the back off depth.
- Recurrent NNLM performs better than the 4-gram feed-forward NNLM. At the same time, both models follow the same patterns to improve n-gram estimates for different types of events, that can hardly make possible gains of using both RNN and FFNN to improve a KN LM additive.
- The above tendencies are valid for both backoff and interpolated versions of smoothing used in n-gram LMs.
- It was shown that there exist regular cases when NNLMs do better or worse than KN models (as compared to the general statistics). These regularities, as shown in Table 3, are conditioned on different parameters that deal with count characteristics of n-grams according to the training data. However, it should be admitted, that the count-based analysis does not provide a picture that is fully clear and further research in this direction dealing with peculiarities of the Kneser-Ney smoothing may bring new insights.

For other directions of future research, we consider addressing the issue of interpolation schemes that take better account of peculiarities of NN and KN LMs that were shown in this paper. Another direction is to perform a systematic stand-alone NNLM analysis (not bounded to the comparison with n-gram LMs) that takes into account word identities and their clustering in a continuous space.

ACKNOWLEDGMENTS.
This work has been partially supported by the Quaero program, funded by OSEO, French state agency for innovation. The Quaero French vocabulary was selected and the data were normalized by Vocapia Research.

5. REFERENCES

[1] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," *Neural Information Processing Systems*, vol. 13, pp. 933–938, 2001.

[2] H. Schwenk and J.-L. Gauvain, "Connectionist language modeling for large vocabulary continuous speech recognition," in *Proc. of ICASSP'02*, 2002, pp. 765–768.

[3] H.-S. Le et al., "Structured output layer neural network language model," in *Proc. of ICASSP'11*, 2011, pp. 5524–5527.

[4] H.-K. Kuo et al., "Morphological and syntactic features for Arabic speech recognition," in *Proc. ICASSP'10*, 2010, pp. 5190–5193.

[5] M. Kang, T. Ng, and L. Nguyen, "Mandarin word-character hybrid-input neural network language model," in *Proc. of Interspeech'11*, 2011, pp. 625–628.

[6] J. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, pp. 179–211, 1990.

[7] T. Mikolov et al., "Recurrent neural network based language model," in *Proc. of Interspeech'10*, 2010, pp. 1045–1048.

[8] T. Mikolov, A. Deoras, L. Burget, and J. Černocký, "Empirical evaluation and combination of advanced language modeling techniques," in *Proc. of Interspeech'11*, 2011, pp. 605–608.

[9] H. Ney, S. Martin, and F. Wessel, "Statistical language modeling using leaving-one-out," *Methods in Language and Speech Processing*, pp. 174–207, 1997.

[10] L. Lamel et al., "Speech recognition for machine translation in Quaero," in *Proc. of IWSLT'11*, 2011.

[11] X. Liu, M.J.F. Gales, and P.C. Woodland, "Use of contexts in language model interpolation and adaptation," in *Proc. of Interspeech'09*, 2009.