PARAPHRASTIC NEURAL NETWORK LANGUAGE MODELS

X. Liu, M. J. F. Gales & P. C. Woodland

Cambridge University Engineering Dept,
Trumpington St., Cambridge, CB2 1PZ U.K.
Email: {*xl207,mjfg,pcw*}@*eng.cam.ac.uk*

ABSTRACT

Expressive richness in natural languages presents a significant challenge for statistical language models (LM). As multiple word sequences can represent the same underlying meaning, only modelling the observed surface word sequence can lead to poor context coverage. To handle this issue, paraphrastic LMs were previously proposed to improve the generalization of back-off n-gram LMs. Paraphrastic neural network LMs (NNLM) are investigated in this paper. Using a paraphrastic multi-level feedforward NNLM modelling both word and phrase sequences, significant error rate reductions of 1.3% absolute (8% relative) and 0.9% absolute (5.5% relative) were obtained over the baseline n-gram and NNLM systems respectively on a state-of-the-art conversational telephone speech recognition system trained on 2000 hours of audio and 545 million words of texts.

Index Terms: neural network language model, paraphrase, speech recognition

1. INTRODUCTION

Natural languages are known for their expressive richness. Multiple surface realizations that are mutually paraphrastic can be used to represent the same meaning. The mapping from the underlying meaning to the observed surface realization is often one-to-many. To handle this problem, it is possible to directly model paraphrase variants when constructing the LM. Since alternative expressions of the same meaning are considered, the resulting LM's context coverage and generalization performance is expected to be improved. Along this line, the use of word level synonym features derived from WordNet-type expert resources [10, 12, 9, 5] have been investigated.

In order to model the rich paraphrastic relationship between longer span syntactic structures, such as phrases, without manually deriving the associated expert semantic labelling, a novel form of language model, the paraphrastic LM, was previously proposed in [18]. A phrase level generative model statistically learnt from large amounts of standard text data is used to explicitly generate multiple paraphrase variants for each training data sentence. Maximizing the marginal probability of these variants produces automatically smoothed n-gram statistics that are re-distributed over multiple surface realizations. This intuitively and interpretable discounting method can be exploited by many different forms of LMs that do not explicitly model the expressive richness of natural languages. In previous research, this technique were used to improve the performance of back-off n-gram LMs [18, 20, 19].

NNLMs are widely used in state-of-the-art speech recognition systems due to their inherently strong generalization performance [2, 27, 25, 13, 22, 28]. As these models do not explicitly model alternative paraphrase variants, paraphrastic modelling can be used to improve their performance. Paraphrastic feedforward NNLMs are investigated in this paper. The rest of the paper is organized as follows. Conventional NNLMs are reviewed in section 2. Paraphrastic LMs are introduced in section 3. Paraphrastic feedforward NNLMs are proposed in section 4. In section 5 paraphrastic NNLMs are evaluated on a state-of-the-art conversational telephone speech transcription task. Section 6 is the conclusion and future work.

2. NEURAL NETWORK LANGUAGE MODELS

In order to handle the data sparsity problem, language modelling techniques based on a continuous vector representation of word sequences, such as neural network LMs (NNLM), can be used. Depending on the network architecture being used, they can be categorised into feedforward NNLMs [2, 27, 25, 13], where a vector representation of fixed length history is used, and recurrent NNLMs [22, 28], which use a recurrent vector representation of longer history contexts. In this paper, feedforward NNLMs are considered.

Feedforward NNLMs represent a fixed length history context $h_i = <w_{i-1}, \ldots, w_{i-N+1}>$ of the preceding $N-1$ words in a continuous vector space. They provide a smoother and full context span probability distribution $P_{\mathsf{NN}}(\cdot|h_i)$ for all words following the current history without a back-off to lower order distributions as is required by back-off n-gram LMs. Feedforward NNLMs can be trained using a cross-entropy based error back-propagation method with a weight decay regularisation in order to reduce over-fitting. To further speed-up the training procedure, stochastic back-propagation with a bunch mode weight update can also be used [27].

To reduce computational cost, shortlist based feedforward NNLMs [27], where the output layer is limited to the most frequent words, can be used. In order to reduce the bias to in-shortlist words during training, alternative NNLM architectures that model a full vocabulary at the output layer, for example, by explicitly modelling the probability mass of out-of-shortlist (OOS) words [25, 13], can be used. An example 4-gram feedforward NNLM using an OOS node based architecture is shown in figure 1. It ensures that all training data are used in NNLM training, and the probabilities of in-shortlist words are smoothed by the OOS probability mass to obtain a more robust parameter estimation. This form of feedforward NNLMs is used in the rest of this paper. When NNLMs are linearly interpolated with back-off n-gram LMs, the probability mass of OOS words needs to be re-distributed among all OOS words [25].

3. PARAPHRASTIC LANGUAGE MODELS

Paraphrastic Language Models (PLM) [18] directly target expres-

The research leading to these results was supported by EPSRC grant EP/I031022/1 (Natural Speech Technology) and DARPA under the Broad Operational Language Translation (BOLT) program.

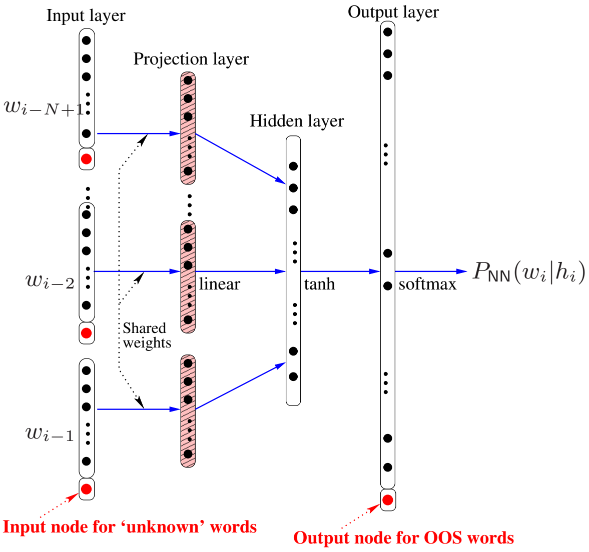

Fig. 1. *A 4-gram feedforward NNLM with an OOS node.*

sive richness related variability in natural languages. A statistically trained phrase level generative model is used to produce multiple paraphrase variants for each training data sentence. Paraphrastic LM probabilities are then estimated by maximizing the marginal probability of these paraphrase sequences. For an L word long sentence $\mathcal{W} =< w_1, w_2, ..., w_i, ..., w_L >$ in the training data, the marginal probability over all paraphrase variant sequences is maximized,

$$\mathcal{F}(\mathcal{W}) = \ln\left(\sum_{\boldsymbol{\psi},\boldsymbol{\psi}',\mathcal{W}'} P(\mathcal{W}|\boldsymbol{\psi})P(\boldsymbol{\psi}|\boldsymbol{\psi}')P(\boldsymbol{\psi}'|\mathcal{W}')P_{\text{PLM}}(\mathcal{W}')\right) \quad (1)$$

where

- $P_{\text{PLM}}(\mathcal{W}')$ is paraphrastic LM probability to be estimated.
- $P(\boldsymbol{\psi}'|\mathcal{W}')$ is a word to phrase segmentation model assigning the probability of a phrase level segmentation, $\boldsymbol{\psi}'$, given a paraphrase word sequence $\mathcal{W}'$;
- $P(\boldsymbol{\psi}|\boldsymbol{\psi}') = \prod_{v,v'} P(v|v')$ uses a phrase to phrase paraphrase model to compute probability of a phrase sequence $\boldsymbol{\psi}$ being paraphrastic to another $\boldsymbol{\psi}'$;
- $P(\mathcal{W}|\boldsymbol{\psi})$ is a phrase to word segmentation model that converts a phrase sequence $\boldsymbol{\psi}$ to a word sequence $\mathcal{W}$, and by definition is a deterministic, one-to-one mapping, thus considered non-informative.

3.1 Paraphrase model estimation: In order to generate multiple paraphrase variants $\{\mathcal{W}'\}$, the phrase level paraphrase model $\{P(v|v')\}$ in equation (1) needs to be estimated first. To obtain sufficient phrase coverage, a large number of paraphrase phrase pairs are required. As it is impractical to obtain expert semantic labelling at the phrase level, a *distributional similarity* [8] based statistical paraphrase extraction scheme that operates on standard text data [14, 26, 1, 21] is employed. The n-gram paraphrase induction algorithm proposed in [18] is used. The co-occurrence counts of two phrases of variable lengths, for example, from one word to four words maximum, sharing the same left and right three word contexts, are used to estimate the paraphrase model. Ambiguity can occur during word to phrase segmentation. If there is no clear reason to favor one phrase segmentation over another, $P(\boldsymbol{\psi}'|\mathcal{W}')$ can be treated as non-informative.

3.2 Paraphrase lattice generation: In order to train paraphrastic LMs, multiple paraphrase variants are required. Weighted finite state transducers (WFST) [24] can be can used to efficiently generate paraphrases [18]. For each training data sentence, the paraphrase word lattice $\mathcal{T}_{\mathcal{W}'}$ is generated using a sequence of WFST composition operations, before being projected onto the word sequence level and compressed via the determinization operation. This is given by

$$\mathcal{T}_{\mathcal{W}'} \;=\; \det\left(\pi_{\mathcal{W}'}\left(\mathcal{T}_{\mathcal{W}:\mathcal{W}} \circ \mathcal{T}_{\mathcal{W}:\boldsymbol{\psi}} \circ \mathcal{T}_{\boldsymbol{\psi}:\boldsymbol{\psi}'} \circ \mathcal{T}_{\boldsymbol{\psi}':\mathcal{W}'}\right)\right) \qquad (2)$$

where $\mathcal{T}_{\mathcal{W}:\mathcal{W}}$ is the transducer containing the original word sequence, $\mathcal{T}_{\mathcal{W}:\boldsymbol{\psi}}$ is the word to phrase segmentation transducer, $\mathcal{T}_{\boldsymbol{\psi}:\boldsymbol{\psi}'}$ the phrase to phrase paraphrase transducer and $\mathcal{T}_{\boldsymbol{\psi}':\mathcal{W}'}$ the phrase to word transducer. $\circ$, $\det(\cdot)$ and $\pi(\cdot)$ denote the WFST composition, determinization and projection operations.

The phrase to word transducer can be derived by taking the word to phrase transducer's inverse (swapping input and output symbols). It is also possible to construct conventional, non-paraphrastic phrase level LMs using the phrase segmentation transducer. A non-paraphrastic phrase level lattice $\mathcal{T}_{\boldsymbol{\psi}}$ containing all possible segmentations for the original sentence $\mathcal{W}$ is derived as $\mathcal{T}_{\boldsymbol{\psi}} = \det\left(\pi_{\boldsymbol{\psi}}\left(\mathcal{T}_{\mathcal{W}:\mathcal{W}} \circ \mathcal{T}_{\mathcal{W}:\boldsymbol{\psi}}\right)\right)$, before taking the shortest path associated with the longest available phrase segmentation to obtain a maximum phrase level constraint. It is possible in general that some phrases may have no suitable paraphrases available. In order to ensure the resulting paraphrase lattice is fully connected, self-reflexive arcs that map the input phrases to the same output are also included in the paraphrase transducer with zero cost.

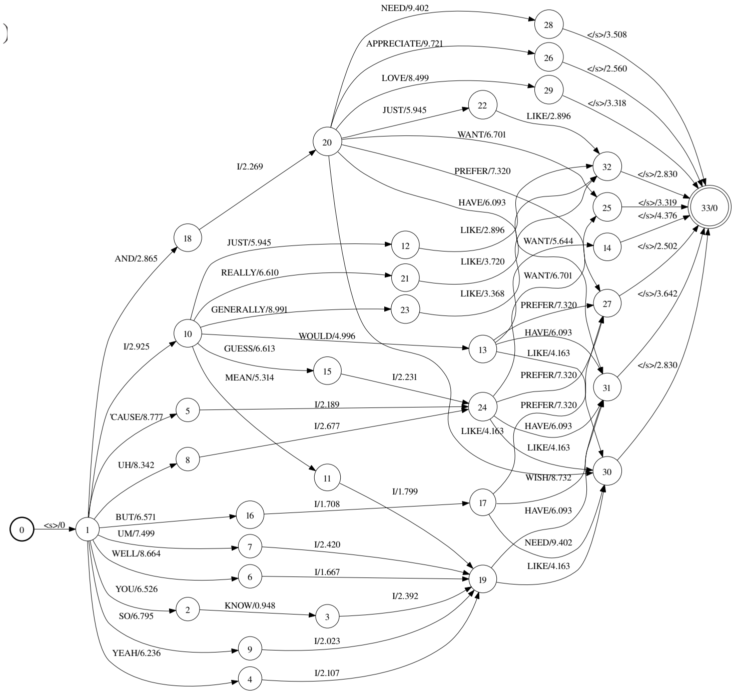

Fig. 2. A paraphrase lattice for sentence “*And I generally prefer*”.

In order to deweight the statistics accumulated from very unlikely paraphrase sequences and improve efficiency, a standard bigram LM trained on the surface word sequence can be applied to the paraphrase lattices generated using equation (2). Using this WFST based decoding approach and a paraphrase model trained on 545 million words of conversational data, for a sentence “*And I generally prefer*”, an example paraphrase lattice after pruning is shown in figure 2. Inside the lattice, the following paraphrase variants are among those generated: “*And I just like*”, “*I mean I want* ”, “*I guess I prefer* ”, “*You know I need*”, “*And I appreciate* ”, “*I would have* ”, “*‘Cause I like* ”, “*Well I need* ” and “*So I like*”. As the n-gram based paraphrase extraction method can also produce phrase pairs that are non-paraphrastic, antonym word sequences such as “*And you know I hate*” were also found in paraphrase lattice before pruning.

3.3 Paraphrastic counts smoothing: The sufficient statistics, $C(h_i, w_i)$, used to estimate the probability of a particular n-gram $P_{\mathsf{PLM}}(w_i|h_i)$ that predicts word w_i following history h_i, are now accumulated in the paraphrase lattices via a forward-backward pass, for example,

$$C(h_i, w_i) \;=\; \sum_{\mathcal{W}'} P(\mathcal{W}'|\mathcal{W}) C_{\mathcal{W}'}(h_i, w_i) \qquad (3)$$

where $C_{\mathcal{W}'}(h_i, w_i)$ is the count of subsequence $<h_i, w_i>$ occurring in paraphrase variant $\mathcal{W}'$. By discounting and re-distributing statistics to alternative paraphrases of the same word sequence, paraphrastic LMs estimated using the above statistics are expected to have a richer context coverage and improved generalization performance. This advantage can be exploited by many forms of LMs that do not explicitly capture the paraphrastic variability in natural languages. These models include, and are not restricted to, back-off n-gram LMs as investigated in previous research [18, 19, 20].

4. PARAPHRASTIC FEEDFORWARD NNLMS

Feedforward NNLMs internally cluster different fixed length history contexts via the similarity measure between their vector space representations. The underlying n-gram level vector space smoothing is different from the sequence level discounting derived from the form of paraphrastic modelling presented in section 3. During the model training process, feedforward NNLMs do not explicitly model expressive richness to improve generalization. An assumption is made that history contexts that differ significantly in their surface form or vector representations, despite being strongly related in meaning, are considered unlikely to share a similar NNLM distribution. This assumption thus limits the resulting NNLM's ability to generalize well to rich alternative expressions of the same meaning. To address this issue, the general form of paraphrastic modelling presented in section 3 can also be used to improve feedforward NNLMs' performance.

Feedforward NNLMs also share the same underlying Markov assumption with back-off n-gram LMs over previous history contexts. This advantage alleviates the need to use an explicit N-best representation of multiple paraphrase variants to obtain the paraphrastic statistics given in equation (3). A more compact lattice based paraphrase representation obtained using equation (2) can still be used. In common with the training of paraphrase back-off n-gram LMs, the first two generic stages described from section 3.1 to 3.2 to estimate the phrase level paraphrase model and paraphrase lattice generation are performed first. Then the paraphrastic statistics of the suitable n-gram order, but no other lower order counts as required in back-off n-gram LMs, are accumulated over all training data paraphrase lattices using equation (3) via a lattice forward backward pass.

After being split into single instances and randomized, these statistics can be used to train paraphrastic feedforward NNLMs using cross-entropy based error back-propagation in bunch mode, in the same fashion as the conventional feedforward NNLMs. The overall model training process is summarized below.

```
phrase level paraphrase model estimation on LM data using the
  $n$-gram paraphrase induction algorithm described in [18];
for every sentence in training data do
    generate a paraphrase lattice using WFSTs as in section 3;
    accumulate paraphrastic $n$-gram counts of equation (3) via a
      forward-backward pass in the paraphrase lattice;
end for
integerise the resulting paraphrastic $n$-gram counts, split them
  into single instances before applying randomization;
feedforward NNLM training using cross-entropy based error
  back-propagation in bunch mode until convergence.
```

In common with standard feedforward NNLMs, the paraphrastic NNLM distribution $P_{\mathsf{PNN}}(\cdot|h_i)$ is interpolated with back-off LMs. Let $P(\tilde{w}_i|h_i)$ denote the interpolated LM probability for an in-vocabulary word $\tilde{w}_i$ following some history h_i, this is given by

$$P(\tilde{w}_i|h_i) \;=\; \lambda P_{\mathsf{NG}}(\tilde{w}_i|h_i) + (1-\lambda)P_{\mathsf{PNN}}(\tilde{w}_i|h_i) \qquad (4)$$

λ is the weight assigned to the back-off n-gram LM distribution $P_{\mathsf{NG}}(\cdot)$, and kept fixed as 0.5 in all experiments of this paper.

In order to increase the context span, phrase level paraphrastic feedforward NNLMs can also be trained. This can be obtained by optimizing a simplified form of the criterion in equation (1), by dropping the word to phrase segmentation model $P(\boldsymbol{\psi}'|\mathcal{W}')$, thus the sufficient paraphrastic n-gram statistics in equation (3) accumulated on phrase level instead. In order to incorporate richer linguistic constraints, NNLMs that model different units, for example, words and phrases, can be used. These NNLMs are interpolated with the comparable back-off n-gram LMs based on their respective modelling unit, before the two interpolated LMs are finally log-linearly combined. The resulting multi-level NNLM can be used to further improve discrimination [15, 16, 17]. This requires word level lattices to be first converted to phrase level lattices before the log-linear combination is performed. The log-linear interpolation weights were set equal for word and phrase level LMs, and kept fixed for all experiments of this paper.

5. EXPERIMENTS AND RESULTS

In this section performance of various paraphrastic NNLMs are evaluated on the CU-HTK LVCSR system for conversational telephone speech (CTS) used in the 2004 DARPA EARS evaluation. The acoustic models were trained on approximately 2000 hours of Fisher conversational speech released by the LDC. A 59k recognition word list was used in decoding. The system uses a multi-pass recognition framework. In the initial lattice generation stage, adapted gender dependent cross-word triphone MPE acoustic models with HLDA projected, conversational side level normalized PLP features, and an interpolated 4-gram word level baseline LM were used. A detailed description of the baseline system can be found in [6]. The 3 hour **dev04** data, which includes 72 Fisher conversations, was used as a test set. For all results presented in this paper, matched pairs sentence-segment word error (MAPSSWE) based statistical significance test was performed at a significance level $\alpha = 0.05$.

The baseline 4-gram back-off LM was trained using a total of 545 million words from 2 text sources: the LDC Fisher acoustic transcriptions, **Fisher**, of 20 million words (weight 0.75), and the

University Washington conversational web data [4], **UWWeb**, of 525 million words (weight 0.25). These LMs are then used for lattice rescoring and word error rate (WER) performance evaluation. A total of 3.0M phrase pairs were extracted from the **Fisher** and **UWWeb** data. The expert semantic labelling by WordNet also gave 480k paraphrase phrase pairs to further improve coverage. These were used to generate paraphrases for the **Fisher** data to train various paraphrastic feedforward NNLMs.

5.1. Perplexity of Paraphrastic Feedforward NNLMs

	Paraphrastic	
LM	**ffNN**	**dev04**
w4g	-	51.8
nn$_{\rm w4g}$	×	60.6
nn$_{\rm w5g}$	×	59.2
nn$^{\star}_{\rm w4g}$	×	58.0
nn$_{\rm w4g}$	√	55.9
w4g+nn$_{\rm w4g}$	×	50.0
w4g+nn$_{\rm w5g}$	×	49.4
w4g+nn$^{\star}_{\rm w4g}$	×	50.0
w4g+nn$_{\rm w4g}$	√	49.0

Table 1. Perplexity of word level LMs on **dev04**. “w4g” denotes a 4-gram back-off LM. “w4g+nn$_{\rm w4g}$” and “w4g+nn$_{\rm w5g}$” are interpolated LMs combining “w4g” with a 4/5-gram feedforward NNLM trained on **Fisher** data. “nn$^{\star}_{\rm w4g}$” was trained also using resampled **UWWeb** data subsets of 7M words at each training epoch.

The perplexity performance of various LMs are shown in table 1. The 4-gram paraphrastic feedforward NNLM (5th line in table 1), consistently outperformed the non-paraphrastic, 4-gram and 5-gram baseline NNLMs (2nd and 3rd lines in table 1) trained on the **Fisher** data. A perplexity reduction of 2 points was also obtained over another comparable 4-gram baseline NNLM “nn$^{\star}_{\rm w4g}$” (4th line in table 1) that was trained on both the 20M word **Fisher** data and additional resampled **UWWeb** data using the method described in [27]. The size of the resampled **UWWeb** data subsets used at each training epoch was approximately 7M words, according to the ratio between the interpolation weights assigned to the **Fisher** (weight 0.75) data and **UWWeb** data (weight 0.25). The same trend was also found after these NNLMs were interpolated with the 4-gram LM (1st line in table 1), as are shown the bottom section of table 1.

5.2. WER Performance of Paraphrastic Feedforward NNLMs

The WER performance of various paraphrastic feedforward NNLMs are shown in table 2 for **dev04**. The first 6 baseline LMs are non-paraphrastic. The word level 4-gram baseline LM “w4g” gave a WER of 16.7%. The 2nd line table 2 is a multi-level baseline LM, “w4g ∘ p4g”, which incorporates phrase level constraints by log-linearly combining the surface word and phrase level 4-gram LMs. The phrase level LM was trained on the phrase level text data obtained using a longest available word to phrase segmentation as described in section 3. This multi-level baseline LM gave a WER of 16.5%. A small improvement of 0.2% absolute was obtained over the word level 4-gram baseline LM. The WER performance of the three baseline feedforward NNLMs previously shown from line 2 to 4 in table 1 are shown from the 3rd to 5th line in table 2. Using the additional web data source **UWWeb** as in LM ”w4g+nn$^{\star}_{\rm w4g}$” reduced the WER by 0.2%. The same WER of 16.1% was also obtained using a 5-gram NNLM “w4g+nn$_{\rm w5g}$”. A comparable multi-level baseline NNLM, “(w4g+nn$_{\rm w4g}$) ∘ (p4g+nn$_{\rm p4g}$)” , which log-linearly combines surface word and phrase sequence (1.2 words per phrase on average) trained non-paraphrastic feedforward 4-gram NNLMs gave a WER of 15.7%.

	Paraphrastic		
LM	**boNG**	**ffNN**	**dev04**
w4g	×	-	16.7
w4g ∘ p4g	×	-	16.5
w4g+nn$_{\rm w4g}$	×	×	16.3
w4g+nn$_{\rm w5g}$	×	×	16.1
w4g+nn$^{\star}_{\rm w4g}$	×	×	16.1
(w4g+nn$_{\rm w4g}$) ∘ (p4g+nn$_{\rm p4g}$)	×	×	15.7
w4g+nn$_{\rm w4g}$	√	×	16.1
w4g+nn$_{\rm w4g}$	×	√	16.0
w4g	√	-	16.4
w4g ∘ p4g	√	-	16.2
w4g+nn$_{\rm w4g}$	√	√	15.9
(w4g+nn$_{\rm w4g}$) ∘ (p4g+nn$_{\rm p4g}$)	√	√	**15.4**

Table 2. WER Performance of LMs on **dev04**. “w4g ∘ p4g” denotes a multi-level LM log-linearly combining word and phrase level 4-gram back-off LMs, and “(w4g+nn$_{\rm w4g}$) ∘ (p4g+nn$_{\rm p4g}$)” a multi-level LM log-linearly combining word and phrase level feedforward NNLMs, Other naming conventions same as table 1.

The WER performance of 4 fully paraphrastic LMs (both back-off LM and NNLM components) are shown from line 9 to 12 in table 2. They gave consistent WER reductions of 0.3%-0.4% over their comparable non-paraphrastic baselines from line 1 to 6 in table 2, as well as consistent improvements over the two partially paraphrastic LMs (back-off LM or NNLM component only) shown from line 7 to 8 in table 2. The best performance was obtained using the paraphrastic multi-level NNLM, “(w4g+nn$_{\rm w4g}$) ∘ (p4g+nn$_{\rm p4g}$)” , as highlighted in the last line of table 2. It log-linearly combines word and phrase level feedforward NNLMs, after a linear interpolation between 4-gram back-off LMs and NNLMs at both word and phrase level is performed. Using this paraphrastic multi-level NNLM, total error rate reductions of 1.3% absolute (8% relative) and 0.9% absolute (5.5% relative) were obtained over the baseline 4-gram word level LM “w4g” and the non-paraphrastic NNLM “w4g+nn$_{\rm w4g}$” respectively, both being statistically significant.

6. CONCLUSION AND RELATION TO PRIOR WORK

Paraphrastic feedforward NNLMs were investigated in this paper. Word error rate reductions of 1.3% (8% relative) absolute were obtained on a state-of-the-art large vocabulary speech recognition task. Consistent with the performance improvements previously obtained on back-off n-gram LMs [18, 20, 19], experimental results presented in this paper suggest the proposed method is also effective in improving the generalization performance of feedforward NNLMs. In contrast, previous research on NNLMs used no explicit paraphrastic modelling [2, 27, 25, 13, 22]. Future research will focus on improving paraphrase extraction and paraphrastic modelling for RNNLMs.

7. REFERENCES

[1] I. Androutsopoulos and P. Malakasiotis (2010). "A survey of paraphrasing and textual entailment methods", *Journal of Artificial Intelligence Research*, 38:135-187, 2010.

[2] Y. Bengio and R. Ducharme (2003). "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[3] P. F. Brown et al. (1992). "Class-based n-gram models of natural language". *Computational Linguistics* 18(4) pp.467-470.

[4] I. Bulyko, M. Ostendorf and A. Stolcke (2003). "Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures", in *Proc. HLT2003*, Edmonton, Canada.

[5] G. Cao, J-Y Nie and J. Bai (2005). "Integrating word relationships into language models", in *Proc. ACM SIGIR2005*, pp. 298-305, Salvador, Brazil.

[6] G. Evermann, H. Y. Chan, M. J. F. Gales, B. Jia, D. Mrva, P. C. Woodland and K. Yu (2005). "Training LVCSR systems on thousands of hours of data," in *Proc. ICASSP2005*, Philadelphia, PA, vol. 1, pp. 209-212.

[7] C. Fellbaum (1998) *WordNet: an electronic lexical database*, MIT Press. Cambridge, MA.

[8] Z. Harris (1954). "Distributional structure", *Word*, 10(2):3 pp.146-162.

[9] R. Hoberman and R. Rosenfeld (2002). "Using WordNet to supplement corpus statistics" (Online Document). Available: http://www.cs.cmu.edu/∼roseh/Papers/wordnet.pdf, 2002.

[10] F. Jelinek, R. Mercer and S. Roukos (1990). "Classifying words for improved statistical language models", in *Proc. IEEE ICASSP1990*, Vol. 1, pp. 621-624, Albuquerque, New Mexico.

[11] R. Kneser and H. Ney (1993), "Improved clustering techniques for class based statistical language modeling," in *Proc. EuroSpeech93'*, Berlin, Germany.

[12] R. Kneser and J. Peters (1997). "Semantic clustering for adaptive language modeling", in *Proc. ICASSP1997*, Vol. 2, pp. 779-782, Munich, Germany.

[13] H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain and F. Yvon (2013). "Structured output layer neural network language models for speech recognition", *IEEE Transactions on Audio, Speech and Language Processing*, 21(1), 2013, pp. 197-206.

[14] D. Lin and P. Pantel (2001). "DIRT - discovery of inference rules from text", in *Proc. ACM SIGKDD2001*, pp.323-328, San Francisco, CA.

[15] X. Liu, M. J. F. Gales, J. L. Hieronymus and P. C. Woodland (2010), "Language model combination and adaptation using weighted finite state transducers," in *Proc. ICASSP2010*, Dallas, TX, 2010, pp. 5390–5393.

[16] X. Liu, M. J. F. Gales and P. C. Woodland (2013). "Use of contexts in language model interpolation and adaptation," *Computer Speech and Language*, vol. 27, no. 1, pp. 301–321, January 2013.

[17] X. Liu, J. L. Hieronymus, M. J. F. Gales and P. C. Woodland (2013). "Syllable language models for Mandarin speech recognition: exploiting character sequence models", *Journal of the Acoustical Society of America*, Volume 133, Issue 1, pp. 519-528, January 2013.

[18] X. Liu, M. J. F. Gales and P. C. Woodland (2012). "Paraphrastic language models", in *Proc. ISCA Interspeech2012*, Portland, Oregon.

[19] X. Liu, M. J. F. Gales and P. C. Woodland (2013). "Paraphrastic language models and combination with neural network language models", in *Proc. IEEE ICASSP2013*, Vancouver, Canada.

[20] X. Liu, M. J. F. Gales and P. C. Woodland (2013). "Cross-domain paraphrasing for improving language modelling using out-of-domain data", in *Proc. ISCA Interspeech2013*, Lyon, France.

[21] N. Madnani and B. Dorr (2010). "Generating phrasal and sentential paraphrases: a survey of data-driven methods", *Computational Linguistics*, Vol. 36, No. 3, 2010.

[22] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky and S. Khudanpur (2010). "Recurrent neural network based language model", in *Proc. ISCA Interspeech2010*, pp. 1045-1048, Makuhari, Japan.

[23] F. Morin and Y. Bengio (2005). "Hierarchical probabilistic neural network language model", in *Proc. 10th International Workshop on Artificial Intelligence and Statistics*, Barbados, 2005, pp.246-252.

[24] M. Mohri (1997). "Finite-state transducers in language and speech processing", *Computational Linguistics*, 23:2, 1997.

[25] J. Park, X. Liu, M. J. F. Gales and P. C. Woodland (2010), "Improved neural network based language modelling and adaptation," in *Proc. ISCA Interspeech*, Makuhari, Japan, 2010, pp. 1041-1044.

[26] M. Pasca and P. Dienes (2005). "Aligning needles in a haystack: Paraphrase acquisition across the Web", In *Proc. IJCNLP2005*, Jeju Island, pp. 119-130.

[27] H. Schwenk (2007), "Continuous space language models", *Computer Speech and Language*, Vol. 21, 2007, pp. 492–518.

[28] M. Sundermeyer, R. Schlueter and H. Ney (2012). "LSTM neural networks for language modeling", in *Proc. ISCA Interspeech2012*, Portland, Oregon.