JOINT LANGUAGE MODELS FOR AUTOMATIC SPEECH RECOGNITION AND UNDERSTANDING

Ali Orkan Bayer and Giuseppe Riccardi

Signals and Interactive Systems Lab - University of Trento, Italy
{bayer, riccardi}@disi.unitn.it

ABSTRACT

Language models (LMs) are one of the main knowledge sources used by automatic speech recognition (ASR) and Spoken Language Understanding (SLU) systems. In ASR systems they are optimized to decode words from speech for a transcription task. In SLU systems they are optimized to map words into concept constructs or interpretation representations. Performance optimization is generally designed independently for ASR and SLU models in terms of word accuracy and concept accuracy respectively. However, the best word accuracy performance does not always yield the best understanding performance. In this paper we investigate how LMs originally trained to maximize word accuracy can be parametrized to account for speech understanding constraints and maximize concept accuracy. Incremental reduction in concept error rate is observed when a LM is trained on word-to-concept mappings. We show how to optimize the joint transcription and understanding task performance in the lexical-semantic relation space.

Index Terms: Spoken Language Understanding, Automatic Speech Recognition, Language Modeling, Recurrent Neural Networks

1. INTRODUCTION

Language models (LMs) in spoken language systems are used to predict the probability of a word sequence for a target language. Automatic speech recognition (ASR) systems use this information, along with acoustic models, to lower word error rate (WER). Spoken Language Understanding (SLU) systems however, map each word to its related concepts and provide, in general, a one-to-many word-concept segmentation. To optimize these systems for understanding, language models must be trained by considering the word-to-concept alignment constraints [1].

In this paper we address the training of joint optimization of LMs for ASR and SLU tasks and provide an automatic procedure for training the joint model and selecting its best parametrization. The baseline mathematical model we have selected is the Neural Networks (NNs) which naturally fits this joint modeling problem. Neural network language models have several advantages over the standard back-off n-gram language models. A NN projects word representations onto a continuous space, which yields to better smoothing of probability distributions. Therefore, NNs make better generalizations for unseen n-grams [2]. The NNs first applied to language modeling in [3], which reported improvements in perplexity. In particular in this work we use Recurrent neural networks (RNNs) which are a special instance of NNs with recurrent connections to model a short-time memory. RNNs have been used for training ASR language models in [4], where significant reductions in perplexity and WER are reported for ASR. Recently ASR-SLU joint models have been used for cache language modeling [5], however no parametrization of the joint model has been provided and no optimization provided.

This paper presents the training algorithm for joint ASR-SLU LMs that use lexical and semantic constraints and their optimization. The LMs that are used are constructed by using class-based RNNs. By performing re-scoring experiments over ASR output 100-best lists, we have shown that a spoken language system can be optimized either for a transcription or an understanding task by considering different constraints. In the rest of the paper we present the LUNA spoken language corpus we have evaluated our algorithms on, the architecture and training of the RNN and the experimental setup and results.

2. THE SPOKEN LANGUAGE CORPUS

We have used the Human-Machine (HM) part of the LUNA Italian conversational corpus [6] for the experiments. The LUNA corpus is collected by a customer care and technical support center for software and hardware. The HM part is collected with a Wizard of Oz approach. The corpus is split into training, development, and test sets, which includes 3171, 387, and 634 utterances respectively. The training set contains 30472 word and 14683 concept tokens, the development set contains 3765 word and 1818 concept tokens, and the test set contains 6436 word and 3057 concept tokens. Also the training set contains 2399 distinct words, 44 distinct concepts, and 3638 distinct word-concept pair tokens.

This work is partially funded by the FP7 PORTDIAL project n. 296170

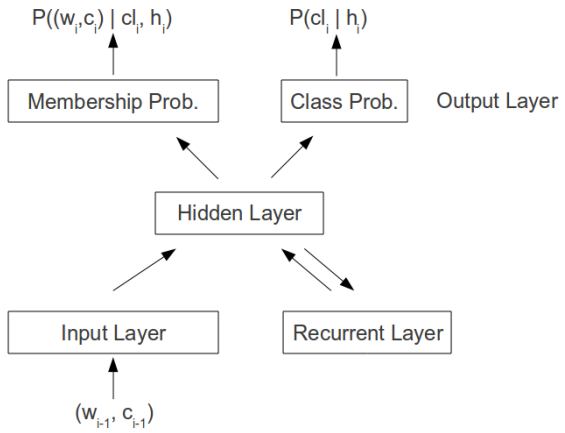

Fig. 1. *RNN structure. The input layer has as many nodes as the number of distinct word-concept (w_i, c_i) pairs. The output layer estimates probabilities for all the classes and word-concept pairs. The classes are determined manually by mapping each word-concept pair that have the same concept label to the same class. The previous word-concept pair is fed to the input layer using 1-of-n encoding. (w_i, c_i) denotes the ith word-concept pair, cl_i denotes its class h_i denotes the history for that pair.*

For the utterance: *"Buongiorno io ho un problema con la stampante da questa mattina non riesco piu a stampare"*

The corresponding semantic annotation which is derived from an ontology is: "**null**{Buongiorno io ho} **HardwareProblem.type**{un problema} **Peripheral.type**{con la stampante} **Time.relative**{da questa mattina} **HardwareOperation.negate**{non riesco} **null**{piu} **HardwareOperation.operationType**{a stampare}".

The word-concept pairs are constructed by using a one-to-one mapping of words with their annotated concepts. For example the first five pairs are: *"buongiorno - null, io - null, ho - null, un - HardwareProblem.type, problema - HardwareProblem.type"*.

3. RNN STRUCTURE

In this study, we have used RNNs to build a joint LM over words and concepts. The purpose of this LM is to predict the probability of word-concept pairs, which aims at improving the understanding performance.

The RNN structure we have used is a modified version of the class-based RNN structure given in Kombrink et. al. [7], which is available as a toolkit at http://www.fit.vutbr.cz/~imikolov/rnnlm/. The toolkit automatically assigns words to classes with respect to the frequencies of the words. We have modified the toolkit to handle manual clustering of language model units (word or word-concept pairs), i.e. we can map a language model unit to a designated class. In addition, the LMs are constructed over word-concept pairs, rather than only over words. In our joint LMs we have put the word-concept pairs that have the same concepts in the same class. Therefore, word-concept pairs which are semantically related, i.e. that have the same concept label, are mapped to the same class. The input layer has a node for each word-concept pair (w_i, c_i) available. Each word-concept pair is fed into the network using 1-of-n encoding. The LM probabilities at the output layer is factorized into class probabilities given the history and class membership probabilities as in Equation 1, where (w_i, c_i) denotes the ith word-concept pair, h_i denotes the history for the ith pair, cl_i denotes the ith class, which is the class that (w_i, c_i), the ith word-concept pair, is assigned to.

$$P((w_i, c_i)|h_i) = P(cl_i|h_i)P((w_i, c_i)|cl_i, h_i) \qquad (1)$$

The training of the RNN is done using back-propagation through time (BPTT), in which the error is propagated through recurrent connections up to a certain previous time step. As given in [7], in this way it is guaranteed that the RNN learns the history. When calculating the activations of the layers, the input layer and the recurrent layer is directly fed to the hidden layer. The activation of the hidden layer is computed using the sigmoid function, and the output probabilities are computed using the softmax function to guarantee a valid probability distribution. The structure of RNN is given in Figure 1.

4. EXPERIMENTAL SETUP

We have used an ASR system to generate 100-best lists for the LUNA corpus. This system uses acoustic models that were adapted to the corpus. It uses a conventional word based tri-gram LM with Kneser-Ney smoothing. It performs finite state transducer (FST) decoding.

For SLU we have trained a stochastic finite state transducer (SFST) based model that is described in [8]. The SLU module, λ_{SLU}, is the composition of three SFSTs. The first one, λ_W, represents the sequence of words, the second one, λ_{w2c}, maps words to concepts, and the third one, λ_{SLM}, is a concept tri-gram LM that is represented by a SFST. Therefore, our model can be described as:

$$\lambda_{SLU} = \lambda_W \circ \lambda_{w2c} \circ \lambda_{SLM}$$

The SLU model has a 29.6% CER on the transcription of the test set. The model is applied to the output of ASR to get the concepts that correspond to the hypotheses that the ASR generates. The performance of the ASR and SLU are given in Table 1.

Table 1. *Performance of ASR. ASR uses a word based tri-gram LM. Oracle error rates are given for the 100-best list.*

	WER	CER
1-best	22.3%	46.3%
Oracle	15.9%	35.2%

4.1. Baseline system

The baseline system that we will compare our joint LM with uses a word based LM. So as the baseline, we have re-scored the 100-best list that the ASR outputs by using a class-based RNN LM that was constructed only over the words. The number of classes were given as a parameter, and the words were assigned to the classes with respect to their frequencies as given in [7]. We have found out that 150 classes with 100 hidden units were performing the best for the development set. The re-scoring was also done with linear interpolation of the RNN LM and the same tri-gram LM that was used by the ASR. The results are given in Table 2.

Table 2. *Performance of the baseline system. The class-based RNN LM is constructed over words. The RNN+ngram refers to the linear interpolation of the RNN LM with the tri-gram that is used in the ASR.*

	WER	CER
RNN	21.5%	47.0%
RNN+ngram	21.5%	47.2%

As can be seen from the results; although we are able to reduce WER by performing re-scoring, there is a reduction in the understanding performance and in general SLU performance will not be predictable as WER is perturbed.

4.2. Re-scoring by using joint language models

We have optimized our system for SLU by using semantic components in the LM. For improving the understanding performance word-concept pairs are used when constructing the LM.

The construction of the LMs is performed by using the reference ontology annotations in the training data. By using these annotations word-concept pairs are extracted for each utterance. Therefore, the LMs we have constructed are joint LMs over word-concept pairs. We have trained a n-gram joint LM, and several RNNs with different sizes of hidden layers. We only report the results for the one which has the hidden layer size of 150, which gives the lowest perplexity on the reference word-concept pair annotation of the development set.

The re-scoring experiments were performed over the 100-best list that is generated by the ASR. The 100-best list is passed through the SLU module to generate the word-concept pairs for that 100-best list. RNN LM, n-gram LM and the linear interpolation of these models are used for re-scoring the output of the SLU module. The setting of the experiment can be seen in Figure 2.

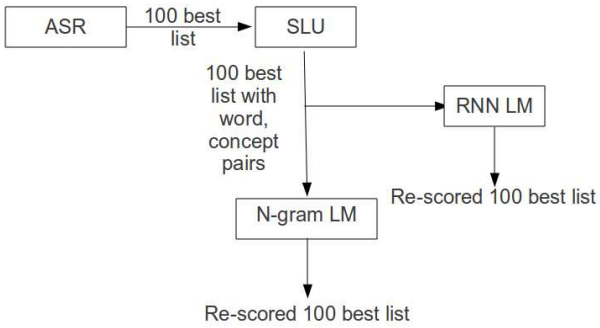

Fig. 2. *Experimental setting. The output of ASR, is fed into the SLU model to get the word-concept pairs for the 100-best list. The LM probabilities for these pairs are computed using the joint RNN and n-gram LMs.*

The RNN has 3639 nodes in the input layer, which is equal to the number of distinct word-concept pairs in the training set and a special token that denotes the end of utterance. The output layer consists of 45 classes; 44 for concepts and 1 for *null* concepts. In addition, it has 3639 nodes for each word-concept pair and the end of utterance token. The size of the hidden layer and the recurrent layer is 150. The network was trained for 14 iterations, by using the development set for setting the learning rate. The n-gram model was trained also by using the word-concept pairs. It is a tri-gram model with Kneser-Ney smoothing. The performance of 100-best re-scoring experiments with RNN, n-gram, and their linear interpolation is given in Table 3.

Table 3. *Performance of the re-scored 100-best by using joint LMs. RNN and n-gram were trained on word-concept pairs from the reference transcription. RNN+n-gram refers to the linear interpolation of the two models.*

	WER	CER
RNN	23.0%	44.1%
n-gram	26.7%	47.3%
RNN+n-gram	25.8%	46.8%

As can be seen from the results, we have obtained an improvement in CER using the joint LM. The WER, on the other hand, has increased with respect to the baseline. Alternatively, the transcription performance can be improved by reducing the concept space. Therefore, the joint LM that is based on word-concept pairs is appropriate to optimize the systems for understanding tasks. N-gram LM has suffered from data sparseness and performed worse than the baseline

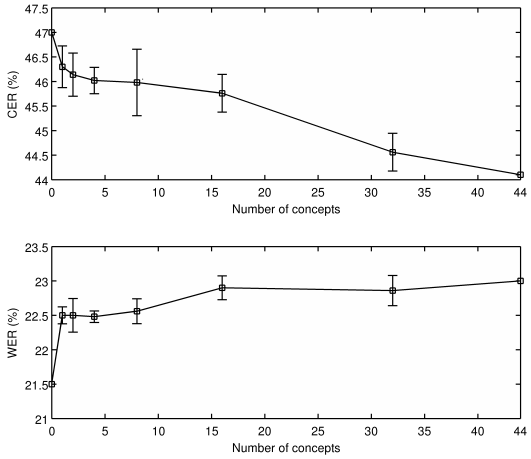

Fig. 3. *WER and CER for different number of SLU concepts. For each concept number we have sampled over the entire concept space and plot the mean and the standard deviation for 5 random concept draws.*

both for CER and WER. Better generalization ability of NNs makes them more robust to data sparseness, and makes them suitable for joint LMs.

4.3. Parameter optimization of the joint model

To see the effect of different amount of semantic information on the performance of ASR and SLU, we have trained RNN LMs for different samplings of the concepts to be included in the joint model parameters. We have selected 1, 2, 4, 8, 16, and 32 concepts from the set of concepts and map the other concepts to *null*. The concepts were grouped into 5 sets with respect to their frequencies. It was guaranteed that the concepts from all of these sets were selected randomly, while favoring the most frequent ones, i.e. when 8 concepts were selected, 2 concepts were selected from each of the most frequent 3 sets; and 1 each, from the rest. This randomization was performed for 5 times for each sampling. The mean and standard deviation of WER and CER for each of the samplings are given in Figure 3. As can be seen from the figure as the number of concepts incorporated into the LM increases there is a significant drop in the CER. On the other hand, WER increases initially as small number of concepts are included in the model and then as more concepts are added to the model WER is slightly affected.

This shows that the best word accuracy does not necessarily yields the best understanding accuracy. It has been argued in [9] that a two pass strategy for SLU may not be the best solution, where at the first stage the system is optimized for word accuracy and at the next stage SLU model is applied to the output of ASR. The authors have used a semantic LM which aimed at improving the understanding accuracy. Their model increased the WER significantly, while the overall understanding accuracy improved. Similarly in this study, we have shown that a LM that uses semantic information is well suited for understanding tasks, whereas a word based model is preferable for transcription tasks. Also, the system may be tuned by using different amount of semantic information in the LM.

4.4. Statistical significance of the results

In this study we have presented that systems can be optimized either for a transcription or an understanding task. We have observed that for the transcription task we have an improvement on WER, whereas for the understanding task we have an improvement on CER. To show the statistical significance of these results we used two different methods. We have used the bootstrap method that is given in [10] to calculate the confidence intervals. We have calculated *bootstrap-t* confidence intervals using 10^4 bootstrap replications. The p-value is calculated using the randomization method given in [11]. The results are given in Table 4. It can be seen that the improvements are statistically significant.

Table 4. *WER and CER of the two systems that are optimized for ASR and SLU. 90% confidence intervals using* 10^4 *bootstrap replications are given in brackets. Also p-values for WER and CER of the systems are presented. The results show that the improvements are significant.*

	WER	CER
Best ASR	21.5% [20.2 - 22.7]	47.0% [44.2 - 49.7]
Best SLU	23.0% [21.7 - 24.4]	44.1% [41.3 - 46.7]
p-value	1.99e-4	9.99e-5

5. CONCLUSION

In this study, we have presented LMs that are built over word-concepts pairs and aimed at increasing the understanding performance while compromising for higher WER. By performing re-scoring experiments over 100-best lists, we have obtained 6% relative improvement with respect to the baseline, which is statistically significant. We have also shown that best transcription performance does not yield the best understanding performance. Spoken language systems may be tuned either for transcription or understanding task. The lexical-semantic relations used in the LM is very important when optimizing the system for a specific task. By searching over the lexical-semantic relation space, we may control the system with respect to its performance metric (e.g. classification error). It has been also shown that both improvements in the transcription and understanding tasks are statistically significant.

6. REFERENCES

[1] Giuseppe Riccardi and Allen L. Gorin, "Stochastic language models for speech recognition and understanding.," in *Proceedings of ICSLP*. 1998, ISCA.

[2] Holger Schwenk, "Continuous space language models," *Computer Speech Language*, vol. 21, no. 3, pp. 492–518, 2007.

[3] Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2000.

[4] Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernock, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *Proceedings of Interspeech*. 2010, pp. 1045–1048, ISCA.

[5] F. Zamora-Martinez, S. Espana-Boquera, J. Castro-Bleda, M., and R. De-Mori, "Cache neural network language models based on long-distance dependencies for a spoken dialog system," in *Proceedings of ICASSP*. 2012, IEEE.

[6] Marco Dinarelli, Silvia Quarteroni, Sara Tonelli, Alessandro Moschitti, and Giuseppe Riccardi, "Annotating spoken dialogs: from speech segments to dialog acts and frame semantics," in *Proceedings of SRSL 2009 Workshop of EACL*, Athens, Greece, 2009.

[7] Tomas Mikolov, Stefan Kombrink, Lukas Burget, Jan Cernock, and Sanjeev Khudanpur, "Extensions of recurrent neural network language model.," in *Proceedings of ICASSP*. 2011, pp. 5528–5531, IEEE.

[8] Christian Raymond and Giuseppe Riccardi, "Generative and discriminative algorithms for spoken language understanding," in *Proceedings of Interspeech*. 2007, pp. 1605–1608, ISCA.

[9] Ye-Yi Wang, A. Acero, and C. Chelba, "Is word error rate a good indicator for spoken language understanding accuracy," in *Proceedings of ASRU*, 2003, pp. 577–582.

[10] M. Bisani and H. Ney, "Bootstrap estimates for confidence intervals in asr performance evaluation," in *Proceedings of ICASSP*, 2004.

[11] Alexander Yeh, "More accurate tests for the statistical significance of result differences," in *Proceedings of the 18th conference on Computational linguistics - Volume 2*, Stroudsburg, PA, USA, 2000, COLING '00, pp. 947–953, Association for Computational Linguistics.