Deep Semantic Encodings for Language Modeling

Ali Orkan Bayer and Giuseppe Riccardi

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

Abstract

Word error rate (WER) is not an appropriate metric for spoken language systems (SLS) because lower WER does not necessarily yield better understanding performance. Therefore, language models (LMs) that are used in SLS should be trained to jointly optimize transcription and understanding performance. Semantic LMs (SELMs) are based on the theory of frame semantics and incorporate features of frames and meaning bearing words (*target words*) as semantic context when training LMs. The performance of SELMs is affected by the errors on the ASR and the semantic parser output. In this paper we address the problem of coping with such noise in the training phase of the neural network-based architecture of LMs. We propose the use of deep autoencoders for the encoding of semantic context while accounting for ASR errors. We investigate the optimization of SELMs both for transcription and understanding by using deep semantic encodings. Deep semantic encodings suppress the noise introduced by the ASR module, and enable SELMs to be optimized adequately. We assess the understanding performance by measuring the errors made on target words and we achieve 3.7% relative improvement over recurrent neural network LMs.

Index Terms: Language Modeling, Semantic Language Models, Recurrent Neural Networks, Deep Autoencoders

1. Introduction

The performance of automatic speech recognition (ASR) systems is measured by word error rate (WER). However, in the literature the use of WER has been criticized because of its nature of poorly capturing the understanding performance [1, 2]. Therefore, a joint optimization over transcription and understanding must be employed by accounting the semantic constraints. The most notable LMs that consider semantic constraints are the latent semantic analysis (LSA) work in [3] and the *recognition for understanding* LM training in [1].

Deep autoencoders can be used to reduce the dimensionality of data with higher precision than principle component analysis [4]. In addition, it has been observed that deep autoencoders outperform LSA for document similarity tasks. Semantic hashing [5] is a method for document retrieval that maps documents to binary vectors such that the Hamming distance between two vectors represents the similarity between those documents. Also deep denoising autoencoders shown to learn high-level representations of the input which improves the performance of digit recognition systems [6].

Semantic LMs (SELMs) we present in this paper are neural network LMs (NNLMs) [7] that learn distributed representations for words. The architecture of SELMs are similar to the context dependent recurrent NNLMs (RNNLMs) that use recurrent connections as a short-term memory and embody a feature layer [8]. SELMs are based on the theory of frame semantics and model the linguistic scene based on either the target word or the frame features that are evoked in the utterance [9]. The linguistic scene is obtained from the ASR hypothesis and affected by the ASR noise. The noise can be reduced by pruning the erroneous frames. However, this prevents the model to capture the whole linguistic scene, and also this may not be performed well for the unseen data. In this paper, we propose to use deep autoencoders to encode frames and targets in a noisy representation for handling the ASR noise and optimizing SELMs for the whole linguistic scene. We show that SELMs can be utilized for optimizing spoken language systems both for the transcription and the understanding performance.

2. Semantic LMs

Traditional LMs model words as sequence of symbols and do not consider any linguistic information related to them [10]. Hence, they fail to capture semantic relationships between the words and the semantic context of utterances. SELMs overcome this problem by incorporating the semantic context of utterances into the LM. SELMs are based on the theory of frame semantics developed in the FrameNet project [11]. In FrameNet, word meanings are defined in the context of semantic *frames* which are evoked by linguistic forms called *target words* or *targets* [11]. The other words that complete the meaning in frames are called *frame elements*. The following shows an example of a semantic frame: “Lee **sold** a textbook *to Abby*”. In this example, the target word “**sold**” evokes the frame “COMMERCE-SELL”, and the “buyer” frame element is filled with the phrase “to Abby”. SELMs use frames and targets for semantic information. For automatic extraction of frames and targets from utterances, we have used the open-source frame-semantic parser, SEMAFOR [12]. SEMAFOR performs semantic parsing by first recognizing targets with a rule-based system, then by identifying frames by using a statistical model. At the final step, frame elements are filled by using another statistical model. SEMAFOR relies on the output of a statistical dependency parser. The reader may refer to [12] for a detailed description of SEMAFOR.

The performance of ASR can be improved by re-scoring an n-best list of hypotheses by using a more advanced LM than the one that is used for decoding. There may be various ways to select the hypotheses during re-scoring. Figure 1 shows the transcription versus the understanding performance for possible different selections of hypotheses. We measure the understanding performance by target error rate (TER), which is calculated from the errors made on target words that are the main meaning bearing elements of semantic frames. If the sole purpose of improving the performance is to optimize with respect to the transcription performance (WER), one may not improve the understanding performance (TER). Hence, LMs for re-scoring must be built to jointly optimize the transcription and the understanding performance.

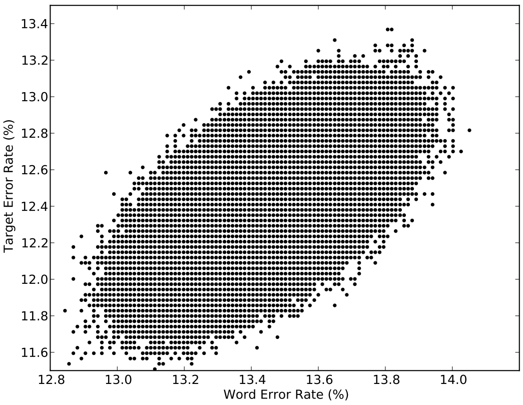

Figure 1: *Scatter plot of transcription performance (WER) versus understanding performance (TER) for random selections of hypotheses from the 100-best list of the development set of Wall Street Journal corpus.*

SELMs incorporate semantic information over the frames evoked and the targets occur in an utterance. In this respect, they are well suited for optimizing both the recognition and the understanding performance jointly. SELMs are based on the context-dependent RNNLM architecture given in [8], with the removal of the connection between the feature layer and the hidden layer. In this paper, we introduce SELMs which use deep semantic encodings of frames and targets as the semantic context. The structure of SELMs are given in Figure 2. The SELMs we have used have a class-based implementation that estimates the probability of the next word by factorizing them into class and class membership probabilities. The current word is fed into the input layer by 1-of-n encoding. The semantic layer uses the semantic encoding for the current utterance. SELMs are trained by using the backpropagation through time algorithm, which unfolds the network for N time steps back for the recurrent layer and updates the weights with the standard backpropagation [13]. SELMs also use n-gram maximum entropy features which are implemented as direct connections between n-gram histories and the output layer. The implementation applies hashing on the n-gram histories as given in [14].

3. Deep semantic encodings

A binary vector that is used in semantic hashing [5], compared to a continuous vector, introduces noise to the high-level representation of the document. For that reason, it is suitable to be used as a noisy representation of semantic information for utterances. This section describes how the training of deep autoencoders is performed for obtaining deep semantic encodings for utterances.

The training of the deep autoencoder is done in two phases as given in [5]. The phases of training is depicted in Figure 3. The input is represented with normalized bag-of-words (BoW) vectors of frames and targets in both of the phases. The first phase is the unsupervised pretraining phase for finding a good initialization of the weights. For this purpose the greedy layer-by-layer training [15] is performed. In this approach, each pair of layers are modeled by Restricted Boltzmann Machines (RBMs) and each RBM is trained from bottom to top. During the pretraining phase the bottom RBM (RBM 1) is modeled by a Constrained Poisson Model as given in [5]. Therefore, unnormalized BoW vectors are used only when computing the activations of the hidden layer, and the softmax activation function is used for the reconstruction of the input as the normalized BoW vector. The other RBMs use the sigmoid function as the activation function. The network is pretrained by using the single-step contrastive divergence [16]. In the second phase, the network is unrolled as shown in Figure 3, so that the network reconstructs the input at the output layer. The output layer uses the softmax function and reconstructs the normalized BoW input vector, the other layers use the sigmoid activation function. The backpropagation algorithm is used to fine-tune the weights by using the reconstruction error at the output layer. The codes at the “code layer” is made binary by using stochastic binary units at that layer i.e. the state of each node is set to 1 if its activation value is greater than a random value that is generated at run time; or set to 0 otherwise. This state value is used for the forward-pass. However, when backpropagating the errors the actual activation values are used. After training the autoencoder, deep semantic encodings can be obtained only by using the bottom part of the network (the part inside the dashed box in Figure 3).

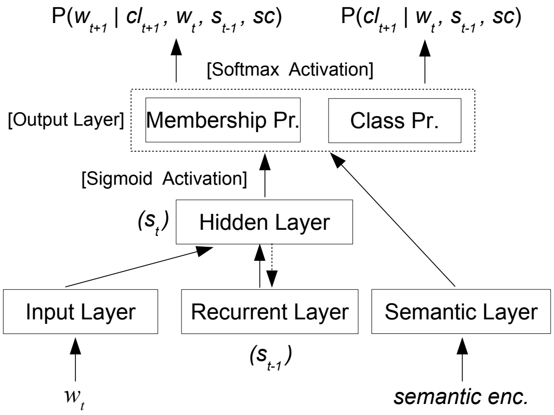

Figure 2: *The class-based SELM structure. The network takes the current word w_t and the semantic encoding for the current utterance as input. The output layer estimates the probability for the next word w_{t+1} factorized into class probabilities and class-membership probabilities (cl_{t+1} denotes the recognized class for the next word). The direct connections for the n-gram maximum entropy features are not shown.*

4. Wall Street Journal (WSJ) experiments

We present the performance of SELMs on N-best re-scoring experiments on the WSJ speech recognition task. The re-scored hypotheses are evaluated on both recognition performance (WER) and the target error rate (TER), a proxy for understanding performance. All of the experiments presented in this section are performed on the publicly available WSJ0/WSJ1 (DARPA November’92 and November’93 Benchmark) sets. All the development data under WSJ1 for speaker independent 20k vocabulary is used as the development set (“Dev 93” - 503 utterances). The evaluation is done on the November 92 CSR Speaker independent 20k NVP test set (“Test 92” - 333 utterances) and on the November 93 CSR HUB 1 test set (“Test 93” - 213 utterances).

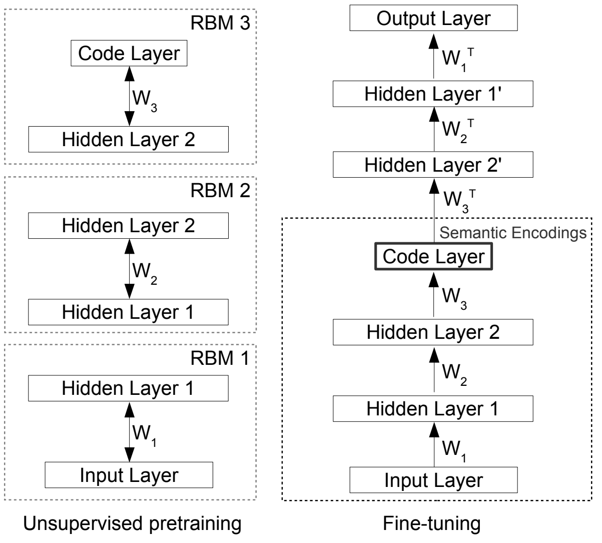

Figure 3: *The training phases of the autoencoder for deep semantic encodings. The bottom part of the fine-tuned network (dashed box) is used to obtain semantic encodings.*

4.1. ASR baseline

The baseline ASR system is trained by using the Kaldi speech recognition toolkit [17]. The vocabulary is the 20K open vocabulary word list for non-verbalized punctuation that is available in WSJ0/WSJ1 corpus. The language model that the baseline system uses is the baseline tri-gram backoff model for 20K open vocabulary for non-verbalized punctuation which is available in WSJ0/WSJ1 corpus. The acoustic models are trained on the SI-284 set, by using the Kaldi recipe with the following settings. MFCCs features are extracted and spliced in time with a context window of $[-3, +3]$. Linear discriminant analysis (LDA) and maximum likelihood linear transform (MLLT) are applied. Triphone Gaussian mixture models are trained over these features. The system performs weighted finite state decoding. We have extracted 100-best lists for both the development set and the evaluation sets. The performance of the ASR baseline is given in Table 1.

Table 1: The ASR baseline recognition performance (WER) on Dev 93, Test 92, and Test 93 sets.

	Dev 93	**Test 92**	**Test 93**
ASR 1-best	15.3%	10.2%	14.0%
Oracle on 100-best	8.3%	5.1%	7.3%

4.2. Re-scoring Experiments

The re-scoring experiments are performed on the 100-best lists that are obtained from the ASR baseline system. We have re-scored these 100-best lists by using the SELMs that are trained on frames and targets separately. In addition we have trained a RNNLM model and a 5-gram model with modified Kneser-Ney smoothing with singleton cut-offs. All models are trained on the whole WSJ 87, 88, and 89 data with the vocabulary that is limited to the 20K open vocabulary for non-verbalized punctuation. Therefore, the LMs used for re-scoring includes a 5-gram modified Kneser-Ney model with singleton cut-offs (KN5), a RNNLM model that has 200 nodes in the hidden layer and uses a maximum-entropy model that has 4-gram features with 10^9 connections (RNNME). RNNME uses 200 word classes that are constructed based on the frequencies of words, however the KN5 do not contain any classes.

Table 2: The WER performance for frame encoding models (SELM - Frame Enc.) target encoding models (SELM - Target Enc.). SELMs use ASR encodings (ASR Enc.) and reference encodings (Ref Enc.). The actual performance is given in bold.

Language Model	**Dev 93**	**Test 92**	**Test 93**
KN5	14.6%	9.7%	13.3%
RNNME	13.4%	8.8%	12.7%
(1) SELM - Frame Enc.			
ASR Enc.	**13.6%**	**8.4%**	**12.6%**
Reference Enc.	13.6%	8.4%	12.3%
(2) SELM - Target Enc.			
ASR Enc.	**13.4%**	**8.7%**	**12.0%**
Reference Enc.	13.2%	8.6%	11.9%
(1) + (2) (Lin. Interpolation)			
ASR Enc.	**13.3%**	**8.5%**	**12.0%**
Reference Enc.	13.2%	8.4%	11.8%

The SELMs use semantic encodings of frames and targets. The frames and targets for the LM training data is obtained using the SEMAFOR semantic parser. We use the most frequent frames and targets that cover the 80% of the training corpus i.e. 184 distinct frames and 1184 distinct targets. For obtaining deep semantic encodings, we have trained autoencoders of size (184-200-200-12) for frames and of size (1184-400-400-12) for targets. Pretraining is performed for 20 iterations with a mini-batch size of 100 over the frames and targets. Fine-tuning is performed by using stochastic gradient descent by considering the reconstruction error on the development set (Dev93) to avoid overfitting by adjusting the learning rate and by early stopping.

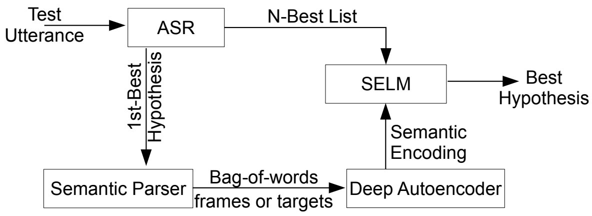

Figure 4: *The SELM re-scoring diagram. The test utterance is fed into the ASR. The 1st-best ASR hypothesis is passed through the semantic parser and BoW features are given to the autoencoder for extracting semantic encodings for the test utterance. The n-best list is re-scored by using the SELM that uses the semantic encoding as the semantic context for the test utterance.*

The SELMs are trained by using either frame encodings or target encodings that are obtained with the autoencoders. The SELMs have the same configuration with the RNNME model, i.e. they have 200 nodes in the hidden layer and use a maximum-entropy model that has 4-gram features with 10^9 connections. They also use the same word classes. All NNLMs (RNNME and SELMs) are initialized with the same random weights to make the experiments more controlled. In addition to that, the training of all NNLMs are done by using the same randomization of the training data. Since the training data is randomized we have built independent sentence models i.e. the state of the network is reset after each sentence. Dev93 is used to adjust the learning rate and for early stopping.

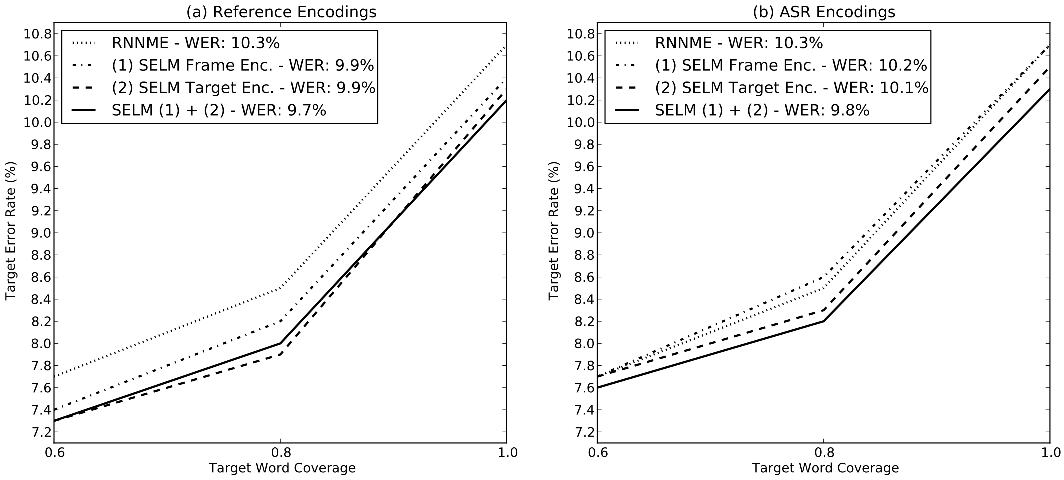

Figure 5: *TER of LMs at various coverages of target words: (a) SELMs with reference encodings, (b) SELMs with ASR encodings (actual performance). SELMs with reference encodings consistently perform better than RNNME. The target encodings suppress the ASR noise more robustly than the frame encodings. The linear interpolation of the SELMs performs the best.*

The flow of the re-scoring experiments for SELMs are shown in Figure 4. The ASR 1st-best hypothesis is passed through SEMAFOR to extract frames and targets, then deep semantic encodings are obtained by feeding them into the relevant autoencoder. Therefore, when re-scoring an utterance, semantic encodings for the whole utterance that is based on the 1st-best ASR hypothesis is used. To see how much ASR noise degrades the performance we have also performed re-scoring experiments by using the semantic encodings of the reference transcriptions. Apparently, the actual performance is given when the ASR hypothesis is used. Hence, we present two results for SELMs, 1) *ASR encodings*, refers to the actual performance, where the ASR 1st-best hypotheses are used for the semantic encodings, 2) *Reference Encodings*, where the reference transcriptions are used for the semantic encodings. In addition, we present the linear interpolation of the two SELMs on frame encodings and target encodings with equal weights. The WER performance of all the models are given in Table 2. The SELMs have a better WER performance than RNNME on the test sets. We observe that target encodings are more robust to noise than frame encodings. In addition, the linear interpolation of SELMs achieve 4.9% relative improvement in WER for the combination of “Test 92” and “Test 93” sets over RNNME.

4.3. Target Recognition Performance

WSJ corpus is designed for the speech recognition task, and it does not have any gold standards for measuring the understanding performance. Therefore, we evaluate our models on the targets recognized by the automatic semantic parser on the reference transcriptions of the development and evaluation sets. The target error rates (TER) of all models are given in Table 3. Also we analyze the error rate on the most frequent targets that cover the 60%, 80%, and 100% of the training corpus. We present results on the combination of “Test 92” and “Test 93” evaluation set in Figure 5. Both results show that if accurate semantic context (reference encodings) is used SELMs are consistently good at optimizing the performance both in terms of WER and TER. When ASR encodings are used the ASR noise affects the TER performance slightly, especially the SELMs with frame encodings. The target encodings, on the other hand, are more robust to noise. The linear interpolation of SELMs achieves 3.7% relative improvement in TER over RNNME.

Table 3: The TER performance for frame encoding models (SELM - Frame Enc.) and target encoding models (SELM - Target Enc.). SELMs use ASR encodings (ASR Enc.) and reference encodings (Ref Enc.). The actual performance of SELMs are given in bold.

Model	**Dev 93**	**Test 92**	**Test 93**
KN5	13.4%	10.4%	13.2%
RNNME	12.7%	9.6%	12.6%
(1) SELM - Frame Enc.			
ASR Enc.	**12.4%**	**9.1%**	**13.3%**
Reference Enc.	12.1%	9.1%	12.6%
(2) SELM - Target Enc.			
ASR Enc.	**12.5%**	**9.3%**	**12.5%**
Reference Enc.	12.1%	9.1%	12.3%
(1) + (2) (Lin. Interpolation)			
ASR Enc.	**12.1%**	**9.1%**	**12.3%**
Reference Enc.	11.9%	9.1%	11.9%

5. Conclusion

In this paper, we present the use of deep semantic encodings for training SELMs that exploits the semantic constraints in the language. Deep semantic encodings enable SELMs to be optimized both for the transcription and the understanding performance by suppressing the ASR noise. We observe that the target encodings are more robust to ASR noise than the frame encodings. We achieve 4.9% relative improvement in WER and 3.7% relative improvement in TER over the RNNME model for the whole evaluation set with the linear interpolation of SELMs that use frame and target encodings.

6. References

[1] G. Riccardi and A. L. Gorin, "Stochastic language models for speech recognition and understanding," in *ICSLP, Sydney, Nov. 1998*, 1998.

[2] Y.-Y. Wang, A. Acero, and C. Chelba, "Is word error rate a good indicator for spoken language understanding accuracy," in *Automatic Speech Recognition and Understanding, 2003. ASRU '03. 2003 IEEE Workshop on*, Nov 2003, pp. 577–582.

[3] J. Bellegarda, "Exploiting latent semantic information in statistical language modeling," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1279–1296, Aug 2000.

[4] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[5] R. Salakhutdinov and G. Hinton, "Semantic hashing," *International Journal of Approximate Reasoning*, vol. 50, no. 7, pp. 969–978, Jul. 2009.

[6] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, Dec. 2010.

[7] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2000.

[8] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model." in *Proceedings of SLT*. IEEE, 2012, pp. 234–239.

[9] A. O. Bayer and G. Riccardi, "Semantic language models for automatic speech recognition," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*, Dec 2014, pp. 7–12.

[10] R. Rosenfeld, "Two decades of statistical language modeling: where do we go from here?" *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1270–1278, Aug 2000.

[11] C. J. Fillmore, C. R. Johnson, and M. R. L. Petruck, "Background to Framenet," *International Journal of Lexicography*, vol. 16, no. 3, pp. 235–250, Sep. 2003.

[12] D. Das, D. Chen, A. F. T. Martins, N. Schneider, and N. Smith, "Frame-semantic parsing," *Computational Linguistics*, vol. 40, no. 1, pp. 9–56, 2014.

[13] T. Mikolov, M. Karafiat, L. Burget, J. Cernock, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of Interspeech*. ISCA, 2010, pp. 1045–1048.

[14] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. ernock, "Strategies for training large scale neural network language models," in *Proceedings of ASRU*. IEEE, 2011, pp. 196–201.

[15] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[16] G. E. Hinton, "Training products of experts by minimizing contrastive divergence." *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[17] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *Proceedings of ASRU*. IEEE, 2011.