DNN-based semantic rescoring models for speech recognition

Irina Illina, Dominique Fohr

DNN-based semantic rescoring models for speech recognition

Irina Illina and Dominique Fohr

Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy, France
{dominique.fohr,irina.illina}@loria.fr

Abstract. In this work, we address the problem of improving an automatic speech recognition (ASR) system. We want to efficiently model long-term semantic relations between words and introduce this information through a semantic model. We propose *neural network (NN) semantic models* for rescoring the N-best hypothesis list. These models use two types of representations as part of DNN input features: static word embeddings (from *word2vec*) and dynamic contextual embeddings (from *BERT*). Semantic information is computed thanks to these representations and used in the hypothesis pair comparison mode. We perform experiments on the publicly available dataset TED-LIUM. Clean speech and speech mixed with real noise are experimented, according to our industrial project context. The proposed *BERT*-based rescoring approach gives a significant improvement of the word error rate (WER) over the ASR system without rescoring semantic models under all experimented conditions and with n-gram and recurrent NN language model (Long Short-Term model, LSTM).

Keywords: Automatic speech recognition, semantics, embeddings, *BERT*.

1 INTRODUCTION

The performance of ASR is determined by the precision with which spoken words are modeled. Using acoustic and linguistic knowledge, an ASR system generates the best hypothesis corresponding to the recognized sentence. Our work is performed in the context of an industrial project. Due to the constraints of this project, we chose to study only the *N-best list rescoring approaches* to improve recognition accuracy.

State of the art ASR systems only take into account acoustic (acoustic model), lexical, and syntactic information (local n-gram language models (LM)). It may be of interest to incorporate additional knowledge into the decoding process to help ASR tackle not only clean conditions but also mismatched conditions, noisy environments, conditions specific to a particular application, etc. Some studies have attempted to include such information in an ASR. In [5], recognizer score, linguistic analysis, grammar construction, semantic discrimination score are used to rescore the N-best list. [9] indicate that articulation can provide additional information in rescoring. The use of external knowledge sources such as knowledge graph is proposed in [10]. The authors proposed to utilize the *DBpedia* knowledge graph in form of a connected graph. An N-best rescoring based on a Statistical Language Model or Dynamic Semantic Model is designed in [21].

In this article, we want to introduce semantic information in the ASR system via the N-best rescoring. Previous studies have shown that this information can be useful for ASR rescoring. The integration of semantic frames and target words in the recurrent neural network LM [1], the use of an in-domain LM and a semantic parser [2], the introduction of the semantic grammars with ambiguous context information [6] improve the accuracy of the transcriptions. Several techniques including subword units, adaptive softmax, and knowledge distillation with a large-scale model to train Transformer LMs are proposed in [8]. The authors have shown that the combination of all these techniques can significantly reduce the size of the model and improve the ASR accuracy with N-best rescoring. [14] introduce a deep duel model composed of an LSTM-based encoder followed by fully-connected linear layer and binary classifier. In [15], this approach is improved by employing ensemble encoders, which have powerful encoding capability. [18] adapt *BERT* [3, 23] to sentence scoring, and the left and right representations are mixed with a bidirectional language model.

In our work, we aim to add long-range semantic information to ASR by reevaluating the list of ASR N-best hypotheses. This research work has been carried out in the framework of an industrial project that aimed to perform the ASR in noisy conditions (fighter aircrafts). We are interested in two types of experimental conditions: clean conditions, and the context of noisy test data. These conditions are very common in real applications. We believe that some ASR errors can be corrected by taking into account distant contextual dependencies. In noisy conditions, the acoustic information is less reliable. We hope that in noisy parts of speech, the semantic model might help to remove acoustic ambiguities. The main points of the proposed rescoring approaches are: (a) rescoring the ASR N-best list using two types of continuous semantic models applied to each hypothesis: static word-based *word2vec* [13] and dynamic sentence-based *BERT*; (b) using a deep NN (DNN) framework on these semantic representations; (c) comparing hypotheses two per two; (d) combining semantic information with the ASR scores of each hypothesis (acoustic and linguistic).

Compared to [18], where only one sentence is taken at inference and masked word prediction is performed with *BERT*, we use hypothesis pairs and the sentence prediction capability of *BERT*. Compared to our previous work [11], we employ a more powerful model (*BERT)* and train a DNN network. Compared to [14], we use an efficient transformer model (*BERT*) to compare hypotheses.

2 PROPOSED METHODOLOGY

2.1 Introduction

For each of the hypothesized word *w* of the sentence to recognize, an ASR system provides an acoustic score P_{ac} *(w)* and a linguistic score P_{lm} *(w).* The best sentence hypothesis is the one that maximizes the likelihood of the word sequence:

$$\widehat{W} = argmax_{h_i \epsilon H} \prod_{w \epsilon h_i} P_{ac}(w)^{\alpha} * P_{lm}(w)^{\beta} \qquad (1)$$

$\widehat{W}$ is the recognized sentence (the end result); *w* is a hypothesized word; *H* is the set of N-best hypotheses; h_i is the *i*-th sentence hypothesis; α and β represent the weights of the acoustic and the language models.

To take into account the semantic information, one powerful solution can be to re-evaluate (*rescore*) the best hypotheses of the ASR system. In [11] we proposed to introduce the semantic probability for each hypothesis $P_{sem}(h)$ to take into account the semantic context of the sentence. This was performed through a definition of context part and possibility zones. In this rescoring approach, $P_{ac}(h)$, $P_{lm}(h)$, and the semantic score $P_{sem}(h)$ are computed separately and *combined* using specific weights α, β and γ (for $P_{sem}(h)$) for each hypothesis:

$$\widehat{W} = argmax_{h_i \epsilon H} \; P_{ac}(h_i)^{\alpha} * P_{lm}(h_i)^{\beta} * P_{sem}(h_i)^{\gamma} \quad (2)$$

In the current work, we propose a DNN-based rescoring models that rescore a pair of ASR hypotheses, one at a time. We use hypothesis pairs to get a tractable size of the DNN input vectors. Each of these pairs is represented by *acoustic, linguistic,* and *semantic* information. In our current approach, semantic information is introduced using two types of semantic representations: *word2vec* or *BERT.*

2.2 DNN-based rescoring models

The main idea behind our rescoring approach is: (a) to train DNN-based rescoring models with input features extracted from the ASR N-best list of training data; (b) to apply these models to each hypothesis pair of N-best list of a sentence to be recognized and recompute the hypothesis scores; (c) to select as the recognized sentence the hypothesis with the best recomputed score.

As mentioned before, our DNN-based rescoring models rescore *pairs of ASR hypotheses.* For each pair of hypotheses (h_i, h_j), the expected *DNN output* is: 1, if WER of h_i is lower than WER of h_j; otherwise, 0.

The global algorithm of the N-best list rescoring is as follows. From the N-best list of a sentence to recognize, for each hypothesis h_i we want to compute the cumulated score $score_{sem}(h_i)$. To perform this, for each hypothesis pair (h_i, h_j) in the N-best list of this sentence:

- We apply the DNN rescoring model and obtain the output value v_{ij} (between 0 and 1). A value v_{ij} greather than 0.5 means h_i is better than h_j.
- We update the scores of both hypotheses as:

$$score_{sem}(h_i) \mathrel{+}= v_{ij}; \qquad score_{sem}(h_j) \mathrel{+}= 1\text{-}v_{ij}. \quad (3)$$

After dealing with all the hypothesis pairs, for each hypothesis h_i, we obtain the cumulated score $score_{sem}(h_i)$ and employ it as a *pseudo* probability $P_{sem}(h_i)$, combined with the acoustic and linguistic likelihoods according to the equation (2).

2.2.1 *word2vec*-based rescoring approach

For this method, we define the contextual part and the possibility zones of the N-best list [11]. A *context part* consists of the words common to all the N-best hypotheses generated by the ASR for one sentence. We assume that this part captures the semantic information of the topic context of the sentence. We represent the contextual part with the average of the *word2vec* embedding vectors of the words of the contextual part:

$$V_{context} = \sum_{w \,\in\, context} V_{word2vec}(w) \; / \; nbrw_{context} \qquad (4)$$

where *nbrw*$_{context}$ is the number of words in the context part, and $V_{word2vec}(w)$ corresponds to a *word2vec* embedding vector *w* of the contextual part.

The *possibility zones* of a hypothesis are the set of words that do not belong to the contextual part. Possibility zones correspond to the area where we want to find the words to be corrected. We represent the possibility zones of each hypothesis by the average of the *word2vec* embedding vectors of the words of the possibility zones:

$$V_{hi} = (\sum_{\substack{w \in hi \\ w \notin context}} V_{word2Vec}(w)) \ / \ nbrw_{poss} \qquad (5)$$

where *nbrw* $_{poss}$ is the number of words in the possibility zones.

For a pair of hypotheses (h_i, h_j), the input vector for DNN network of the proposed *word2vec*-based rescoring model could contain the following features:

- context part vector $V_{context}$;
- possibility part vector V_{hi} for hypothesis h_i;
- possibility part vector V_{hj} for hypothesis h_j;
- cosine distance between $V_{context}$ and V_{hi};
- cosine distance between $V_{context}$ and V_{hj};
- acoustic score of h_i: $P_{ac}(h_i) = \prod_{w \epsilon h_i} P_{ac}(\mathrm{w})$;
- acoustic score of h_j: $P_{ac}(h_j) = \prod_{w \epsilon h_j} P_{ac}(\mathrm{w})$;
- linguistic score of h_i: $P_{lm}(h_i) = \prod_{w \epsilon h_j} P_{lm}(\mathrm{w})$;
- linguistic score of h_j: $P_{lm}(h_j) = \prod_{w \epsilon h_j} P_{lm}(\mathrm{w})$.

During training, the DNN output is set to 1 (or 0) if the first (or the second) hypothesis of the hypothesis pair achieved the lowest WER. . The *main advantage* of the proposed approach is that acoustic, linguistic, and semantic information are trained together thanks to NN-based framework. Then, according to equation (3), we obtain the cumulated score $score_{sem}(h_i)$. This cumulated score is used as $P_{sem}(h_i)$ with an appropriate weighting factor γ for combination according to equation (2). The hypothesis which obtains the greatest combined score is chosen as the recognized sentence. The proposed DNN configuration for the *word2vec*-based rescoring model is presented in the left side of figure 1, and corresponds to a neural network with 3 fully connected layers. Fully-connected layers are used to process the hypothesis pair-level representations, presented previously, and a sigmoid activation is used at the last layer to give v_{ij} in output. We call this rescoring model *word2vec*$_{sem}$.

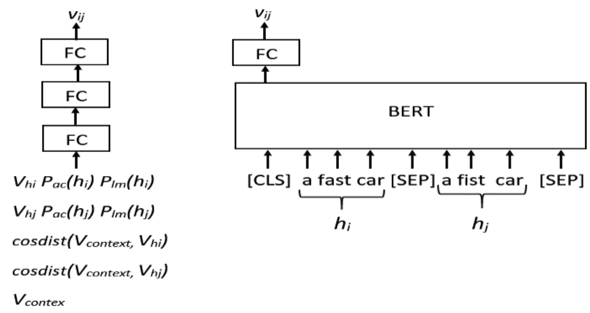
v_{ij}
FC
FC
FC
V_{hi} $P_{ac}(h_i)$ $P_{lm}(h_i)$
V_{hj} $P_{ac}(h_j)$ $P_{lm}(h_j)$
cosdist($V_{context}$, V_{hi})
cosdist($V_{context}$, V_{hj})
V_{contex}
v_{ij}
FC
BERT
[CLS] a fast car [SEP] a fist car [SEP]
h_i
h_j

Fig. 1. Architecture of the proposed DNN networks (inference stage): (*left*) *word2vec*-based rescoring DNN network; (*right*) *BERT*-based rescoring DNN network.

2.2.2 *BERT*-based rescoring approach

BERT is a multi-layer bidirectional transformer encoder that achieves state-of-the-art performance for multiples natural language tasks. The pre-trained *BERT* model can be fine-tuned using task-specific data [19].

As the cosine distance is not meaningful for *BERT* semantic model [24, 25], we cannot use it to compare the hypotheses, as we did with the *word2vec* model. So, we only compute the semantic information at the sentence level, as described below.

In our approach, we propose to take a pre-trained *BERT* model and fine-tune it using application-specific data. Two methods can be used to fine-tune the *BERT*: masked LM and next sentence prediction. We are basing our *BERT* fine-tuning on a task similar to the last one. We fine-tune *BERT* using only embeddings of CLS tokens (see Figure 1, right side). We enter a hypothesis pair (h_i, h_j), that we want to compare, to a *BERT* model. The output is set to 1 (or 0) if the first (or the second) hypothesis achieved the lowest WER. For each hypothesis h_i, we obtain the cumulated score $score_{sem}(h_i)$ (see equation (3)) and use it as a *pseudo* probability $P_{sem}(h_i)$. As for the *word2vec*-based rescoring model, this semantic probability is combined with the acoustic and linguistic likelihoods according to equation (2) with an appropriate weighting factor γ (to be optimized). In the end, the hypothesis that obtains the highest combined score is chosen as the recognized sentence. We call this rescoring model $BERT_{sem}$.

3 EXPERIMENTAL CONDITIONS

3.1 Corpus description

TED-LIUM corpus [4], containing recordings from TED conferences, is used. This corpus is publicly available. Each conference is focused on a particular subject, so the corpus is well suited to our study of exploring the semantic information. The train, development and test partitions provided within the corpus, are employed: 452 hours for training, 8 conferences for development, and 11 conferences for test (see Table 1).

This research work was carried out as *part of an industrial project*. The project concerns the recognition of speech in noisy conditions, more precisely in a *fighter aircraft*. To get closer to real aircraft conditions, we add noise to the development and test sets: noise added at 5 dB and 10 dB Signal-to-Noise Ratio (SNR) of an F16 from the NOISEX-92 corpus [20]. F-16 Fighting Falcon is a single-engine multirole fighter aircraft. The noise is *not added to the training part*. In addition to that, the proposed approaches are evaluated in clean conditions (development and testing).

Data	*Nbr. of talks*	*Nbr. of words*	*Duration*	*Nbr. of segments*
Train	2,351	4,778,000	452h	268,000
Development	8	17,783	1h36	507
Test	11	27,500	2h37	1,155

Table 1. The statistics of the TED-LIUM dataset.

3.2 Recognition system

The recognition system based on the Kaldi voice recognition toolbox [17] is employed. TDNN (Time Delay Neural Network) [22, 16] triphone acoustic models are trained on the training part (without added noise) of TED-LIUM. We perform State-level Minimum Bayes Risk training. The lexicon and LM were provided in the TED-LIUM distribution. The lexicon contains 150k words. We perform recognition using the 4-grams and RNNLM (LSTM) models [11]. We want to explore if using more powerful LM, the proposed rescoring models can improve the ASR. In all experiments, during rescoring, the LM (4-grams or RNNLM) is not modified. The 4-grams LM has 2 million grams. 4-grams and RNNLM were estimated from a textual corpus of 250 million words.

As usual, we employ the development set to choose the best parameter configuration and the test set to evaluate the proposed methods with this best configuration. We compute the WER to measure the performance. It is not possible to calculate the perplexity of our models, because the proposed models only compare two hypotheses. Therefore, in this article, we will not be providing any results related to perplexity. According to our previous work on the semantic model [11], we chose to employ an N-best list of 20 hypotheses. This size of the N-best list is reasonable to generate the pairs of hypotheses and to have a tractable computational load during the training of rescoring models.

3.3 Rescoring models

During DNN rescoring model training, the hypothesis pairs that get the same WER are not used. During evaluation (with development and test sets), all hypothesis pairs are considered. For all experiments, combination weights are: α=1, β is between 8 and 10. γ is between 80 and 100.

***word2vec*-based rescoring model.** We train the *word2vec* model on a text corpus of one billion words extracted from the *OpenWebText* corpus. The size of the generated embedding vector is 300 and the embedding models 700k words. DNN configuration for *word2vec*-based rescoring model is a neural network with 3 fully connected layers (see figure 1, left part). The dropout is 30 %, the activation function is ReLU. For the last layer, the activation function is a sigmoid. The loss function is a binary cross-entropy. We use Adam optimizer. Mini-batch size is 64 samples.

***BERT*-based rescoring model.** We download the pre-trained *BERT* models provided by Google [19]. We perform the experiments using models with 4, 8, or 12 transformer layers and the size of the hidden layers is 128, 256, or 512 neurons. In the figures, we note these models as *LxxHyyy*. For instance, *L8H256* means the *BERT* model with 8 transformer layers and 256 as the size of each hidden layer. Three epochs of fine-tuning are performed with mini-batch size of 32 samples.

4 EXPERIMENTAL RESULTS

4.1 Impact of hyperparameters

In this section, we investigate the different hyperparameters of the proposed models. As our task concerns noisy conditions, we decided to perform this study on speech in noisy conditions. The hyperparameters are studied on the development set of TED-LIUM and the best values were applied for the final evaluation on the test set. We use 4-grams LM for recognition. During rescoring the LM is not modified.

4.1.1. *word2vec*-based rescoring model

Impact of training corpus size. We utilize three different sizes of the training data: 1 million pairs of hypotheses (corresponding to 100 TED-LIUM talks of the training set), 6.6 million pairs of hypotheses (500 TED-LIUM talks of the training set), and 13.2 million pairs of hypotheses (1000 TED-LIUM talks of the training set). We observe the similar performance of the *word2vec*-based model for all data sizes. For lack of space, we do not give these results in the article and will use 500 training talks.

Impact of different DNN input features. We evaluate three configurations (*config1, config2, config3* in Figure 2): in *config1*, the DNN input contains only acoustic scores differences, linguistic scores differences and cosine differences for each hypothesis pair (3 features); in *config2*, we utilize the acoustic, linguistic scores, and cosine distances (6 features); *config3* implements all input features, presented in Section 2.2.1 (906 features). Figure 2 shows that *config1* achieves the best performance and *config3* is less efficient than *config1*. Then, embedding features provide no benefit. It is possible, that the relevant acoustic and linguistic data are diluted because the size of the embedding features (900) tends to dominate the size of the acoustic and linguistic features (4). In the following experiments, a *word2vec* rescoring model based on 500 training talks and *config1* will be used.

4.1.2 *BERT*-based rescoring model

Acoustic and LM probabilities combination (see eq. (2)) is not used in these experiments. They will be used in the overall evaluation.

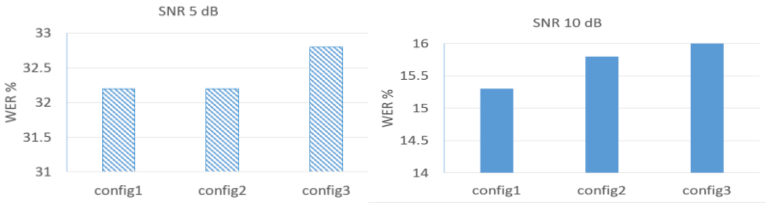

Fig. 2. ASR WER (%) on the TED-LIUM development set for different *word2vec* model configurations (different DNN input features). SNR of 5 and 10 dB. 4-grams LM, training using 100 talks.

Impact of the training corpus size. Figure 3 presents the results on the development corpus using *L8H128 BERT$_{sem}$* rescoring model with different amounts of data, i.e. pairs of N-best hypotheses, for fine-tuning. These results show that increasing the size of the fine-tuning data has a significant effect on the WER: more fine-tuning data is profitable to obtain an efficient *BERT*-based semantic model up to 1000 talks, beyond a degradation is observed.

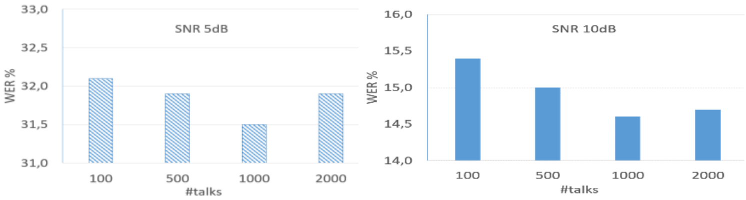

Fig. 3. ASR WER (%) on the TED-LIUM development corpus as function of the amount of *BERT* fine-tuning data. SNR of 5 dB and 10 dB, 4-grams LM, *L8H128 BERT$_{sem}$* model.

Impact of the number of hidden layers. Figure 4 shows the recognition performances as a function of the number of layers of the *BERT$_{sem}$* model. The size of the hidden layers is 128 and the size of the fine-tuning data is 1000 talks. Using 12 layers gives the best performance for the two SNR levels. We observe that this parameter plays an important role.

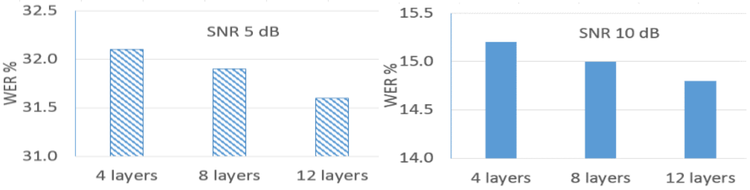

Fig. 4. ASR WER (%) on the TED-LIUM development corpus according to the number of layers for the *BERT* model. SNR of 5 dB and 10 dB, 4-grams LM, *LxH128 BERT$_{sem}$* model fine-tuned using 1000 training talks.

Impact of the hidden layers size. Figure 5 reports the importance of the hidden layers size. We use the L12Hyyy *BERT* model fine-tuned on 1000 training talks. We may observe a variation according to the size of the hidden layers. The best performance is obtained for a size of 256.

In conclusion, we can say that for the *BERT*-based rescoring model, it is important to utilize a large enough corpus for fine-tuning the model and to choose a model with many transformer layers. The size of 256 for hidden layers, 12 layers and 1000 talks for fine-tuning seems to be a good compromise. These values will be used in the following.

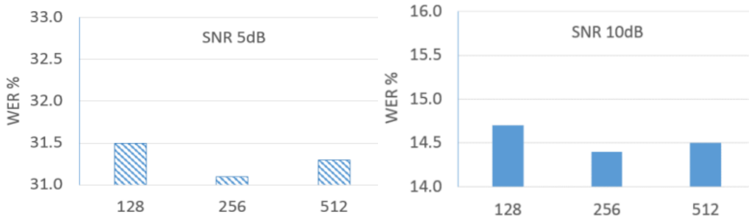

Fig. 5. ASR WER (%) on the TED-LIUM development corpus as function of the size of hidden layer of *BERT*. SNR of 5 and 10 dB, 4-grams LM, *L12Hyyy BERT$_{sem}$* fine-tuned on 1000 talks.

4.2 Global results

To further analyze the impact of proposed rescoring models, Table 2 and 3 report the WER for the development and the test sets of TED-LIUM with noise conditions of 10 and 5 dB and with clean speech. In the tables, method *Random* corresponds to the random selection of the recognition result from the N-best list, without the proposed rescoring models. Method *Baseline* corresponds to not using the rescoring models (*standard ASR*). Method *Oracle* represents the maximum performance that can be obtained by searching in the N-best hypotheses: we select the hypothesis which minimizes the WER for each sentence. The other lines of the table display the performance of the proposed approaches. For all experiments, the N-best list of 20 is used.

For the proposed rescoring models, we study 3 configurations:

(1) Rescoring using only the scores *score$_{sem}$(h)* computed with rescoring models as presented in section 2.2 (denoted *X$_{sem}$* in Tables). In this case, in equation (2) α=0, β=0, and γ=1.

(2) Rescoring using a combination of the score *score$_{sem}$(h)*, and the acoustic score *P$_{ac}$ (h)* (denoted *X$_{sem}$ comb. with ac. scores* in Tables). In this case, *score$_{sem}$(h)* is used as a *pseudo probability* and multiplied to the acoustic likelihood with a proper weighting factor γ (to optimize). In this case, *P$_{lm}$ (h)* is not used, namely, in equation (2) β=0.

(3) Rescoring using a combination of the score *score$_{sem}$(h)*, the acoustic *P$_{ac}$ (h)*, and the linguistic score *P$_{lm}$ (h)* (*X$_{sem}$ comb. with ac./ling. scores* in Tables).

We present the results only for the best *BERT*-based rescoring model L12H256 fine-tuned using 1000 training talks.

From Table 2, we can observe that *word2vec$_{sem}$* rescoring model gives a small but *significant* improvement compared to the baseline system (confidence interval is computed according to the matched-pairs test [7], used for deciding whether the difference in error-rates between two algorithms tested on the same data set is statistically significant). Unsurprisingly, the proposed *BERT*-based rescoring model outperforms the *word2vec*-based model. It is important to note that in the *word2vec*, the word embed-

Methods/systems	***SNR 5 dB***		***SNR 10 dB***		***no added noise***	
	Dev.	***Test***	***Dev.***	***Test***	***Dev.***	***Test***
Random system	33.5	41.3	16.9	22.9	10.6	12.1
Baseline system	32.7	40.3	15.7	21.1	8.7	8.9
word2vec$_{sem}$	32.1	39.2	15.3	20.6	8.5	8.8
word2vec$_{sem}$ *comb with ac. scores*	31.8	39.2	15.2	20.5	8.5	8.8
word2vec$_{sem}$*comb.with.ac./4grams.sc.*	31.5	38.8	15.2	20.4	8.5	8.8
BERT$_{sem}$	31.1	38.7	14.4	19.8	8.0	8.7
BERT$_{sem}$ *comb with ac. scores*	**30.6**	**37.9**	14.2	**19.4**	7.9	8.6
BERT$_{sem}$*.comb with ac./4grams sc.*	**30.6**	**37.9**	**14.1**	**19.4**	**7.8**	**8.5**
Oracle	27.5	33.2	11.2	15.0	5.2	4.7

Table 2. ASR WER (%) on the TED-LIUM development and test sets, SNR of 10 and 5 dB, and no added noise. N-best hypotheses list of 20 hypotheses, **4-grams LM**. L12H256 *BERT* model fine-tuned on 1000 training talks.

dings are static and a word with multiple meanings is conflated into a single representation. In the *BERT* model, the word embeddings are dynamic and more powerful, because one word can have several embeddings in the function of the context words.

Adding the acoustic score to the rescoring models (*X*$_{sem}$ *comb. with ac. scores* in Tables) improves the performance. Indeed, the acoustic score is an important feature and should be taken into account. On the other hand, adding the linguistic score during rescoring gives no improvement compared to the *X*$_{sem}$ model. We do not present this result in the tables. Using the linguistic and acoustic scores in the *BERT* rescoring model (*BERT*$_{sem}$ *comb. with ac./4-grams scores*) brings only small improvement compared to *BERT*$_{sem}$ *comb. with ac. score*: Google's *BERT* model, trained on billions of sentences, probably captures the linguistic structure of the language better than an n-gram LM trained on a much smaller corpus.

For *BERT*-based rescoring results, all improvements are *significant* compared to the baseline system. On the test set, *BERT*$_{sem}$ *comb. with ac./4-grams scores* obtains an absolute improvement of 2.4 % for 5 dB (37.9 % versus 40.3 %), 1.7 % for 10 dB (19.4 % WER versus 21.1 % WER), and 0.4 % for clean speech (8.5 % versus 8.9 %) compared to the baseline system. This corresponds to about of 6 % (for 5 dB), 8 % (for 10 dB), and 4 % (for clean speech) of relative WER improvement.

To better model long-range dependencies of LM, we perform the ASR experiments using a more powerful RNNLM (LSTM). In this case, the RNNLM is used. RNNLM is applied on the ASR word lattices and employed to generate the N-best list. Table 3 reports the results for the same set of experiments but using RNNLM. We can observe that the proposed rescoring methods give consistent and *significant* improvements, except for clean speech. In clean conditions, only *BERT*$_{sem}$ *comb. with ac./RNNLM scores* give an improvement compared to the baseline system. Finally, the best system (*BERT*$_{sem}$ *comb. with ac./RNNLM scores*) on the test set gives relative improvement of about 4.6 % for 5 dB (35.4 % versus 37.1 %), 4.5 % for 10 dB (16.9 % versus 17.7 %), and 8.3 % for clean conditions (6.6 % versus 7.2 %) compared to the baseline system. These improvements are *significant.* We observe also, that in the case of RNNLM, for some cases, the improvements are smaller compared to the 4-grams case. It is possible that RNNLM may reduce the effect of semantic rescoring because RNNLM takes better into account the long-range context dependences.

Methods/systems	***SNR 5 dB***		***SNR 10 dB***		***no added noise***	
	Dev	***Test***	***Dev***	***Test***	***Dev***	***Test***
Random system	29.2	38.4	13.9	20.2	8.9	10.8
Baseline system	28.2	37.1	12.3	17.7	6.6	7.2
word2vec$_{sem}$	27.4	36.3	12.0	17.5	6.6	7.2
word2vec$_{sem}$ *comb with ac. scores*	27.3	35.6	12.1	17.5	6.8	7.2
word2vec$_{sem}$ *comb with ac./RNNLM sc.*	27.3	35.5	12.0	17.4	6.6	7.2
BERT$_{sem}$	27.0	35.9	12.0	17.4	7.1	8.1
BERT$_{sem}$ *comb with ac. scores*	26.6	**35.3**	11.6	17.1	6.9	7.1
BERT$_{sem}$ *comb with ac./RNNLM sc.*	**26.5**	35.4	**11.5**	**16.9**	**6.0**	**6.6**
Oracle	23.1	30.2	8.3	12.1	3.8	3.5

Table 3. ASR WER (%). N-best hypotheses list of 20 hypotheses. TED-LIUM development and test sets, SNR of 10 and 5 dB, and no added noise. **RNNLM (LSTM).** L12H256 *BERT*$_{sem}$ model fine-tuned on 1000 talks.

5 CONCLUSION

The goal of this article is to improve the ASR using a rescoring of ASR N-best hypotheses. The main idea of the two proposed approaches is to model the semantic characteristics of words and their contexts. Two approaches are proposed: *word2vec*-based and *BERT*-based rescoring models. The information, extracted thanks to these representations, is learned using DNN-based training. Acoustic and linguistic information is integrated too. To evaluate our methodology, the corpus of TED-LIUM conferences is used. The best rescoring system *BERT*, combined with acoustic and linguistic scores, brings between 4 % and 8 % of relative WER improvement compared to the baseline system, using 4-grams or RNNLMs, and evaluated in clean and noisy conditions. These improvements are statistically significant.

6 Acknowledgments

The authors thank the DGA (*Direction Générale de l'Armement, part of the French Ministry of Defence*), Thales AVS and Dassault Aviation who are supporting the funding of this study and the "Man-Machine Teaming" scientific program.

7 REFERENCES

1. Bayer A., Riccardi G.: Semantic Language Models for Automatic Speech Recognition. In: Proceedings of the IEEE Spoken Language Technology Workshop (SLT) (2014)
2. Corona R., Thomason J., Mooney R.: Improving Black-box Speech Recognition using Semantic Parsing. In: Proceedings of the The 8th International Joint Conference on Natural Language Processing, pp.122–127 (2017)
3. Devlin J., Chang M.-W., Toutanova K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)

4. Fernandez H., Nguyen H., Ghannay S.,Tomashenko N., Esteve Y.:TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation. In: Proceedings of SPECOM, pp. 18–22 (2018)
5. Fuchun P., Roy S., Shahshahani B., Beaufays F.: Search results based N-best hypothesis rescoring with maximum entropy classification. In: ASRU, pp. 422-427 (2013)
6. Gaspers J., Cimiano P.: Semantic parsing of speech using grammars learned with weak supervision. In: Proceedings of the HLT-NAACL, pp. 872-881 (2015)
7. Gillick L., Cox S.: Some Statistical Issues in the Comparison of Speech Recognition Algorithms. In: Proceedings of ICASSP, v. 1, pp. 532-535 (1989)
8. Huang H., Peng F.: An Empirical Study of Efficient ASR rescoring with Transformers. In: arXiv:1910.11450v1 (2019)
9. Jinyu L., Tsao Y., Lee C.-H.: A Study on Knowledge Source Integration for Candidate Rescoring in Automatic Speech Recognition. In: In ICASSP (1), pp. 837-840 (2005)
10. Kumar A., Morales C., Vidal M.-E., Schmidt C., Auer S.: Use of Knowledge Graph in Rescoring the N-best List in Automatic Speech Recognition. In: *arXiv:1705.08018v1* (2017)
11. Level S., Illina I., Fohr D.: Introduction of semantic model to help speech recognition. In: International Conference on Text, Speech and Dialogue (2020)
12. Mikolov T., Kombrink S., Burget L., Cernocky J.-H., Khudanpur S.: Extensions of recurrent neural network language model. In: Proceedings of the ICASSP, pp. 5528–5531 (2011)
13. Mikolov T., Sutskever I., Chen K., Corrado G., Dean J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, 26, pp. 3111-3119 (2013)
14. Ogawa A., Delcroix M., Karita S., Nakatani T.: Rescoring N-Best Speech Recognition List Based on One-on-One Hypothesis Comparison Using Encoder-Classifier Model. In: IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP (2018)
15. Ogawa A., Delcroix M., Karita S., Nakatani T.: Improved Deep Duel Model for Rescoring N-best Speech Recognition List Using Backward LSTMLM and Ensemble Encoders. In: Proceedings of Interspeech (2019)
16. Peddinti V, Povey D., Khudanpur S.: A time delay neural network architecture for efficient modeling of long temporal contexts. In: Interspeech (2015)
17. Povey D., Ghoshal A., Boulianne G., Burget L., Glembek O., Goel N., Hannemann M., Motlicek P., Qian Y., Schwarz P., Silovsky J., Stemmer G., Vesely K.: The Kaldi Speech Recognition Toolkit. In: Proceedings of IEEEWorkshop on Automatic Speech Recognition and Understanding (ASRU) (2011)
18. Shin J., Lee Y., Yung K.: Effective Sentence Scoring Method Using BERT for Speech Recognition. In: Proceedings of ACML (2019)
19. Turc I., Chang M.-W., Lee K.,Toutanova K.: Well-Read Students Learn Better: On the Importance of Pre-training Compact Models. In: arXiv:1908.08962v2 (2019)
20. Varga A., Steeneken H.: Assessment for automatic speech recognition II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. In: Speech Communication, Volume 12, Issue 3, pp. 247-251 (1993)
21. Verhasselt J., Dercks H.: N-best list rescoring in speech recognition. In: The Journal of the of the Acoustical Society of America 128(6) (2010)
22. Waibel A., Hanazawa T., Hinton G., Shikano K., Lang K. J.: Phoneme Recognition Using Time-Delay Neural Networks. In: IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume 37, No. 3, pp. 328. - 339 March (1989)
23. Wang A., Cho K.: BERT has a mouth, and it must speak: BERT as a Markov random field language model. In: Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation, pages 30–36 (2019)

24. https://github.com/hanxiao/bert-as-service/

25. https://github.com/hanxiao/bert-as-service#q-thecosine-similarity-of-two-sentence-vectors-is-unreasonably-high-eg-always--08-whats-wrong