BERT-based Semantic Model for Rescoring N-best Speech Recognition List

Dominique Fohr, Irina Illina

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

Abstract

The word error rate (WER) of an automatic speech recognition (ASR) system increases when a mismatch occurs between the training and the testing conditions due to the noise, etc. In this case, the acoustic information can be less reliable. This work aims to improve ASR by modeling long-term semantic relations to compensate for distorted acoustic features. We propose to perform this through rescoring of the ASR N-best hypotheses list. To achieve this, we train a deep neural network (DNN) combining semantic, acoustic and linguistic information. Our DNN rescoring model is aimed at selecting hypotheses that have better semantic consistency and therefore lower WER. We investigate a powerful representation as part of input features to our DNN model: dynamic contextual embeddings from Transformer-based BERT. Acoustic and linguistic features are also included. We perform experiments on the publicly available dataset TED-LIUM. We evaluate in clean and in noisy conditions, with n-gram and Recurrent Neural Network Language Model (RNNLM), more precisely Long Short-Term Memory (LSTM) model. The proposed rescoring approaches give significant WER improvements over the ASR system without rescoring models. Furthermore, the combination of rescoring methods based on BERT and GPT-2 achieves the best results

Index Terms: automatic speech recognition, semantic context, embeddings, BERT

1. Introduction

ASR systems have made significant progress in recent years. Classical ASR systems take into account only acoustic (acoustic model), lexical, and syntactic information (local n-gram language models (LM)). In the case of mismatch conditions between training and testing, like noise, the signal is distorted and the acoustic model may not be able to compensate for this variability. Even if the noise compensation methods work well [9], it may be interesting to incorporate *semantic knowledge* in the decoding process to help the ASR to better take into account the long term semantic context to combat adverse conditions. It could also be useful in matched conditions. Indeed, semantic information is important for ASR systems. Some studies have tried to include this information into an ASR. The authors of [14] use a semantic context for recovering proper names missed in the ASR process. [1] integrates semantic frames and target words into recurrent neural network LM. In 0, the re-ranking of the ASR hypotheses using an in-domain LM and a semantic parser significantly improve the accuracy of transcription and semantic understanding. [5] introduces semantic grammars applicable for ASR and understanding using ambiguous context information.

Rescoring the ASR N-best hypotheses list can be an efficient solution to incorporate long-range semantic information. [16] formalizes the N-best list rescoring as a learning problem and use a wide range of features with automatically optimized weights to re-rank the N-best lists. [12] introduces N-best rescoring through LSTM-based encoder network followed by a fully-connected feedforward NN-based binary-class classifier. [15] propose a bi-directional LM for rescoring, and utilize the word prediction capability of the *BERT* model [3][19] ~~using masked word tokens~~.

In this work, we aim to *add long-range semantic information to ASR through the rescoring the ASR N-best hypotheses list*. We believe that some ASR errors can be corrected by taking into account distant contextual dependencies. This is especially important for noisy conditions. We hope that in noisy parts of speech, the semantic model could help to remove the acoustic ambiguities. The core ideas of the proposed rescoring approaches are as follows. First, we use continuous?? semantic model to represent each hypothesis: dynamic sentence-based?? *BERT*. Compared to [15], where masked word prediction is performed for *BERT*, we use the *sentence prediction capability* of the *BERT* model. Second, we compare ASR hypotheses two per two and propose two BERT-based models. Finally, we propose efficient DNN architecture to train together semantic, acoustic and linguistic information. The obtained score is combined with the ASR scores attached to each hypothesis (acoustic and linguistic) and used to re-score the ASR N-best list hypotheses. Compared to [16], we use different DNN features. Compared to our previous work [8], we use the powerful *BERT*-based semantic model, represent the hypotheses at the sentence level, and train hypotheses representations by a DNN. In experiments using a publicly available speech corpus, we systematically explore the effectiveness of the proposed features and their combinations. The proposed approaches steadily outperform the baseline ASR system in clean and all noisy conditions

2. Proposed methodology

2.1. Introduction

A classical speech recognition system provides an acoustic score *P_{ac} (w)* and a linguistic score *P_{lm} (w)* for each of the hypothesized word *w* of the sentence to recognize. 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); *H* is the set of *N*-best hypotheses; h_i is the *i*-th sentence hypothesis; *w* is a hypothesized word. α and β represent the weights of the acoustic and the language models.

An efficient way to take into account semantic information is to re-evaluate (rescore) the best hypotheses of the ASR system. We propose to introduce for each hypothesis h_i the semantic probability $P_{sem}(h_i)$ to take into account the semantic context of the sentence. In our rescoring approach, $P_{ac}(h_i)$, $P_{lm}(h_i)$, and the semantic score $P_{sem}(h_i)$ are computed and combined using specific weights α, β and γ (for $P_{sem}(h_i)$) for each hypothesis:

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

We propose to rescore using a pair of ASR hypotheses, one at a time. Each hypothesis of each pair is represented by *acoustic, linguistic,* and *semantic* information. Semantic information is produced by our proposed DNN-based rescoring models. ~~These informations are combined like in formula (2) using specific weights at the hypotheses pair level. We use hypotheses pairs to get a tractable size of the rescoring DNN input vectors.~~ In our approach, semantic information is introduced using BERT representation. Different semantic properties and efficiencies of this representation motivate us to explore it for our task of ASR N-best rescoring. We also explored the *word2vec* model [11] but given its poorer performance we do not give its results in this article.

Furthermore, we propose to go beyond a simple score combination, like in (2). We propose two DNN-based rescoring models producing $P_{sem}(h_i)$: (a) the first model is purely semantic and uses as input only textual information. We call this model $BERT_{sem}$. (b) the second model takes as input *acoustic, linguistic,* and *textual information*. We call this model $BERT_{als}$. We believe that the acoustic and linguistic information should be trained together with semantic information to give accurate rescoring model.

2.2. DNN-based rescoring models

We decided to put at the input of the DNN the features computed from a pair of hypotheses. For each pair of hypotheses (h_i, h_j), the expected DNN *output* is: (a) *1*, if WER of h_i is lower than WER of h_j ; (b) *0*, otherwise. The overall algorithm of the N-best list rescoring is as follows. For a given sentence, for each hypothesis h_i we want to compute the cumulated score *score*(h_i). To perform this, for each hypotheses pair (h_i, h_j) of the N-best list of this sentence:

- we apply the DNN model and obtain the output value v (between 0 and 1). A value v close to 1 means that the h_i is better than h_j. We use this value to compute the scores for these hypotheses.
- we update the scores of both hypotheses as:
  score(h_i) += v; *score*(h_j) += *1-v*.

The obtained cumulated score *score*(h_i) is used as a *pseudo* probability $P_{sem}(h_i)$ and combined to the acoustic and linguistic likelihoods with a proper weighting factor (to be optimized) according to eq. (2). At the end, the hypothesis that obtains the greatest score is chosen as the recognized sentence.

Our two proposed DNN-based rescoring models producing $P_{sem}(h_i)$ are based on *BERT*. It is a multi-layer bidirectional transformer encoder that achieves state-of-the-art performances for multiples natural language tasks (NLP). The pre-trained *BERT* model can be fine-tuned using task-specific data [17]. As the cosine distance is not meaningful for *BERT* semantic model [20][21], we compute the semantic information at the sentence level, as described below.

In our approach, we use a pre-trained *BERT* model. Two methods can be used to fine-tune *BERT* using application-specific data: *masked LM* and *next sentence prediction*. We based our *BERT* fine-tuning on a task similar to the latter one.

The first proposed $BERT_{sem}$ model consists in performing the fine-tuning of BERT. We input a hypotheses pair (h_i, h_j), that we want to compare and the output is set to 1 (or 0) if the first (or the second) hypothesis achieved the lowest WER.

The second proposed model $BERT_{al\_sem}$ takes in input a feature vector including *acoustic, linguistic,* and *semantic information*. Figure 1 shows the architecture of this model: the hypotheses pair is given to the BERT-transformer model. After, the embedding token of BERT, representing this pair, is given to Bi-LSTM and a fully connected layer (FC). Finally the output of this FC is concatenated with the acoustic and linguistic information of hypotheses pair and passed through the last FC layer.

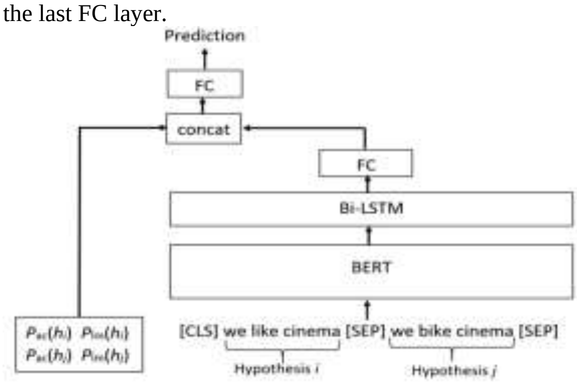
Prediction
FC
concat
FC
Bi-LSTM
BERT
$P_{ac}(h_i)$ $P_{lm}(h_i)$
$P_{ac}(h_j)$ $P_{lm}(h_j)$
[CLS] we like cinema [SEP] we bike cinema [SEP]
Hypothesis *i*
Hypothesis *j*

Figure 1: *Proposed $BERT_{als}$ rescoring model.*

3. Experimental conditions

3.1. Corpus description

We used publicly available TED-LIUM corpus [4], containing the recordings of the TED conferences. Each conference is focused on a particular subject and so the corpus is well suited to our study. We used the train, development and test partitions provided with the TED-LIUM corpus: 452 hours for training (268k segments, 4778k words, 452 hours), 8 conferences (507 segments, 17783 words, 1 hour 36 minutes) for development, and 11 conferences (1155 segments, 27500 words, 2 hours 37 minutes) for test set (see Table 1). As usual, we used the development set to choose the best parameter configuration and the test set to evaluate the proposed methods with this best configuration. We used the WER to measure the performance. As our model only compares two hypotheses and does not has the ability to estimate the word probabilities, it is not possible to calculate the perplexity of our model. Therefore, in this article, we will not give any results in terms of perplexity.

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

Data	*Nbr. of talks*	*Nbr. of words*	*Duration*
Train	2351	4.8M	452h
Development	8	17783	1h36
Test	11	27500	2h37

This research work was carried out as part of an industrial project, concerning the recognition of speech in noisy conditions, more precisely in a fighter aircraft. So, we added noise to the development and test sets to get closer to real aircraft conditions: additive noise at 10 dB and 5dB SNR (noise of a F16 from the NOISEX-92 corpus [18]). The noise is *not added* to the training part. Furthermore, we evaluated the proposed approaches in clean conditions (training and testing).

3.2. Recognition system description

We used a recognition system based on the Kaldi voice recognition toolbox [13]. TDNN triphone acoustic models are trained on the training part (without noise) of TED-LIUM using sMBR training (State-level Minimum Bayes Risk). The lexicon and LM were provided in the TED-LIUM distribution. The lexicon contains 150k words. The LM has 2 million 4-grams and was estimated from a textual corpus of 250 million words. We also performed N-best list generation using the RNNLM model (LSTM) [10]. We want to see if using more powerful LM, the proposed rescoring models can improve the ASR. In all experiments, during N-best rescoring, the LM (4-gram or RNNLM) is not modified.

The WER of our ASR system on TED-LIUM publicly provided test set is around 8 % using n-gram LM (using the training and the test sets without added noise).

As *Generative Pre-Training Transformer 2* model (GPT-2) showed a good performance in several NLP tasks [Radfort2019??], we used this model in our experiments. The pre-trained GPT-2 language model was downloaded from *Hugging Face* site. This model contains 117M parameters and was trained by OpenIA to predict the next word in 40GB of Internet text. This model is used as language model score during the N-best rescoring.

According to our previous work on the semantic model [8], the use of 5 or 10 hypotheses of the N-best list is not enough for the efficient rescoring. Using more than 25 hypotheses shows no further improvement. In the current work, we chose to use N-best lists of 20 hypotheses in all our experiments. Moreover, this size of the N-best lists seems to be reasonable to generate the pairs of hypotheses and to have a tractable computational load during the training of rescoring models. In the case of large size of N-best list, another strategy of pair comparison can be considered [12]. During the training, the hypotheses pairs that get the same WER are not used. During evaluation (with development and test sets), all hypotheses pairs are considered.

4. Experimental results

4.1. Impact of hyperparameters

In this section, we investigate the different hyperparameters of the proposed *BERT*-based rescoring models. We use 4-gram LM for recognition. As mentioned previously, during rescoring the LM is not modified. We study BERT$_{sem}$ rescoring model only. The obtained results are valid for BERT$_{als}$ model. We downloaded the pre-trained *BERT* models provided by Google [17]. Acoustic and LM probabilities are not used in these experiments. They will be used in the overall evaluation.

Figure 2 presents the results on the development corpus using *BERT$_{sem}$* model with different amounts of data. As acoustic and linguistic probabilities are not used here, fine-tuning data is performed using only text data. 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 model.

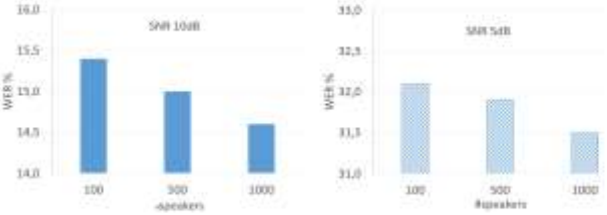

Figure 2: *ASR WER (%) on the TED-LIUM development corpus as function of the amount of BERT fine-tuning data: 100, 500 and 1000 training speakers. The left bar chart corresponds to 10dB SNR, right to 5dB SNR, 4-gram LM, BERT model with 8 layers and dimension 128. Rescoring model is BERT$_{sem}$.*

Figure 3 shows the recognition performances as a function of the number of layers of the BERT$_{sem}$ model. Using 12 layers gives the best performance for the two SNR levels. We observe that this parameter plays an important role.

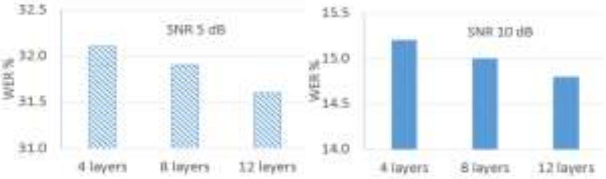

Figure 3: *ASR WER (%) on the TED-LIUM development corpus according to the number of layers for the BERT_sem rescoring model. The left bar chart corresponds to 10 dB SNR, right to 5 dB SNR, 4-gram LM, BERT model (dimension 128) fine-tuned using 1000 speakers.*

We performed also the preliminary experiments with different number of heads. The more heads is the better. For lack of space we do not give these results in this article.

In conclusion, we can say that for the *BERT*-based rescoring models, it is important to use a large corpus for fine-tuning the model and to choose a model with many hidden layers. In the overall experiments we use pre-trained *BERT-base* model with 110M parameters, 12 layers and the size of the hidden layers of 768, fine-tuned using 1000 speakers.

4.2. Overall results

We reports the WER for the development and the test sets of TED-LIUM with clean speech and in noise conditions of 10 and 5 dB. In tables, the first line of results (method *Random*), corresponds to the random selection of the recognition result from the N-best hypotheses without the use of the proposed rescoring models. The second line of tables (method *Baseline*), corresponds to not using the rescoring models (standard ASR). The last line of tables (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 tables give the performance of the proposed approaches.

To compare the proposed Transformer-based models to other Transformer-based error correction models introducing long range context dependences, we experimented a rescoring based on GPT-2 model (called *GPT-2 comb. with ac. score* in tables). It corresponds to the rescoring of N-best hypotheses using the formula (1) with linguistic scores given by pre-trained GPT-2 model, instead of n-gram (semantic model is not used).

For *BERT*-based rescoring models, we studied 3 configurations:

- Rescoring using only the scores *score(h)* computed with *BERT*-based rescoring methods (denoted $BERT_x$ in tables).
- Rescoring using a combination of the *BERT*-based *score(h),* and the acoustic score $P_{ac}(h)$ ($BERT_x$ *comb. with ac. score* in tables). In this case, *score(h)* is used as a *pseudo probability* and multiplied (combined) to the acoustic likelihood with a proper weighting factor γ. $P_{lm}(h_i)$ is not used in this combination.
- Rescoring using a combination of the *BERT*-based score, the acoustic score $P_{ac}(h)$ and the linguistic score $P_{lm}(h)$ ($BERT_x$ *comb. with ac./ling. scores* in tables). To have a fair comparison with the state-of-the-art of rescoring, for the most efficient $BERT_{als}$ model we use GPT-2 score as linguistic score ($BERT_{als}$ *comb. with ac./GPT-2 scores* in tables).

For all experiments, combination weights are: α=1, β is between 8 and 10 and γ is between 80 and 100.

From table 2 we observe that for all conditions and all evaluated rescoring models, the proposed rescoring models outperform the baseline system. This shows that the proposed Transformer-based rescoring models generating dynamic embeddings are efficient to capture a significant proportion of the semantic information. Combining the acoustic score with the $BERT_{sem}$ model ($BERT_{sem}$ *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, combining the linguistic score alone with the *BERT* rescoring gives no improvement compared to the *BERT* model. We do not present this result in the tables. Google's BERT model, trained on the billions of sentences, probably captures the linguistic structure of the language better than a simple n-gram LM trained on a much smaller corpus. Using the linguistic and acoustic scores with the *BERT* rescoring model ($BERT_{sem}$ *comb. with ac./ling. scores*) brings small additional improvement.

For *BERT*-based results, all improvements are significant compared to the baseline system (confidence interval is computed according to the matched-pairs test [6]). On the test set, $BERT_{als}$ *comb.w ith ac./ling. scores* obtains an absolute improvement of 2.8 % for 10 dB (18.3 % WER versus 21.1 % WER), 3.4 % for 5 dB (36.9 % versus 40.3 %), 1.9 % for clean speech (6.8 % versus 8.7 %) compared to the baseline system. This corresponds to about 13 % (for 10 dB), 8 % (for 5 dB) and 21 % (for clean) of relative improvement. Compared to the *GPT2 comb. with ac. scores,* $BERT_{als}$ *comb. with ac./GPT-2 scores* allows to obtain additional improvement. In all configurations this improvement is significant (denoted by "*" in tables).

As n-gram LM is limited in its ability to model language context (long-range dependencies), we performed the ASR experiments using more powerful LSTM-RNNLM. Table 3 reports the results for the same set of experiments, but, instead of n-gram, the LSTM-RNNLM is used during the speech recognition. The proposed rescoring methods give consistent improvements as for n-gram LM results. So, all previous observations are valid for RNNLM-based experiments. The best system ($BERT_{als}$ *comb. with ac./ling. scores)* gives between ?? % and ?? % of relative improvement compared to the baseline system. These improvements are also significant. In the case of RNNLM, the improvement is smaller compared to the 4-gram case. This can be due to the fact that RNNLM can take into account more distant dependencies than n-gram.

5. Conclusions

In this article, we focus on the task of automatic speech recognition in clean and noisy conditions. Our methodology is based on taking into account semantics through representations that capture the semantic characteristics of words and their context. The semantic information is taken into account through a rescoring module on ASR N-best hypotheses. We proposed two effective DNN approaches based on the *BERT* model: one approach use BERT-finetuning and represent purely semantic model. Second approach uses DNN-model trained using semantic, acoustic and linguistic information together. To evaluate our methodology, the corpus of TED-LIUM conferences is used. The best system $BERT_{als}$ *with ac./ling. scores* gives about 8 % (4-gram) and 4.5 % (RNNLM) for 10 dB; about 6 % (4-gram) and 4.6 % (RNNLM) for 5 dB of relative improvement compared to the baseline system. These improvements are statistically significant.

Table 2: *ASR WER (%) on the TED-LIUM development and test sets, SNR of 10 and 5 dB. 20-best hypotheses,* ***4-gram LM****. BERT model fine-tuned on 13.2M hypotheses pairs. "*" denotes significantly different result compared to "GPT2 comb with ac. score" configuration*

4-gram LM	***SNR 5 dB***		***SNR 10 dB***		***no added noise***	
	dev	test	dev	test	dev	Test
Random	33.5	41.3	16.9	22.9	10.6	12.1
Baseline system (*ac.and ling. Scores)*	32.7	40.3	15.7	21.1	8.7	8.9
GPT2 comb. with ac. scores	30.0	37.1	13.1	17.9	6.8	7.3
$BERT_{sem}$	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./ling. scores*	30.6	37.9	14.1	19.4	7.8	8.5
$BERT_{als}$	30.4	37.5	13.5	18.6	6.9	7.3
$BERT_{als}$ *comb. with ac./ling. scores*	30.2	36.9	13.4	18.3	6.8	7.0
$BERT_{als}$ *comb. with ac./GPT-2 scores*	**29.7**	**36.6***	**12.8**	**17.5***	**6.4***	**6.6***
Oracle	27.5	33.2	11.1	15.0	5.2	4.7

Table 3: *ASR WER (%). 20-best hypotheses list. TED-LIUM development and test sets, SNR of 10 and 5 dB,* ***RNNLM****. BERT model fine-tuned on 13.2M hypotheses pairs.*

RNNLM	***SNR 5 dB***	***SNR 10 dB***	***no added noise***

	dev	test	dev	test	Dev	test
Random	29.2	38.4	13.9	20.2	8.9	10.8
Baseline system (*ac. And ling. Scores)*	28.2	37.1	12.3	17.7	6.6	7.2
GPT2 comb. with ac. scores	26.2	34.9	11.0	15.9	6.1	6.7
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./ling. scores*	26.5	35.4	11.5	16.9	6.0	6.6
BERT$_{als}$	26.1	35.2	11.0	16.4	5.9	6.6
BERT$_{als}$ *comb. with ac./ling. scores*	25.9*	34.5*	10.8*	15.9	5.6*	6.1*
BERT$_{als}$ *comb. with ac./GPT-2 scores*	**25.5***	**34.4***	**10.4***	**15.4***	**5.4***	**5.7***
Oracle	23.1	30.2	8.3	12.1	3.8	3.5

6. References

.

[1] A. Bayer, G. Riccardi, “Semantic Language Models for Automatic Speech Recognition”, *Proceedings of the IEEE Spoken Language Technology Workshop (SLT),* 2014.

[2] R. Corona, J. Thomason, R. Mooney, “Improving Black-box Speech Recognition using Semantic Parsing”, *Proceedings of the The 8th International Joint Conference on Natural Language Processing*, pp.122–127, 2017

[3] J. Devlin, M.-W. Chang and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding”, *NAACL-HLT*, 2019.

[4] H. Fernandez, H. Nguyen, S. Ghannay, N. Tomashenko and Y. Esteve, “TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation”, *Proceedings of SPECOM*, pp. 18–22, 2018.

[5] J. Gaspers, P. Cimiano, B., “Semantic parsing of speech using grammars learned with weak supervision”, *Proceedings of the HLT-NAACL*, pp. 872-881, 2015.

[6] L. Gillick and S. Cox S, “Some Statistical Issues in the Comparison of Speech Recognition Algorithms”, *Proceedings of ICASSP*, v. 1, pp. 532-535, 1989.

[7] T. Kenter, A. Borisov, M. de Rijke, « Siamese CBOW: Optimizing Word Embeddings for Sentence Representations”, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 941–951, Berlin, Germany, August 7-12, 2016.

[8] S. Level, I. Illina and D. Fohr, “Introduction of semantic model to help speech recognition”, *International Conference on Text, Speech and Dialogue*, 2020.

[9] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, “An overview of noise-robust automatic speech recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, Apr. 2014.

[10] T. Mikolov, S. Kombrink, L. Burget, J.-H. Cernocky, S. Khudanpur, “Extensions of recurrent neural network language model”, *Proceedings of the ICASSP*, pp. 5528–5531, 2011.

[11] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, “Distributed representations of words and phrases and their compositionality”, *Advances in Neural Information Processing Systems,* 26, pp. 3111-3119, 2013.

[12] A. Ogawa, M. Delcroix, S. Karita and T. Nakatani, "Rescoring N-Best Speech Recognition List Based on One-on-One Hypothesis Comparison Using Encoder-Classifier Model," *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP*, 2018.

[13] 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”, *Proceedings of IEEEWorkshop on Automatic Speech Recognition and Understanding (ASRU),* 2011.

[14] I. Sheikh, D. Fohr, I. Illina, G. Linares, “Modelling Semantic Context of OOV Words in Large Vocabulary Continuous Speech Recognition”, *IEEE/ACM Transactions on Audio, Speech and Language Processing, Institute of Electrical and Electronics Engineers*, 25 (3), pp.598 – 610, 2017.

[15] J. Shin, Y. Lee, K. Yung, “Effective Sentence Scoring Method Using BERT for Speech Recognition”, *Proceedings of ACML,* 2019.

[16] Y. Song, D. Jiang, X. Zhao, Q. Xu, R. Wong, L. Fan and Q. Yang. “L2RS: a learning-to-rescore mechanism for automatic speech recognition”, *arXiv:1910.11496*, 2019.

[17] I. Turc, M.-W. Chang, K. Lee and K. Toutanova, “Well-Read Students Learn Better: On the Importance of Pre-training Compact Models*”, arXiv:1908.08962v2*, 2019.

[18] A.Varga and H. Steeneken, “Assessment for automatic speech recognition II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems”, *Speech Communication*, Volume 12, Issue 3, pp. 247-251, 1993.

[19] A. Wang and K. Cho, “BERT has a mouth, and it must speak: BERT as a Markov random field language model”, *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation,* pages 30–36, 2019.

[20] https://github.com/hanxiao/bert-as-service/

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

[22] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, “Language Models are Unsupervised Multitask Learners”, 2019.

[23] M. Sundermeyer, R. Schluter, and H. Ney, “Lstm neural networks for language modeling,” in Thirteenth Annual Conference of the International Speech Communication Association, 2012.