BERT Attends the Conversation:  
Improving Low-Resource Conversational ASR

Pablo Ortiz  
*Telenor Research*  
1331 Fornebu, Norway  

Simen Burud  
*Department of Computer Science*  
*Norwegian University of Science and Technology*  
7491 Trondheim, Norway

Abstract

We propose new, data-efficient training tasks for BERT models that improve performance of automatic speech recognition (ASR) systems on conversational speech. We include past conversational context and fine-tune BERT on transcript disambiguation without external data to rescore ASR candidates. Our results show word error rate recoveries up to 37.2%. We test our methods in low-resource data domains, both in language (Norwegian), tone (spontaneous, conversational), and topics (parliament proceedings and customer service phone calls). These techniques are applicable to any ASR system and do not require any additional data, provided a pre-trained BERT model. We also show how the performance of our context-augmented rescoring methods strongly depends on the degree of spontaneity and nature of the conversation.

**Keywords:** BERT, conversational ASR, rescoring, low-resource ASR, language modelling

1. Introduction

In recent years, end-to-end Deep Learning-based systems have proved to be very successful, see for example the works of Graves (2012); Chorowski et al. (2015); Amodei et al. (2015); Watanabe et al. (2017); Dong et al. (2018); Synnaeve et al. (2019). In contrast to HMM-based pipelines, this approach requires very little domain expertise to train since they take the raw audio as input\(^1\) and the target transcript as output. All feature extraction and engineering needed is learned implicitly during training. This makes it relatively easy to adapt an ASR system to a new domain.

Even though these systems are termed end-to-end, most can be decomposed into three distinct components: the acoustic model (AM), the decoder algorithm, and the language model (LM). The acoustic model is primarily concerned with encoding the audio to a sequence of token probability vectors. This sequence is typically decoded using a wide, compute-intensive beam search, often guided by a language model. The top-scoring result from the beam search is usually not the optimal one, so the candidate list can be re-scored

---

\(^1\) Generally, a numerical representation such as MFCCs is extracted from the power spectrum of the Fourier transform of the signal, although there are also attempts to learn input representations using neural networks, see for instance the review of Latif et al. (2020).

©2021 Pablo Ortiz and Simen Burud.
with a secondary LM with the aim of improving the accuracy of the ranking, see e.g.,
(Williams, 2008; Singh et al., 2017; Ogawa et al., 2018; Huang and Peng, 2019; Shenoy
et al., 2021b; Chiu and Chen, 2021; Chiu et al., 2021) and references therein.

ASR systems usually do not make use of context beyond the current utterance. This
lack of information makes it harder to disambiguate phonetically similar or ambiguous
transcripts, and the model often ends up outputting the statistically most common inter-
pretation while disregarding the bigger picture. For this reason there has recently been a
significant amount of research to include dialog context in different forms when transcribing
an utterance in a dialog both in language modelling (Ji and Bilmes, 2004; Xiong et al., 2018)
and ASR (Kim and Metze, 2018; Kim et al., 2019; Shenoy et al., 2021b,a). This task is
facilitated by the rise of LM pre-training approaches able to compute deeper contextual
representations (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2018; Brown et al.,
2020).

Even though ASR systems see widespread deployment, they are much less reliable when
applied to low-resource languages and real-life situations. One can seek help from general-
purpose LMs pre-trained on massive text corpora, which obtain state-of-the-art results in a
wide range of NLP tasks with relatively little task-specific fine-tuning (Devlin et al., 2018;
Brown et al. 2020). These models can be used to aid ASR in a variety of ways, while
we focus on using the conversational context learnt by BERT (Devlin et al., 2018) for the
resoring of ASR hypotheses (Huang and Peng, 2019; Chiu and Chen, 2021; Shenoy et al.,
2021b; Chiu et al., 2021). Our work is similar in spirit to that of Chiu and Chen (2021),
however we pursue a different fine-tuning strategy for BERT and apply it to domain-specific,
spontaneous conversations in low-resource languages. Our contributions are summarized
below.

First, we explore to what extent an LM such as BERT (Devlin et al., 2018) improves
ASR results after training on a conversational disambiguation task, when the acoustic model
(AM) and n-gram LM have only been trained on single, independent utterances, which is our
baseline system. We rescore the N-best results with context-augmented BERT, obtaining
significant improvements over the baseline. Making conversational context available to
BERT in the form of previous utterances is key to our approach. We find that the amount
of context required depends on the target domain.

In particular, formal parliamentary discussions greatly benefit from large conversational
contexts, while shorter contexts are favoured when rescoring the more unpredictable, spon-
taneous conversations from our dataset of calls to customer service. Further, we find that
fine-tuning procedures play a significant role in BERT’s ability to rescore the N-best lists.
We propose a data-efficient strategy that uses the baseline ASR system to generate suffi-
cient training examples for BERT, even from a tiny dataset. Fine-tuning BERT this way
on a small number of relevant samples performs far better than doing it on a much larger,
out-of-domain conversational dataset.

The outline of the paper is as follows. The next section introduces the baseline ASR
system, Section 3 presents our LM integration approach, Section 4 details our LM training
strategies, Section 5 describes the data, Section 6 discusses the experiments and results,
and finally in Section 7 we present our conclusions and future work.
2. Baseline ASR system

Our baseline ASR system is based on Deepspeech 2 (Amodei et al. 2015), a family of ASR systems based on deep neural networks and the CTC framework (Graves et al. 2006). It processes each audio feature only in the context of neighboring features using convolutional layers and then combines them using bidirectional RNN layers. Curriculum learning is applied during the first epoch, using transcript length (in characters) as a proxy for difficulty. We make use of an existing PyTorch implementation\(^2\) as a code basis for our model.

The model operates at character level. Due to the independence assumption made in CTC, its implicit language model is often found to be weak compared to that of attention-based sequence-to-sequence models, see (e.g., Chan et al., 2016). Instead, Deepspeech models rely partly on an external n-gram LM during decoding. We train n-gram models using the implementation of Kneser-Ney smoothed n-gram estimation from Heafield et al. (2013)\(^3\) We train separate models for each speech corpus and prune unique n-grams of order 2 and above. Each utterance is treated as a separate document, meaning no context information from previous utterances is available in the n-gram models.

We train the primary AM with a Norwegian public dataset (NST) consisting of 394.5 hours of recorded and transcribed speech, where 300 hours are used as the training dataset (see Section 5 for further details on the data). The model architecture consists of three strided convolutional layers, nine 1200-dimensional, bidirectional GRU layers, and a fully connected layer with softmax.

As we describe in detail below, to construct the different baseline models on other datasets, we fine-tune the primary AM and build specific LMs using the datasets in question.

We test the performance of our primary AM on NST data by combining it with a 5-gram LM in a beam search of size 64. The LM is trained with approximately 13 million sentences from a non-public corpus gathered by NST consisting of newspaper text.\(^4\) We achieve a word error rate (WER) of 2.87% and character error rate (CER) of 1.16% on the test set, showing that for sufficiently large, clean, generic datasets, these ASR models are rather successful. However, that is certainly not the case when working with small datasets containing noisy, topic-specific, unstructured conversational speech, which motivates the extensions in this work.

In the next section we describe how we augment our baseline system by introducing BERT models. We emphasize that the choice of baseline system is purely circumstantial, and that the extensions discussed in what follows only require the ASR system to produce transcription candidates.

---

\(^2\) https://github.com/SeanNaren/deepspeech.pytorch

\(^3\) https://github.com/kpu/kenlm version 35835f1.

\(^4\) We thank August Moum and Skjalg Winnerdal for providing access to this model. It was trained with a version of the newspaper corpus curated at the Norwegian University of Science and Technology (NTNU) during the SVoG project in collaboration with other institutions.
3. N-best rescoring

We distinguish between three main elements within the ASR system: acoustic model (AM), beam search (BS) and language model (LM).

The AM takes an audio segment $x$ as input and outputs a matrix $Z$ such that each element $z^t_k$ represents the probability of token $k$ at time $t$. The BS decodes $Z$ into a set $Y$ of the $N$ best candidate transcripts, aided by an n-gram LM through shallow fusion (Gülcüehre et al., 2015). That is,

$$P_{BS}(y|x) = P_{AM}(y|x) + \alpha P_{LM1}(y) + \beta \text{WC}(y),$$

where $P_{AM}$ is the sum of path scores for the transcript as estimated by CTC prefix beam search (Graves et al., 2006), $P_{LM1}$ is the transcript’s probability according to a Kneser-Ney n-gram LM, and WC is a word count bonus.

Rather than training the LM on large external datasets, we obtain slightly better performance when training it solely on the transcripts used to fine-tune the AM. We thus observe that for domain-specific datasets, relevance (or topic overlap) compensates for abundance$^5$.

We denote the highest-ranked transcript from the beam search as

$$y^* \equiv \arg\max_{y \in Y} P_{BS}(y|x). \quad (1)$$

For the baseline system, $y^*$ is returned as the final transcript, and its WER with respect to the ground truth $y_{gt}$ is used as the evaluation metric.

We extend this system to perform N-best$^6$ rescoring of $y \in Y$ with BERT to obtain the final ranking. To do that, we interpolate the scores from the beam search and the secondary language model (BERT):

$$P_{\text{rescored}}(y|x) = (1 - \gamma)P_{BS}(y|x) + \gamma P_{\text{BERT}}(y), \quad (2)$$

where $\gamma$ is an adjustable weight parameter. The top candidate after rescoring is denoted by

$$y' \equiv \arg\max_{y \in Y} P_{\text{rescored}}(y|x). \quad (3)$$

N-best rescoring does not make assumptions about the baseline ASR system, making the extension widely applicable. Instead, rescoring makes good use of the many candidate transcripts returned from the wide beam searches typical to CTC-based pipelines.

N-best rescoring exploits the fact that $y^*$ in (1) is often a suboptimal choice of transcript from $Y$. To identify the upper performance bound, we can define an oracle rescoring algorithm that always selects the best transcript from those present in $Y$:

$$y_o \equiv \arg\min_{y \in Y} \text{WED}(y_{gt}, y), \quad (4)$$

where WED is the word-level (Levenshtein) edit distance between two transcripts.

---

$^5$ Our primary AM obtains its best results when combined with an LM trained on abundant external data, but this stems from the fact that the primary AM is trained on data covering an extensive variety of topics, similar to that covered by the external LM.

$^6$ $N = 1024$ throughout the rest of this paper.
4. BERT training strategies

We use BERT as described by Devlin et al. (2018)\(^7\) as our rescoring language model in (2). A challenge in using BERT is that transcripts of spontaneous conversations read very differently from, for example, a novel. People use a very different sentence structure when speaking (such as incomplete sentences and repetitions) and a different choice of words, including fillers and hesitations. In dialogues there are also many cases of overlapping speech and interruptions. Thus the language representations built when training on written text are suboptimal for inference on conversations. To deal with this, BERT needs to be fine-tuned to the task at hand, and also learn spoken language. We propose three different strategies for training BERT, as explained below.

4.1 MLM & NSP

The simplest solution is to keep training BERT with the same objective used during pre-training as Devlin et al. (2018), i.e., Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). These training tasks are known to teach BERT a good language representation. Previous work such as that of Shin et al. (2019) uses a BERT variant trained on these tasks to score sentences and interpolate those LM scores to rescore the hypotheses from a LAS (Chan et al., 2016) acoustic model. Using the NSP head for scoring is straightforward, as it already outputs the probability directly. With the MLM output nodes, we can score sentences as follows (Shin et al., 2019):

1. Pass the sequence \(y = y_1 \ldots y_L\) through the model, masking the first position.
2. Among the output probabilities, note the probability of the token at the first position being the original token, denoted \(P(y_1|y_2, \ldots, y_L)\).
3. Repeat for each remaining position in the sequence and calculate \(P(y) = \prod_{l=1}^L P(y_l|y_\bar{l})\), where \(y_\bar{l}\) denotes all elements in the sequence except \(y_l\).

4.2 Conversational NSP

In NSP as proposed by Devlin et al. (2018), BERT’s input is fully packed with text, with the separator being placed anywhere inside the input text, dividing it into segments rather than sentences. This reduces BERT’s ability to make use of the conversational aspect of our domain. To remedy this, we adapt the concept of NSP to conversations: positive samples are simply triplets of consecutive utterances, while negative samples are generated by replacing the third utterance with a different one.

Lan et al. (2019) propose a sentence order prediction task with pairs of sentences which creates negative samples by swapping the order of the sentences in the pair. Other objectives are studied by Liu et al. (2019), sampling contiguous sentences from the same document or from separate ones. Even though the results of Liu et al. (2019); Lan et al. (2019) improve with respect to the BERT baseline for the datasets they tested, these objectives are not suitable for our spontaneous conversational data: preliminary experiments indicate that

---

\(^7\) We use the implementation by Wolf et al. (2020), in particular the model available at https://huggingface.co/NbAiLab/nb-bert-base/tree/6aad23f6c2ed0df8498397175ec07d497f6319a1, pre-trained in Norwegian.
sampling from the same conversation is too hard a task. We attribute this to our type of data, where overlapping speech often makes the sentence order ambiguous. In addition, quick turn-taking, short utterances and what appears to be sudden topic changes (without the presence of a much larger context and knowledge base) are ubiquitous. Hence, to make the task more feasible for BERT, we use triplets of utterances and create negative samples by replacing the third utterance by one from another dialogue.

4.3 Disambiguation

With the training tasks proposed so far, there is an apparent train-inference mismatch: the N-best lists primarily contain variations of the same text, often with only a few words differing. Spelling and grammatical errors are also far more common, neither of which are captured by the preceding tasks.

To remedy this, we propose a disambiguation task where BERT's classification head is trained to distinguish the best transcript available from similar candidates. Given a dataset $\mathcal{D}$ of candidate transcripts $Y$ and ground-truth conversational context $c$, we optimize

$$\min_{\tilde{y}} \sum_{(c,Y) \in \mathcal{D}} \left[ \mathcal{L}(1, P_{\text{BERT}}(\tilde{y}|c)) + \sum_{y \neq \tilde{y}} \mathcal{L}(0, P_{\text{BERT}}(y|c)) \right],$$

where $\mathcal{L}$ is the cross-entropy loss function and $\tilde{y}$ is either the ground truth $y_{gt}$ or $y_o$ in (4). As we will show in Section 6, using $\tilde{y} = y_o$ as the target is more reliable than $\tilde{y} = y_{gt}$. Indeed, $y_{gt}$ is sometimes very distinct from the predicted transcripts in $Y$, which makes the task rather easy and far from reality at inference time.

The sum in the second term is over the remaining transcripts in $Y$. To avoid a high imbalance, we sum over only two randomly sampled transcripts with strictly higher WER than $y_o$, which yields a better performance on test.

5. Data

As mentioned earlier, our research on LMs focuses on the low-resource data domain with respect to three dimensions: language (Norwegian), tone (spontaneous, conversational), and topics (parliament proceedings and customer service phone calls). The NST dataset used to train the primary AM is however more standard in tone and topics, consisting of grammatically correct, planned, noise-free speech.

A particularly challenging aspect of the Norwegian language is that it has two official written standards, a notably rich spectrum of spoken dialects (Skjekkeland, 2010), and no unequivocal correspondence between many dialect words and the official written form. Thus, transcribers usually follow certain conventions to ensure consistency and produce transcriptions as faithful as possible to the speech while adapting to the written standards. This is one of the core problems in ASR for Norwegian, in addition to data scarcity. The datasets used are described below and relevant properties are displayed in Table 1:
|               | NPSC | TNCS | NST  |
|---------------|------|------|------|
| Number of utterances | Train | 23720 | 8532 | 180237 |
|                | Eval  | 1279 | 1065 | 20321  |
|                | Test  | 1378 | 940  | 3370   |
| Average tokens per utterance | Train | 18.77 | 11.46 | 11.54 |
|                | Eval  | 22.93 | 10.96 | 10.05  |
|                | Test  | 18.9 | 10.66 | 11.70  |
| Blank tokens (%) |      | 2.44  | 5.36  | 0      |
| Vocabulary size |      | 26890 | 4368  | 78221  |
| OOV test (%)    |      | 2.77  | 5.24  | 0.95   |

Table 1: Data specifications. A “blank token” is used to transcribe unintelligible sounds, fillings, and hesitations. “Vocabulary size” denotes the number of unique words in the data. “OOV test” is the percentage of words in the test set not found in the rest of the dataset (out of vocabulary).

NPSC: Norwegian Parliamentary Speech Corpus. This public dataset contains 58h of official proceedings from plenary meetings at the Norwegian parliament, reduced to 52h after cleaning. Recordings and transcriptions are split into sentences. Importantly, inaudible and/or unintelligible sounds and words, as well as fillings and hesitations, are transcribed as the blank token. The split in train, evaluation and test sets was made by placing entire plenary sessions into different sets so that the topics also are split as much as possible, making the test closer to real situations.

TNCS: Telenor Norway Customer Service. This internal dataset contains 149 conversations with the Norwegian customer service call center of Telenor. Recordings are made in two channels (customer and agent). For each channel, we divide the audio into shorter utterances. There is a substantial amount of overlapping speech and interruptions, making the sentence order ambiguous, which is especially relevant when training BERT with conversational context. Furthermore, that and the spontaneous tone cause the data to have a highly ungrammatical structure, making the LMs’ task much harder. In addition, names, addresses and telecommunications-specific terms on the test data are mostly OOV tokens, so in that case the decoder outputs out-of-context candidates due to vocabulary constraints. Noise, mumbling or unintelligible speech is transcribed as the blank token. The train-eval-test split was made by placing entire conversations into different sets.

One important feature of the TNCS data is that the distribution of number of tokens per utterance is extremely skewed towards low values and with a very large tail, despite having an average similar to NST data (see Table 1). This is natural for customer service

---

8. Datasets and documentation are available from [https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-58/](https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-58/) We note that at the time we performed this work, only the beta version was available, containing 58h in contrast to the 140h available at the time of writing.
9. This is not to be confused with the blank token used by the CTC decoder, which signals a separation between tokens.
10. We use the WebRTC VAD Python package available from [https://github.com/wiseman/py-webrtcvad](https://github.com/wiseman/py-webrtcvad)
conversations where quick turn-taking occurs, as well as long explanations. We will see this has a notable impact on our results.

**NST:** Nordic Speech Technology. This public dataset is only used to train the primary AM, hence we just describe the relevant details. After processing we retain 339h of data divided into (300, 33, 6) hours for (train, evaluation, test) sets. In particular, for train and evaluation sets, we remove utterances with less than three words and those just containing three repetitions of the same word. For testing we select only fully grammatical, complete sentences.

### 6. Experiments and results

We test several variations of shallow fusion with n-gram language models and N-best rescoring strategies with BERT. First, we train a baseline AM with greedy decoding by fine-tuning the primary AM described in Section 2. Next, we replace the greedy decoding with beam search, which is run with and without n-gram fusion. Then, we rescore the N-best lists from each beam search output using several BERT models trained on different tasks. We publish our source code to make our results reproducible and to encourage further research.

Our results report the total WER as the total edit distance for all samples divided by the total number of words. This penalizes every mistake equally as opposed to averaged utterance-level WER, that places higher penalties on errors in short transcripts. This distinction is particularly important for our results on the TNCS dataset, where a large portion of the transcripts are very short. For N-best rescoring, we use the WER recovery rate (WERR) to measure improvements with respect to the gold standard (oracle). In the context of N-best rescoring as described in Section 3, it is defined as

\[
\text{WERR}(y') = \frac{\text{WER}(y_s) - \text{WER}(y')}{\text{WER}(y_s) - \text{WER}(y_o)},
\]

where \(y_o\) is the closest transcript to the ground truth and \(y_s, y'\) are the top-ranked transcripts before and after rescoring, respectively. WER(·) is measured with respect to the ground truth \(y_{gt}\).

#### 6.1 Setup

For the beam search, we use a highly optimized, data-parallel beam search implementation. We represent the beam search as a prefix tree, denoting the N highest-ranking nodes “beam nodes”. At each time step, all non-beam nodes without beam node descendants are pruned.

---

11. Datasets and documentation available at [https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-13/](https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-13/).

12. The optimization scheme of the AM is based on constant learning rate annealing, hence does not depend on the decoding strategy.

13. [https://gitlab.com/sburud/master](https://gitlab.com/sburud/master).

14. [https://github.com/parlance/ctcdecode](https://github.com/parlance/ctcdecode).
At every time step, we consider adding every possible character to every beam node, except for characters only leading to OOV words. If the new character is non-blank\textsuperscript{15}, we add the n-gram and word count bonuses.

For BERT, we use a batch size of 768 and otherwise apply the hyperparameters listed under BERT\textsubscript{base} in (Devlin et al., 2018). Due to the difference in size and topics across our data, training on multiple datasets simultaneously negatively impacts performance. Hence, when we list a BERT model trained on multiple datasets, this is to be understood as sequential training.

6.2 Hyperparameter tuning

We perform hyperparameter tuning in two stages: first the beam search parameters, and then the rescoring interpolation parameter, see Section 3.

For the beam search parameters, we perform different random searches for each task on the evaluation sets using Optuna (Akiba et al., 2019):

- Tune $\alpha$ and $\beta$, minimizing $\sum \text{WED}(y^*, y_{gt})$. The parameters found are used to report baseline WER for shallow fusion.

- Tune $\alpha$ and $\beta$, minimizing $\sum \text{WED}(y_o, y_{gt})$. These parameters are used in rescoring experiments.

Using the optimal beam search parameters, we rescore all candidates in $Y$ using BERT. We then perform a grid search for the interpolation parameter $\gamma \in [0, 0.5]$ with step size 0.001.

6.3 N-best rescoring

Table 2 shows rescoring performance for the various BERT fine-tuning strategies. For the sake of efficiency, not all experiments are performed on both datasets, since our tests indicated that conclusions hold across our datasets.

On the one hand, the conversational NSP (C-NSP) objective does not help in rescoring, which we detail in Section 6.4. On the other hand, the disambiguation task improves performance significantly. This shows that more realistic training of transformer-based LMs does matter. In line with Devlin et al. (2018), we found that BERT learns a sufficient language representation to adapt to new writing styles and language use with only minimal fine-tuning.

The fact that the optimal value of $\gamma$ for test and evaluation is different is due to the small size of the datasets, which causes a significant variance among the splits.

6.3.1 Disambiguation with conversational context

Much of the rescoring improvements originate from the presence of conversational context. As shown in Table 2, introducing just two utterances of conversational context already improves rescoring results.

\textsuperscript{15} In CTC decoding, a blank token is used to separate tokens, see (Graves et al., 2006) for details. Not to be confused with the blank tokens used in the transcriptions as described in Section 5.
Decoding strategy | TNCS | NPSC |
|------------------|------|------|
|                  | WER  | WERR | WER  | WERR |
| Greedy decode    | 48.86| -    | 37.70| -    |
| BS with vocabulary| 38.44| -    | 30.70| -    |
| BS with 2-gram   | 33.27| -    | 23.39| -    |
| + Conversational NSP | 33.27| 0    | -    | -    |
| + Disambiguation (no context) | 32.99| 3.33 | -    | -    |
| + Disambiguation (short context) | **32.80** | 5.60 | 22.54 | 11.99 |
| + Disambiguation (long context) | 33.09| 2.14 | 20.75 | 37.24 |
| + Oracle rescoring | **24.87** | 100  | **16.30** | 100  |
| BS with 6-gram   | 32.96| -    | 23.05| -    |
| + Disambiguation (short context) | **32.74** | 3.09 | 22.26 | 11.37 |
| + Disambiguation (long context) | -    | -    | **20.64** | 34.68 |
| + Oracle rescoring | **25.84** | 100  | **16.10** | 100  |

Table 2: Results obtained with the different decoding strategies on test sets. Note that the value of $\gamma$ used to report results on the test data is obtained from minimizing WER on the evaluation data for each experiment. We use the total WER evaluated on the test split of each dataset, and report numbers as percentages. WERR is calculated for each block using the plain BS + n-gram model as baseline and the corresponding oracle rescoring as the gold standard, see (5).

Figure 1: WER as a function of $\gamma$ on the TNCS test set. Reported values (grey dots) correspond to those in Table 2, where the value of $\gamma$ is given by optimizing on the evaluation set. In the case of C-NSP, no improvement is reported since performance is consistently worse than the baseline on the evaluation set, even when small gains are observed on test.
For the TNCS data, extending the context to five utterances reduced performance with respect to short (or no) context, see Figure 1. Indeed, for unstructured conversations with quick turn-taking and topic changes, too long a context misleads BERT.

In contrast, NPSC benefits greatly from the longer context, obtaining 37.24% and 34.68% WERR after rescoring 2- and 6-gram shallow fusion models, respectively. With the longer context, significantly higher values for the interpolation weight $\gamma$ are favoured, as shown in Figure 2. The improvements plateaued around $\gamma \approx 0.2$, but unlike the other experiments, a much higher $\gamma$ value was needed before we observed performance degradation. We attribute this to the long-form speeches being much more similar to the pre-training data. Further, the debates focus on a specific topic and are more structured, making it more likely that previous context is relevant when rescoring the current hypotheses.

### 6.3.2 Sequence length and bounds for improvements

For long utterances, the ratio between potential transcripts and $N$ grows quickly, making the gap between upper and lower WER bounds rather narrow. On the other hand, CTC beam search will perform a near exhaustive search for short utterances, giving a much wider gap. This can be seen in Figure 3, which shows the empirical WER bounds as a function of utterance length, along with WER($y_*$) (baseline), and WER($y'$), obtained after N-best rescoring with BERT fine-tuned on disambiguation with short or long context.

When increasing the context size available to BERT from two to five utterances, Figure 3 clearly shows that the results improve more for shorter than longer utterances. Our interpretation is that short utterances are less prone to topic changes, often containing answers or follow-up questions, and thus are more consistent with the context. If we add to this having large part of the search space available, the task for BERT becomes easier.
Figure 3: Total WER as a function of $y_{gt}$ word count (length) for the NPSC evaluation set. Bin $i$ denotes lengths within the interval $[i, i+4]$. “Short (long) context” rescores the baseline candidates with a disambiguation-BERT model fine-tuned with two (five) context utterances.

For long sequences the gap between the best and worst transcripts is rather narrow due to the little variation among candidates. This severely limits BERT’s ability to improve the results.

6.4 ASR-independent disambiguation with BERT

We have explored different training schemes for BERT. In Section 6.3 we presented our results from integrating BERT as an LM for rescoring, and evaluated the WERR on ASR. Here we provide a deeper analysis of how those BERT fine-tuning strategies perform during disambiguation of ASR hypotheses independently of the scores given by the AM and n-gram LM.

In order to do that, we take the BERT models fine-tuned on C-NSP (see Section 4.2) or disambiguation (see Section 4.3) with short context and give them two ASR hypotheses: $y_0$ and another one from within the same beam. Then we measure their accuracy on the task of identifying $y_0$ under different training conditions. The difference with respect to N-best rescoring is that we only provide two candidates and a flat prior (same initial score).

Our findings are summarized in Table 3. Naturally, the (C-)NSP objective fails in the disambiguation task: the context representations learnt during (C-)NSP training are not useful for disambiguation, as the task at hand is substantially different. This is the main motivation to introduce the disambiguation task for rescoring. However we will see below that C-NSP almost reaches human accuracy at the task it is trained for, even when fine-tuned with small and noisy data such as our TNCS set.

On the other hand, disambiguation fine-tuning leads to substantial improvements. Results are, as expected, better when the training data is from the same dataset as the eva-
BERT attends the conversation: Improving Low-Resource Conversational ASR

| Training data/objective | Acc | TPR | TNR |
|-------------------------|-----|-----|-----|
| Base model NSP          | 47.95 | 92.27 | 3.63 |
| TNCS C-NSP              | 52.33 | 52.98 | 51.68 |
| TNCS Disambiguation ($y_o$) | 80.96 | 91.71 | 70.20 |
| NPSC Disambiguation ($y_o$) | 74.86 | 87.24 | 62.48 |
| + TNCS Disambiguation ($y_o$) | **82.17** | 89.94 | 74.39 |
| + TNCS Disambiguation ($y_{gt}$) | 79.05 | 81.25 | 75.00 |

Table 3: Text-only disambiguation accuracy on a balanced TNCS evaluation set with 2-utterance context. TPR/TNR are the true positive and negative rate, respectively. All values are given in percentage points.

Evaluation set (TNCS), but still we see a significant uplift when training only on NPSC with respect to the base model. Further training on conversational data from TNCS consistently improves results. As argued earlier, using the oracle transcript $y_o$ during training is better than using the ground truth $y_{gt}$, since the former situation is more similar to the task at hand.

These analyses strengthen the rescoring results presented in Section 6.3.1 and shed light on their interpretation. Despite the high accuracy of disambiguation-BERT on this task, during N-best rescoring the amount of negative samples is much larger. More importantly, BERT’s score is little significant when the AM and n-gram LM provide confident predictions, even if they are wrong. This results in a relatively low optimal value for $\gamma$ especially on the TNCS data. Conversely, higher $\gamma$ values lead to BERT overriding the AM and n-gram predictions in favour of likely incorrect predictions based purely on language and conversational context, ignoring what is actually said.

### 6.4.1 Conversational NSP and human performance

We now evaluate C-NSP fine-tuned BERT models on the task they were trained for, and analyze how the nature of the training data affects their performance on TNCS data. As shown in Table 4, the accuracy notably increases once BERT is exposed to the relevant conversational data, even with such a little amount of it.

| C-NSP model | Acc |
|-------------|-----|
| Base        | 53.13 |
| NPSC        | 59.38 |
| TNCS        | 68.75 |
| Human       | 70.31 |

Table 4: C-NSP accuracy (%) on 64 evaluation samples from a balanced C-NSP set from TNCS evaluation data. The human was not fine-tuned on this task nor exposed to the evaluation samples beforehand.
When we attempted the C-NSP task ourselves, we found it to be surprisingly tricky. Most samples required heavy use of domain knowledge, both in the form of common patterns in phone conversations and facts specific to telecommunications. This makes the results of C-NSP quite impressive, nearly reaching human accuracy on such highly unstructured conversational data, although the learnt representations help little in transcript disambiguation and rescoring.

7. Conclusions and future work

We presented novel, efficient techniques to rescore ASR candidates using a large pre-trained BERT model. These techniques require little fine-tuning and are particularly useful in the low-resource domain, since we only fine-tune with the few training transcriptions available for ASR.

In particular, disambiguation-BERT makes use of past conversational context and is trained to identify the best transcript among hypotheses. Applying it for rescoring leverages WER relative recoveries of over 37% when transcribing parliamentary debates. For spontaneous phone conversations improvements are more modest due to the quick turn-taking and topic changes, which also makes the model prefer less conversational context.

In the case of conversational NSP, the task is rather difficult for humans as well. Even so, a BERT model fine-tuned on C-NSP is only 2.2% worse than a human, although the language representations learnt are not as useful for the rescoring of ASR hypotheses.

Our analyses also reveal that the contribution of disambiguation-BERT in rescoring could be further optimized, since $\gamma$ is optimized given a whole evaluation set rather than for each utterance. That is, one could identify when the beam search scores from the AM and n-gram LM have low confidence, and set a higher weight on BERT’s contribution in those cases. Such tighter integration will be the subject of future work.

For long utterances, ASR hypotheses are very similar to each other, since early variations are more likely to be pruned during decoding. This makes the task or the LMs much harder and limits their potential. We did some preliminary investigations on the inclusion of a diversity bonus in the beam search inspired by the work of Vijayakumar et al. (2018) in image captioning, with the aim of forcing the beam search to explore different modes and increase the variability among candidates. This could potentially give higher-quality beams and enable, among other benefits, rescoring improvements independently of the rescoring technique used. In future work we will report on our investigations and the effect of beam diversity in ASR.

In this work we have exploited the large gap between the top-ranked transcription and the oracle transcript in low-resource conversational ASR. Our findings contribute to reducing that gap without modifying the acoustic model, although modifications of that component will also be investigated in future work. Of particular interest in our low-resource domain are end-to-end architectures that make better use of the information contained in the (limited) data available for ASR training, in addition to exploiting the language representations provided by large pre-trained language models.

16. https://gitlab.com/sburud/master/-/tree/master/speechlm/ctc-decode
Acknowledgments

We are thankful for feedback and enlightening discussions with Knut Kvale, Ole Jakob Mengshoel, Massimiliano Ruocco, Giampiero Salvi, Marco Siniscalchi and Torbjørn Svendsen. We are very grateful to Knut Kvale for his valuable collaboration on transcribing Telenor conversational data. We also wish to thank Petr Taborsky for collaboration during the initial stages of this work on training the acoustic models, and to Weiqing Zhang for technical support related to the GPU servers.

This work was partially carried out as part of SB’s master thesis (Burud, 2021). PO has been partially supported by the Norwegian Research Council through the IKTPLUSS grant for the SCRIBE project$^{17}$ (KSP21PD).

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2623–2631. Association for Computing Machinery, 2019. ISBN 978-1-4503-6201-6. doi: 10.1145/3292500.3330701. URL https://doi.org/10.1145/3292500.3330701.

Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse H. Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Y. Hannun, Billy Jun, Patrick LeGresley, Libby Lin, Sharan Narang, Andrew Y. Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang, Bo Xiao, Dani Yogatama, Jun Zhan, and Zhenyao Zhu. Deep speech 2: End-to-end speech recognition in English and Mandarin. CoRR, 1512.02595, 2015. URL http://arxiv.org/abs/1512.02595.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. CoRR, 2005.14165, 2020. URL https://arxiv.org/abs/2005.14165.

Simen Burud. Conversational language models for low-resource speech recognition, 2021. URL https://ntnuopen.ntnu.no/ntnu-xmlii/handle/11250/2827059. MSc thesis, Department of Computer Science, NTNU.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4960–4964, 2016. doi: 10.1109/ICASSP.2016.7472621.

$^{17}$ https://scribe-project.github.io/
Shih-Hsuan Chiu and Berlin Chen. Innovative Bert-based Reranking Language Models for Speech Recognition. *CoRR*, 2104.04950, 2021. URL https://arxiv.org/abs/2104.04950.

Shih-Hsuan Chiu, Tien-Hong Lo, and Berlin Chen. Cross-sentence neural language models for conversational speech recognition. *CoRR*, 2106.06922, 2021. URL https://arxiv.org/abs/2106.06922.

Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, KyungHyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. *CoRR*, 1506.07503, 2015. URL http://arxiv.org/abs/1506.07503.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, 1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Lin Hao Dong, Shuang Xu, and Bo Xu. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888, 2018. doi: 10.1109/ICASSP.2018.8462506.

Alex Graves. Sequence transduction with recurrent neural networks. *CoRR*, 1211.3711, 2012. URL http://arxiv.org/abs/1211.3711.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML ’06, page 369–376, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143891. URL https://doi.org/10.1145/1143844.1143891.

Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. On using monolingual corpora in neural machine translation. *CoRR*, 1503.03535, 2015. URL http://arxiv.org/abs/1503.03535.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696. Association for Computational Linguistics, 2013. URL https://www.aclweb.org/anthology/P13-2121.

Hongzhao Huang and Fuchun Peng. An empirical study of efficient ASR rescoring with transformers. *CoRR*, 1910.11450, 2019. URL http://arxiv.org/abs/1910.11450.

Gang Ji and Jeff Bilmes. Multi-speaker language modeling. In *Proceedings of HLT-NAACL 2004: Short Papers*, page 133–136, USA, 2004. Association for Computational Linguistics. ISBN 1932432248.

Suyoun Kim and Florian Metze. Dialog-context aware end-to-end speech recognition. *CoRR*, 1808.02171, 2018. URL http://arxiv.org/abs/1808.02171.
Suyoun Kim, Siddharth Dalmia, and Florian Metze. Cross-attention end-to-end ASR for two-party conversations. In *INTERSPEECH*, pages 4380–4384, 2019. URL https://doi.org/10.21437/Interspeech.2019-3173.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, 1909.11942, 2019. URL http://arxiv.org/abs/1909.11942.

Siddique Latif, Rajib Rana, Sara Khalifa, Raja Jurdak, Junaid Qadir, and Björn W. Schuller. Deep representation learning in speech processing: Challenges, recent advances, and future trends. *CoRR*, 2001.00378, 2020. URL http://arxiv.org/abs/2001.00378.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, 1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Atsunori Ogawa, Marc Delcroix, Shigeki Karita, and Tomohiro Nakatani. Rescoring N-best speech recognition list based on one-on-one hypothesis comparison using encoder-classifier model. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6099–6103, 2018. doi: 10.1109/ICASSP.2018.8461405.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *CoRR*, 1802.05365, 2018. URL http://arxiv.org/abs/1802.05365.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Ashish Shenoy, Sravan Bodapati, and Katrin Kirchhoff. ASR adaptation for e-commerce chatbots using cross-utterance context and multi-task language modeling. *Proceedings of The 4th Workshop on e-Commerce and NLP*, 2021a. doi: 10.18653/v1/2021.ecnlp-1.3. URL http://dx.doi.org/10.18653/v1/2021.ecnlp-1.3.

Ashish Shenoy, Sravan Bodapati, Monica Sunkara, Srikanth Ronanki, and Katrin Kirchhoff. "What’s the context?" : Long context NLM adaptation for ASR rescoring in conversational agents. *CoRR*, 2104.11070, 2021b. URL https://arxiv.org/abs/2104.11070.

Joonbo Shin, Yoonhyung Lee, and Kyomin Jung. Effective sentence scoring method using BERT for speech recognition. pages 1081–1093. PMLR, 2019. URL http://proceedings.mlr.press/v101/shin19a.html.

Mittul Singh, Youssef Oualil, and Dietrich Klakow. Approximated and Domain-Adapted LSTM Language Models for First-Pass Decoding in Speech Recognition. In *Proc. Interspeech 2017*, pages 2720–2724, 2017. doi: 10.21437/Interspeech.2017-147.

Martin Skjekkeland. *Dialektlandet*. Portal, Kristiansand, 2010. ISBN 9788292712313.
Gabriel Synnaeve, Qiantong Xu, Jacob Kahn, Edouard Grave, Tatiana Likhomanenko, Vineel Pratap, Anuroop Sriram, Vitaliy Liptchinsky, and Ronan Collobert. End-to-end ASR: from supervised to semi-supervised learning with modern architectures. *CoRR*, 1911.08460, 2019. URL http://arxiv.org/abs/1911.08460.

Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search for improved description of complex scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. URL https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17329.

Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R. Hershey, and Tomoki Hayashi. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal on Selected Topics in Signal Processing*, 11(8):1240–1253, December 2017. ISSN 1932-4553. doi: 10.1109/JSTSP.2017.2763455.

Jason D. Williams. Exploiting the ASR n-best by tracking multiple dialog state hypotheses. In *Proc. Interspeech 2008*, pages 191–194, 2008. doi: 10.21437/Interspeech.2008-61.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics, October 2020. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Wayne Xiong, Lingfeng Wu, Jun Zhang, and Andreas Stolcke. Session-level language modeling for conversational speech. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2764–2768, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1296. URL https://aclanthology.org/D18-1296.