# Robust Prediction of Punctuation and Truecasing for Medical ASR

**Monica Sunkara    Srikanth Ronanki    Kalpit Dixit    Sravan Bodapati    Katrin Kirchhoff**
Amazon AWS AI, USA
{sunkaral, ronanks}@amazon.com

## Abstract

Automatic speech recognition (ASR) systems in the medical domain that focus on transcribing clinical dictations and doctor-patient conversations often pose many challenges due to the complexity of the domain. ASR output typically undergoes automatic punctuation to enable users to speak naturally, without having to vocalise awkward and explicit punctuation commands, such as "period", "add comma" or "exclamation point", while truecasing enhances user readability and improves the performance of downstream NLP tasks. This paper proposes a conditional joint modeling framework for prediction of punctuation and truecasing using pretrained masked language models such as BERT, BioBERT and RoBERTa. We also present techniques for domain and task specific adaptation by fine-tuning masked language models with medical domain data. Finally, we improve the robustness of the model against common errors made in ASR by performing data augmentation. Experiments performed on dictation and conversational style corpora show that our proposed model achieves ∼5% absolute improvement on ground truth text and ∼10% improvement on ASR outputs over baseline models under F1 metric.

## 1 Introduction

Medical ASR systems automatically transcribe medical speech found in a variety of use cases like physician-dictated notes (Edwards et al., 2017), telemedicine and even doctor-patient conversations (Chiu et al., 2017), without any human intervention. These systems ease the burden of long hours of administrative work and also promote better engagement with patients. However, the generated ASR outputs are typically devoid of punctuation and truecasing thereby making it difficult to comprehend. Furthermore, their recovery improves the accuracy of subsequent natural language understanding algorithms (Peitz et al., 2011a; Makhoul et al., 2005) to identify information such as patient diagnosis, treatments, dosages, symptoms and signs. Typically, clinicians explicitly dictate the punctuation commands like "period", "add comma" etc., and a postprocessing component takes care of punctuation restoration. This process is usually error-prone as the clinicians may struggle with appropriate punctuation insertion during dictation. Moreover, doctor-patient conversations lack explicit vocalization of punctuation marks motivating the need for automatic prediction of punctuation and truecasing. In this work, we aim to solve the problem of automatic punctuation and truecasing restoration to medical ASR system text outputs.

Most recent approaches to punctuation and truecasing restoration problem rely on deep learning (Nguyen et al., 2019a; Salloum et al., 2017). Although it is a well explored problem in the literature, most of these improvements do not directly translate to great real world performance in all settings. For example, unlike general text, it is a harder problem to solve when applied to the medical domain for various reasons and we illustrate each of them:

- **Large vocabulary:** ASR systems in the medical domain have a large set of domain-specific vocabulary and several abbreviations. Owing to the domain specific data set and the open vocabulary in LVCSR (large-vocabulary continuous speech recognition) outputs, we often run into OOV (out of vocabulary) or rare word problems. Furthermore, a large vocabulary set leads to data sparsity issues. We address both these problems by using subword models. Subwords have been shown to work well in open-vocabulary speech recognition and several NLP tasks (Sennrich et al., 2015;

Bodapati et al., 2019). We compare word and subword models across different architectures and show that subword models consistently outperform the former.

- **Data scarcity:** Data scarcity is one of the major bottlenecks in supervised learning. When it comes to the medical domain, obtaining data is not as straight-forward as some of the other domains where abundance of text is available. On the other hand, obtaining large amounts of data is a tedious and costly process; procuring and maintaining it could be a challenge owing to the strict privacy laws. We overcome the data scarcity problem, by using pretrained masked language models like BERT (Devlin et al., 2018) and its successors (Liu et al., 2019; Yang et al., 2019) which have successfully been shown to produce state-of-the-art results when finetuned for several downstream tasks like question answering and language inference. We approach the prediction task as a sequence labeling problem and jointly learn punctuation and truecasing. We show that finetuning a pretrained model with a very small medical dataset (∼500k words) has ∼5% absolute performance improvement in terms of F1 compared to a model trained from scratch. We further boost the performance by first finetuning the masked language model to the medical speech domain and then to the downstream task.

- **ASR Robustness:** Models trained on ground truth data are not exposed to typical errors in speech recognition and perform poorly when evaluated on ASR outputs. Our objective is to make the punctuation prediction and truecasing more robust to speech recognition errors and establish a mechanism to test the performance of the model quantitatively. To address this issue, we propose a data augmentation based approach using n-best lists from ASR.

The contributions of this work are:

- A general post-processing framework for conditional joint labeling of punctuation and truecasing for medical ASR (clinical dictation and conversations).

- An analysis comparing different embeddings that are suitable for the medical domain. An

in-depth analysis of the effectiveness of using pretrained masked language models like BERT and its successors to address the data scarcity problem.

- Techniques for effective domain and task adaptation using Masked Language Model (*MLM*) finetuning of BERT on medical domain data to boost the downstream task performance.

- Method for enhancing robustness of the models via data augmentation with n-best lists (from ASR output) to the ground truth during training to improve performance on ASR hypothesis at inference time.

The rest of this paper is organized as follows. Section 2 presents related work on punctuation and truecasing restoration. Section 3 introduces the model architecture used in this paper and describes various techniques for improving accuracy and robustness. The experimental evaluation and results are discussed in Section 4 and finally, Section 5 presents the conclusions.

## 2 Related work

Several researchers have proposed a number of methodologies such as the use of probabilistic machine learning models, neural network models, and the acoustic fusion approaches for punctuation prediction. We review related work in these areas below.

### 2.1 Earlier methods

In earlier efforts, punctuation prediction has been approached by using finite state or hidden Markov models (Gotoh and Renals, 2000; Christensen et al., 2001a). Several other approaches addressed it as a language modeling problem by predicting the most probable sequence of words with punctuation marks inserted (Stolcke et al., 1998; Beeferman et al., 1998; Gravano et al., 2009). Some others used conditional random fields (CRFs) (Lu and Ng, 2010; Ueffing et al., 2013) and maximum entropy using n-grams (Huang and Zweig, 2002). The rise of stronger machine learning techniques such as deep and/or recurrent neural networks replaced these conventional models.

### 2.2 Using acoustic information

Some methods used only acoustic information such as speech rate, intonation, pause duration etc.,

(Christensen et al., 2001b; Levy et al., 2012). While pauses influence in the prediction of Comma, intonation helps in disambiguation between punctuation marks like period and exclamation. Although this seemed to work, the most effective approach is to combine acoustic information with lexical information at word level using force-aligned duration (Klejch et al., 2017). In this work, we only considered lexical input and a pretrained lexical encoder for prediction of punctuation and truecasing. The use of pretrained acoustic encoder and fusion with lexical outputs are possible extensions in future work.

## 2.3 Neural approaches

Neural approaches for punctuation and truecasing can be classified into two broad categories: sequence labeling based models and MT-based seq2seq models. These approaches have proven to be quite effective in capturing the contextual information and achieved huge success. While some approaches considered only punctuation prediction, some others jointly modeled punctuation and truecasing.

One set of approaches treated punctuation as a machine translation problem and used phrase based statistical machine translation systems to output punctuated and true cased text (Peitz et al., 2011b; Cho et al., 2012; Driesen et al., 2014). Inspired by recent end-to-end approaches, (Yi and Tao, 2019) proposed the use of self-attention based transformer model to predict punctuation marks as output sequence for given word sequences. Most recently, (Nguyen et al., 2019b) proposed joint modeling of punctuation and truecasing by generating words with punctuation marks as part of the decoding. Although seq2seq based approaches have shown a strong performance, they are intensive, demanding and are not suitable for production deployment at large scale.

For sequence labeling problem, each word in the input is tagged with a punctuation. If there is no punctuation associated with a word, a blank label is used and is often referred as "no punc". (Cho et al., 2015) used a combination of neural networks and CRFs for joint prediction of punctuation and disfluencies. With growing popularity in deep recurrent neural networks, LSTMs and BLSTMs with attention mechanism were introduced for punctuation restoration (Tilk and Alumäe, 2015, 2016). Later, (Pahuja et al., 2017) proposed joint training of punc-

tuation and truecasing using BLSTM models. This work addressed joint learning as two correlated tasks, and predicted punctuation and truecasing as two independent outputs. Our proposed approach is similar to this work, but we rather condition truecasing prediction on punctuation output; this is discussed in detail in Section 3.

Punctuation and casing restoration for speech/ASR outputs in the medical domain has not been explored extensively. Recently, (Salloum et al., 2017) proposed a sequence labeling model using bi-directional RNNs with an attention mechanism and late fusion for punctuation restoration to clinical dictation. To our knowledge, there has not been any work on medical conversations, and we aim to bridge the gap here with latest advances in NLP with large-scale pretrained language models.

## 3 Modeling : Conditional Joint labeling of Punctuation + Casing

We propose a postprocessing framework for conditional and joint learning of punctuation and truecasing prediction. Consider an input utterance $x_{1:T} = \{x_1, x_2, ..., x_T\}$, of length $T$ and consisting of words $x_i$. The first step in our modeling process involves punctuation prediction as a sequence tagging task. Once the model predicts a probability distribution over punctuation, this along with the input utterance is fed in as input for predicting the case of a word $x_i$. We consider the punctuation to be independent of casing and a conditional dependence of the truecase of a word on punctuation given the learned input representations. Our plausible reasoning follows from this example sentence – "She took dance classes. She had no natural grace or sense of rhythm.". The word after the period is capitalized, which implies that punctuation information can help in better prediction of casing. A pair of punctuation and truecasing is assigned per word:

$$\mathbf{Pr}(\mathbf{p_{1:T}}, \mathbf{c_{1:T}} | \mathbf{x_{1:T}}) = \mathbf{Pr}(\mathbf{p_{1:T}} | \mathbf{x_{1:T}}) \mathbf{Pr}(\mathbf{c_{1:T}} | \mathbf{p_{1:T}}, \mathbf{x_{1:T}}) \quad (1)$$

where $c_i \in C$, a fixed set of casing labels {Lower_Case, Upper_Case, All_Caps, Mixed_Case}, and $p_i \in P$, a fixed set of punctuation labels {Comma, Period, Question_Mark, No_Punct}.

### 3.1 Pretrained lexical encoder

We propose to use a pretrained model like BERT, trained on a large text corpus, as a lexical encoder
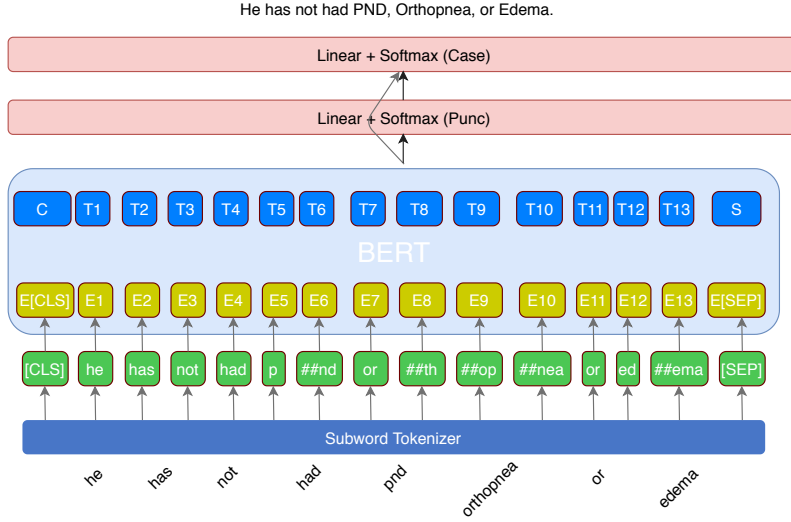
He has not had PND, Orthopnea, or Edema.



Figure 1: Pre-trained BERT encoder for prediction of punctuation and truecasing.

for learning an effective representation of the input utterance. Figure 1 illustrates our proposed model architecture.

**Subword embeddings** Given a sequence of input vectors $(x_1, x_2, ..., x_T)$, where $x_i$ represents a word $w_i$, we extract the subword embeddings $(s_1, s_2, ..., s_n)$ using a wordpiece tokenizer (Schuster and Nakajima, 2012). Using subwords is especially effective in medical domain, as it contains more compound words with common subwords. For example consider the six words {hypotension, hypertension, hypoactive, hyperactive, active, tension } with four common subwords {hyper, hypo, active, tension}. In Section 4.2, we provide a comparative analysis of word and subword models across different architectures on medical data.

**BERT encoder** We provide subword embeddings $(s_1, s_2, ..., s_n)$ as input to the BERT encoder, which outputs a sequence of hidden states: H = $(h1, ..., h_n$) at its final layer. The pretrained BERT base encoder consists of 12 transformer encoder self-attention layers. For this task, we truncate the BERT encoder and fine-tune only the first six layers to reduce the model complexity. Although a deep encoder might enable us to learn a long memory context dependent representation of the input utterance, the performance gain is very minimal compared to the increased latency[1].

For punctuation, we input the last layer representations of truncated BERT encoder $h_1, h_2, ..., h_n$ to a linear layer with softmax activation to classify over the punctuation labels generating $(p_1, p_2, ..., p_n)$ as outputs. For casing, we concatenate the softmax probabilities of punctuation output with BERT encoder's outputs and feed to a linear layer with softmax activation generating case labels $(c_1, c_2, ..., c_n)$ for the sequence. The softmax output for punctuation ($\hat{p}_i$) and truecasing ($\hat{c}_i$) is as follows:

$$\hat{p}_i = softmax(W^k h_i + b^k) \qquad (2)$$

$$\hat{c}_i = softmax(W^l(\hat{p}_i \oplus h_i) + b^l) \qquad (3)$$

where $W^k, b^k$ denote weights and bias of punctuation linear output layer and $W^l, b^l$ denote weights and bias of truecasing linear output layer.

**Joint learning objective:** We model our learning objective to maximize the joint probability $\mathbf{Pr}(\mathbf{p_{1:T}}, \mathbf{c_{1:T}}|\mathbf{x_{1:T}})$. The model is finetuned end-to-end to minimize the cross-entropy loss between the assigned distribution and the training data. The parameters of BERT encoder are shared across punctuation and casing prediction tasks and are jointly trained. We compute the losses ($L^p, L^c$) for each task using cross entropy loss function. The final loss L to be optimized is a weighted average of the task-specific losses:

$$L = \alpha L^p + L^c \qquad (4)$$

where $\alpha$ is a fixed weight optimized for best predictions across both the tasks. In our experiments, we explored $\alpha$ values in the range of (0.2-2) and found 0.6 to be the optimal value.

---

[1]We experimentally found that 12-layer BERT base model gives ∼1% improvement over 6-layer BERT base model whereas the inference and training times were double for the former.

56

## 3.2 Finetuning using Masked Language Model with Medical domain data

BERT and its successors have shown great performance on downstream NLP tasks. But just like any other model, these Language Models are biased by their training data. In particular, they are typically trained on data that is easily available in large quantities on the internet e.g. Wikipedia, Common-Crawl etc. Our domain, Medical ASR Text, is not "common" and is very under-represented in the training data for these Language Models. One way to correct this situation is to perform a few steps of unsupervised Masked Language Model finetuning on the BERT models before performing cross-entropy training using the labeled task data (Han and Eisenstein, 2019).

**Domain adaptation** We finetune the pretrained BERT model for MLM (Masked LM) objective on medical domain data. 15% of input tokens are masked randomly before feeding into the BERT model as proposed by (Devlin et al., 2018). The main goal is to adapt and learn better representations of speech data. The domain adapted model can be further finetuned with an additional layer to a downstream task like punctuation and casing prediction.

**Domain+Task adaptation** Building on the previous technique, we attempt to finetune the pretrained model for task adaptation in combination with domain adaptation. In this technique, instead of randomly masking 15% of the input tokens, we do selective masking i.e. 50% of the masked tokens would be random and the other 50% would be punctuation marks (["."; ","; "?"] in our case). Therefore, the finetuned model would not only adapt to speech domain, but would also effectively learn the placement of punctuation marks in a text based on the context.

## 3.3 Robustness to ASR errors

Models trained on ground truth text inputs may not perform well when tested with ASR output, especially when the system introduces grammatical errors. To make models more robust against ASR errors, we perform data augmentation with ASR outputs for training. For punctuation restoration, we use edit distance measure to align ASR hypothesis with ground truth punctuated text. Before computing alignment, we strip all punctuation from ground truth and lowercase the text. This helps us find the best alignment between ASR hypothesis and ground truth text. Once the alignment is found, we restore the punctuation from each word in ground truth text to hypothesis. If there are words that are punctuated in ground truth but got deleted in ASR hypothesis, we restore the punctuation to previous word. For truecasing, we try to match the reference word with hypothesis word from aligned sequences with a window size of 5, two words to the left and two words to the right of current word and restore truecasing only in the cases where reference word is found. We performed experiments with data augmentation using 1-best hypothesis and n-best lists as additional training data and the results are reported in Section 4.4.

## 4 Experiments and results

### 4.1 Data

We evaluate our proposed framework and models on a subset of two internal medical datasets: dictation and conversational. The dictation corpus contains 3.7M words and the conversational corpus contains 51M words. The medical data comes with special tags masking personal identifiable and patient health information. We also use a general domain Wikipedia dataset for comparative analysis with Medical domain data. This data is a subset of the publicly available release of Wiki dataset (Sproat and Jaitly, 2016). The corpus contains 35M words and relatively shorter sentences ranging from 8 to 200 words in length. 90% of the data from each corpus is used for training, 5% for fine-tuning and remaining 5% is held-out for testing.

For robustness experiments presented in Section 4.4, we used data from the dictation corpus consisting of 2265 text files and corresponding audio files with an average duration of ∼15 minutes. The total length of the corpus is 550 hours. For augmentation with ground-truth transcription, we transcribed audio files using a speech recognition system. Restoration of punctuation and truecasing to transcribed text can be erroneous as the word error rate(*WER*) goes up. We therefore discarded the transcribed text of those audio files whose WER is more than 25%. We sorted the remaining transcriptions based on WER to make further splits: hypothesis from top 50 files with best WER is set as test data, and the next 50 files were chosen as development and rest of the transcribed text was used for training. The partition was done this way to minimize the number of errors that may occur

| | | Punctuation | | | Truecasing | | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | Token | No Punc | Full stop | Comma | LC | UC | CA | MC |
| CNN-Highway | word | 0.97 | 0.81 | 0.71 | 0.98 | 0.84 | 0.95 | 0.99 |
| | subword | 0.98 | 0.83 | 0.70 | 0.99 | 0.87 | 0.95 | 0.99 |
| 3-LSTM | word | 0.97 | 0.82 | 0.73 | 0.98 | 0.84 | 0.96 | 0.98 |
| | subword | 0.98 | 0.84 | 0.75 | 0.99 | 0.87 | 0.97 | 0.99 |
| 3-BLSTM | word | 0.98 | 0.86 | 0.75 | 0.99 | 0.88 | 0.97 | 0.98 |
| | subword | 0.99 | 0.87 | 0.76 | 0.99 | 0.90 | 0.97 | 1.0 |
| Transformer encoder | word | 0.97 | 0.84 | 0.7 | 0.98 | 0.86 | 0.97 | 0.98 |
| | subword | 0.98 | 0.85 | 0.72 | 0.99 | 0.87 | 0.97 | 0.99 |

Table 1: Dictation corpus: Comparison of F1 scores for punctuation and truecasing across different model architectures using word and subword tokens (LC: lower case; UC: Upper case; CA: CAPS All; MC: Mixed Case).

| | | Punctuation | | | | Truecasing | | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | Token | No Punc | Full stop | Comma | QM | LC | UC | CA | MC |
| CNN-Highway | word | 0.96 | 0.72 | 0.64 | 0.60 | 0.96 | 0.78 | 0.99 | 0.91 |
| | subword | 0.97 | 0.74 | 0.65 | 0.61 | 0.97 | 0.80 | 0.98 | 0.99 |
| 3-LSTM | word | 0.96 | 0.74 | 0.64 | 0.65 | 0.96 | 0.79 | 0.99 | 0.95 |
| | subword | 0.97 | 0.75 | 0.65 | 0.66 | 0.97 | 0.79 | 0.97 | 1.0 |
| 3-BLSTM | word | 0.97 | 0.77 | 0.68 | 0.68 | 0.97 | 0.82 | 0.99 | 0.95 |
| | subword | 0.98 | 0.79 | 0.68 | 0.69 | 0.97 | 0.83 | 0.99 | 1.0 |
| Transformer encoder | word | 0.97 | 0.77 | 0.68 | 0.68 | 0.97 | 0.83 | 0.99 | 0.92 |
| | subword | 0.98 | 0.79 | 0.69 | 0.69 | 0.98 | 0.83 | 0.99 | 1.0 |

Table 2: Conversational corpus: Comparison of F1 scores for punctuation and truecasing across different model architectures using word and subword tokens (QM: Question Mark; LC: lower case; UC: Upper case; CA: CAPS All; MC: Mixed Case).

during restoration.

**Preprocessing long-speech transcriptions** Conversational style speech has long-speech transcripts, in which the context is spread across multiple segments. we use an overlapped chunking and merging component to pre and post process the data. We use a sliding window approach (Nguyen et al., 2019a) to split long ASR outputs into chunks of 200 words each with an overlapping window of 50 words each to the left and right. The overlap helps in preserving the context for all the words after splitting and ensures accurate prediction of punctuation and case corresponding to each word.

## 4.2 Large Vocabulary: Word vs Subword models

For a fair comparison with BERT, we evaluate various recurrent and non-recurrent architectures with both word and subword embeddings. The two recurrent models include a 3 layer uni-directional LSTM (*3-LSTM*) and a 3 layer Bi-directional LSTM (*3-BLSTM*). One of the non recurrent encoders, implements a CNN-Highway architecture based on the work proposed by (Kim et al., 2016),

whereas the other one implements a transformer encoder based model (Vaswani et al., 2017). We train all four models on medical data from dictation and conversation corpus with weights initialized randomly. The vocabulary for word models is derived by considering all the unique words from training corpus, with additional tokens for unknown and padding. This yielded a vocabulary size of 30k for dictation and 64k for conversational corpus. Subwords are extracted using a wordpiece model (Schuster and Nakajima, 2012) and its inventory is less than half that of word model for conversation. Tables 1 and 2 summarize our results on dictation and conversation datasets respectively. We observe that subword models consistently performed same or better than word models. On punctuation task, for Full stop and Comma, we notice an absolute ~1-2% improvement respectively on dictation set. Similarly, on the conversation dataset, we notice an absolute ~1-2% improvement on Full stop, Comma and Question Mark. For the casing task, we notice that word and subword models performed equally well except in dictation dataset where we see an absolute ~3% improvement for Upper_Case.

| | | Punctuation | | | Truecasing | | | |
|---|---|---|---|---|---|---|---|---|
| Model | Dataset | No Punc | Full stop | Comma | LC | UC | CA | MC |
| 3-BLSTM | Wiki | 0.95 | 0.17 | 0.27 | 0.95 | 0.31 | 0.55 | 0.19 |
| BERT | Wiki | 0.96 | 0.2 | 0.39 | 0.95 | 0.36 | 0.65 | 0.2 |
| 3-BLSTM | Medical | 0.99 | 0.87 | 0.76 | 0.99 | 0.9 | 0.97 | 1.0 |
| BERT | Medical | 0.99 | 0.9 | 0.81 | 0.99 | 0.93 | 0.99 | 1.0 |
| FT-BERT | Medical | 0.99 | 0.92 | 0.82 | 0.99 | 0.93 | 0.99 | 1.0 |
| PM-BERT | Medical | 0.99 | 0.93 | 0.82 | 0.99 | 0.94 | 0.99 | 1.0 |
| Bio-BERT | Medical | 0.99 | 0.92 | 0.82 | 0.99 | 0.93 | 0.99 | 1.0 |
| RoBERTa | Medical | 0.99 | 0.92 | 0.81 | 0.99 | 0.94 | 0.99 | 1.0 |

Table 3: Comparison of F1 scores for punctuation and truecasing using BERT and BLSTM when trained on Wiki data and Medical dictation data (FT-BERT: Finetuned BERT for domain adapation, PM-BERT: Finetuned BERT by punctuation masking for domain and task adapation).

| | | Punctuation | | | | Truecasing | | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | Dataset | No Punc | Full stop | Comma | QM | LC | UC | CA | MC |
| 3-BLSTM | Wiki | 0.89 | 0.001 | 0.25 | 0.002 | 0.93 | 0.13 | 0.9 | 0.95 |
| BERT | Wiki | 0.93 | 0.004 | 0.4 | 0.007 | 0.93 | 0.4 | 0.95 | 0.95 |
| 3-BLSTM | Medical | 0.98 | 0.79 | 0.68 | 0.69 | 0.97 | 0.83 | 0.99 | 1.0 |
| BERT | Medical | 0.98 | 0.8 | 0.71 | 0.72 | 0.98 | 0.85 | 0.99 | 1.0 |
| FT-BERT | Medical | 0.98 | 0.81 | 0.72 | 0.73 | 0.98 | 0.85 | 0.99 | 1.0 |
| PM-BERT | Medical | 0.98 | 0.82 | 0.72 | 0.74 | 0.98 | 0.86 | 0.99 | 1.0 |
| Bio-BERT | Medical | 0.98 | 0.81 | 0.71 | 0.72 | 0.98 | 0.85 | 0.99 | 1.0 |
| RoBERTa | Medical | 0.98 | 0.82 | 0.73 | 0.74 | 0.98 | 0.86 | 0.99 | 1.0 |

Table 4: Comparison of F1 scores for punctuation and truecasing using BERT and BLSTM when trained on Wiki data and Medical conversation data (FT-BERT: Finetuned BERT for domain adapation, PM-BERT: Finetuned BERT by punctuation masking for domain and task adapation).

We hypothesize that medical vocabulary contains a large set of compound words, which a subword based model works effectively over word model. Upon examining few utterances, we noticed that subword models can learn effective representations of these compound medical words by tokenizing them into subwords. On the other hand, word models often run into rare word or OOV issues.

### 4.3 Pretrained language models

**Significance of in-domain data** For analyzing the importance of in-domain data, we train a baseline BLSTM model and a pretrained BERT model on Wiki and Medical data from both dictation and conversational corpus and tested the models on Medical held-out data. The first four rows of Tables 3 and 4 summarize the results. The models trained on Wiki data performed very poorly when compared to models trained on Medical data from either dictation or conversation corpus. Although dictation corpus (3.7M words) is relatively smaller than Wiki corpus (35M words), the difference in accuracy is significantly higher across both

models. Imbalanced classes like Full stop, Comma, Question_Mark were most affected. Another interesting observation is that the models trained on Medical data performed better on Full stop compared to Comma; whereas general domain models performed better on Comma compared to Full stop. The degradation in general models might be due to Wiki sentences being short and ending with a Full stop unlike lengthy medical transcripts. Also, the F1 scores are lower on conversation data across both the tasks, indicating the complexity involved in modeling conversational data due to their highly unstructured format. Overall, the pretrained BERT model consistently outperformed baseline BLSTM model on both dictation and conversation data. This motivated us to focus on adapting the pretrained models for this task.

**Finetuning Masked LM** We have run two levels of fine-tuning as explained in Section 3.2. First, we finetuned BERT with Medical domain data using random masking (*FT-BERT*) and for task adaptation, we performed fine-tuning with

| Model | n-best | Punctuation | | | | Truecasing | | | |
| | | No Punc | Full stop | Comma | QM | LC | UC | CA | MC |
|---|---|---|---|---|---|---|---|---|---|
| BERT-GT | - | 0.97 | 0.58 | 0.45 | 0.0 | 0.98 | 0.60 | 0.78 | 0.90 |
| BERT-ASR | 1-best | 0.97 | 0.66 | 0.56 | 0.54 | 0.99 | 0.72 | 0.86 | 1.0 |
| | 3-best | 0.98 | 0.67 | 0.57 | 0.42 | 0.98 | 0.69 | 0.79 | 0.84 |
| | 5-best | 0.97 | 0.61 | 0.5 | 0.35 | 0.98 | 0.65 | 0.79 | 0.83 |

Table 5: Comparison of F1 scores for punctuation and truecasing with ground truth and ASR augmented data.
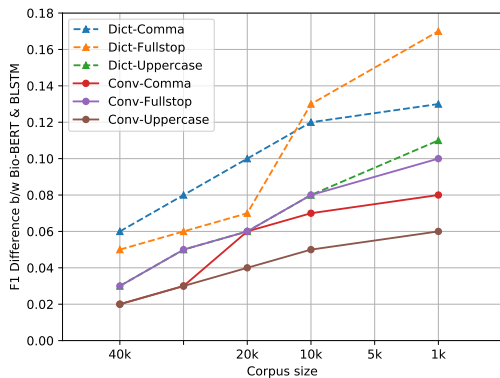


Figure 2: Difference in F1 scores between Bio-BERT and BLSTM for varying data sizes.

punctuation based masking (*PM-BERT*). For both experiments, we used the same data as we have used for finetuning the downstream task. From the results presented in Table 3 and 4, we infer that finetuning boosts the performance of punctuation and truecasing (an absolute improvement of ~1-2%). From both the datasets, it is clear that task specific masking helps better than simple random masking. For dictation dataset, Full stop improved by an absolute 3% by performing punctuation specific masking, suggesting that finetuning MLM can give higher benefits when the amount of data is low.

**Variants of BERT** We compare three pretrained models namely, BERT and its successor RoBERTa (Liu et al., 2019) and Bio-BERT (Lee et al., 2020) which was trained on large scale Biomedical corpora. The results are summarized in last two rows of Table 3 and 4. First, we observe that both Bio-BERT and RoBERTa outperformed the initial BERT model and has shown an absolute ~3-5% improvement over the baseline 3-BSLTM. To further validate this, we extended our experiments to understand how the performance of our best model(Bio-BERT) varies across different training dataset sizes compared to the baseline. From Figure 2, we observe that the difference increases significantly as we move towards smaller datasets. For the smallest data set size of 500k words (1k transcripts), there

is an absolute improvement of 6-17% over the baseline in accuracy in terms of F1. This shows that pretraining on a large dataset helps to overcome data scarcity issue effectively.

### 4.4 Robustness

For testing robustness, we performed experiments with augmentation of ASR data from n-best lists (*BERT-ASR*). We considered top-1, top-3 and top-5 hypotheses for n-best lists augmentation with ground truth text and the results are presented in Table 5. Additionally, the best BERT model trained using only ground truth text inputs (*BERT-GT*) from Table 3 is also evaluated on ASR outputs. To compute F1 scores on held-out test set, we first aligned the ASR hypothesis with ground truth data and restored the punctuation and truecasing as described in Section 3.3. From the results presented in Table 5, we infer that adding ASR hypothesis to the training data helped improve the performance of both punctuation and truecasing. In punctuation, both Full stop and Comma have seen an absolute 10% improvement in F1 score. Although the number of question marks is less in test data, the augmented systems performed really well compared to the system trained purely on ground truth text. However, we found that using n-best lists with $n > 1$ did not help much compared to the 1-best list. This may be due to sub-optimal restoration of punctuation and truecasing as the WER with n-best lists is likely to go up as $n$ increases.

### 5 Conclusion

In this paper, we have presented a framework for conditional joint modeling of punctuation and truecasing in medical transcriptions using pretrained language models such as BERT. We also demonstrated the benefit from MLM objective finetuning of the pretrained model with task specific masking. We further improved the robustness of punctuation and truecasing on ASR outputs by data augmentation during training. Experiments performed on both dictation and conversation corpora show the

effectiveness of the proposed approach. Future work includes the use of either pretrained acoustic features or pretrained acoustic encoder to perform fusion with pretrained linguistic encoder to further boost the performance of punctuation.

# References

Doug Beeferman, Adam Berger, and John Lafferty. 1998. Cyberpunc: A lightweight punctuation annotation system for speech. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, volume 2, pages 689–692. IEEE.

Sravan Bodapati, Spandana Gella, Kasturi Bhattacharjee, and Yaser Al-Onaizan. 2019. Neural word decomposition models for abusive language detection. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 135–145, Florence, Italy. Association for Computational Linguistics.

Chung-Cheng Chiu, Anshuman Tripathi, Katherine Chou, Chris Co, Navdeep Jaitly, Diana Jaunzeikare, Anjuli Kannan, Patrick Nguyen, Hasim Sak, Ananth Sankar, et al. 2017. Speech recognition for medical conversations. *arXiv preprint arXiv:1711.07274*.

Eunah Cho, Kevin Kilgour, Jan Niehues, and Alex Waibel. 2015. Combination of nn and crf models for joint detection of punctuation and disfluencies. In *Sixteenth annual conference of the international speech communication association*.

Eunah Cho, Jan Niehues, and Alex Waibel. 2012. Segmentation and punctuation prediction in speech language translation using a monolingual translation system. In *International Workshop on Spoken Language Translation (IWSLT) 2012*.

Heidi Christensen, Yoshihiko Gotoh, and Steve Renals. 2001a. Punctuation annotation using statistical prosody models. In *ISCA tutorial and research workshop (ITRW) on prosody in speech recognition and understanding*.

Heidi Christensen, Yoshihiko Gotoh, and Steve Renals. 2001b. Punctuation annotation using statistical prosody models. In *ISCA tutorial and research workshop (ITRW) on prosody in speech recognition and understanding*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Joris Driesen, Alexandra Birch, Simon Grimsey, Saeid Safarfashandi, Juliet Gauthier, Matt Simpson, and Steve Renals. 2014. Automated production of truecased punctuated subtitles for weather and news broadcasts. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Erik Edwards, Wael Salloum, Greg P Finley, James Fone, Greg Cardiff, Mark Miller, and David Suendermann-Oeft. 2017. Medical speech recognition: reaching parity with humans. In *International Conference on Speech and Computer*, pages 512–524. Springer.

Yoshihiko Gotoh and Steve Renals. 2000. Sentence boundary detection in broadcast speech transcripts.

Agustin Gravano, Martin Jansche, and Michiel Bacchiani. 2009. Restoring punctuation and capitalization in transcribed speech. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4741–4744. IEEE.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4229–4239.

Jing Huang and Geoffrey Zweig. 2002. Maximum entropy model for punctuation annotation from speech. In *Seventh International Conference on Spoken Language Processing*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI*, pages 2741–2749.

Ondřej Klejch, Peter Bell, and Steve Renals. 2017. Sequence-to-sequence models for punctuated transcription combining lexical and acoustic features. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5700–5704. IEEE.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Tal Levy, Vered Silber-Varod, and Ami Moyal. 2012. The effect of pitch, intensity and pause duration in punctuation detection. In *2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel*, pages 1–4. IEEE.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Wei Lu and Hwee Tou Ng. 2010. Better punctuation prediction with dynamic conditional random fields. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 177–186.

John Makhoul, Alex Baron, Ivan Bulyko, Long Nguyen, Lance Ramshaw, David Stallard, Richard Schwartz, and Bing Xiang. 2005. The effects of speech recognition and punctuation on information extraction performance. In *Ninth European Conference on Speech Communication and Technology*.

Binh Nguyen, Vu Bao Hung Nguyen, Hien Nguyen, Pham Ngoc Phuong, The-Loc Nguyen, Quoc Truong Do, and Luong Chi Mai. 2019a. Fast and accurate capitalization and punctuation for automatic speech recognition using transformer and chunk merging. *arXiv preprint arXiv:1908.02404*.

Binh Nguyen, Vu Bao Hung Nguyen, Hien Nguyen, Pham Ngoc Phuong, The-Loc Nguyen, Quoc Truong Do, and Luong Chi Mai. 2019b. Fast and accurate capitalization and punctuation for automatic speech recognition using transformer and chunk merging. *arXiv preprint arXiv:1908.02404*.

Vardaan Pahuja, Anirban Laha, Shachar Mirkin, Vikas Raykar, Lili Kotlerman, and Guy Lev. 2017. Joint learning of correlated sequence labelling tasks using bidirectional recurrent neural networks. *arXiv preprint arXiv:1703.04650*.

Stephan Peitz, Markus Freitag, Arne Mauser, and Hermann Ney. 2011a. Modeling punctuation prediction as machine translation. In *International Workshop on Spoken Language Translation (IWSLT) 2011*.

Stephan Peitz, Markus Freitag, Arne Mauser, and Hermann Ney. 2011b. Modeling punctuation prediction as machine translation. In *International Workshop on Spoken Language Translation (IWSLT) 2011*.

Wael Salloum, Gregory Finley, Erik Edwards, Mark Miller, and David Suendermann-Oeft. 2017. Deep learning for punctuation restoration in medical reports. In *BioNLP 2017*, pages 159–164.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Richard Sproat and Navdeep Jaitly. 2016. Rnn approaches to text normalization: A challenge. *arXiv preprint arXiv:1611.00068*.

Andreas Stolcke, Elizabeth Shriberg, Rebecca Bates, Mari Ostendorf, Dilek Hakkani, Madelaine Plauche, Gokhan Tur, and Yu Lu. 1998. Automatic detection of sentence boundaries and disfluencies based on recognized words. In *Fifth International Conference on Spoken Language Processing*.

Ottokar Tilk and Tanel Alumäe. 2015. Lstm for punctuation restoration in speech transcripts. In *Sixteenth annual conference of the international speech communication association*.

Ottokar Tilk and Tanel Alumäe. 2016. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In *Interspeech*, pages 3047–3051.

Nicola Ueffing, Maximilian Bisani, and Paul Vozila. 2013. Improved models for automatic punctuation prediction for spoken and written text. In *Interspeech*, pages 3097–3101.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Jiangyan Yi and Jianhua Tao. 2019. Self-attention based model for punctuation prediction using word and speech embeddings. In *Proc. ICASSP*, pages 7270–7274.