Common data sources

ACL

EMNLP sas

ICCASR

Interspeech

<https://www.opensubtitles.org/en/search>

Switchboard corpus

IMDA National Speech Corpus

Librespeech etc. not very useful since its just reading of books or articles, might as well just use open wiki etc. with more data.

En-cr TEDtalks or [TED – Ultimate Dataset | Kaggle](https://www.kaggle.com/miguelcorraljr/ted-ultimate-dataset) or Open speech more relavant.

Progress for 9-Dec-20: Cleanup of TED Ultimate Dataset- Removal of speaker name before semicolons, removal of round brackets, TODO: removal of square brackets (split using square brackets, compare case before after).

10-Dec-20: Completed TED cleanup.

Punctuation to predict: No Apostrophes. Include all

Lvl 1: .,?!- – —;: // Include all possibly readable symbols as text

Lvl 2: round brackets, square brackets, quotation marks

Leading Punctuation

Don’t handle caps together, things like Acronyms cannot be processed in the same manner?

~~TEDLIUM3 (contains audio to text without punctuation) example of output of many ASR systems.~~

~~Key features of TEDLIUM3 (Jun 2019):~~

~~207 hours -> 452 hours of audio (not relavant atm).~~

~~2.2M -> 4.9M words | 92976 -> 268231 word segments | 1495 -> 2351 talks~~

~~Autoalignment of audio files (.sph) with transcripts (.stm) using Kaldi.~~

Rich Transcription Categories

Speaker diarization

Sentence segmentation

Punctuation recovery or detection

Capitalization / truecasing

Disfluency detection and filtering:

1. [Recovering Capitalization and Punctuation Marks on Speech Transcriptions (inesc-id.pt)](https://www.inesc-id.pt/publications/4467/pdf) Recovering Capitalization and Punctuation Marks on Speech Transcriptions
2. [2004.00248.pdf (arxiv.org)](https://arxiv.org/pdf/2004.00248.pdf#:~:text=A%20pre-trained%20bidirectional%20encoder%20represen-%20tations%20from%20transformers,to%20learn%20task%20invariant%20knowledge%20for%20punctuation%20prediction.) Adversarial Transfer Learning for Punctuation Restoration 2020
3. [Robust Prediction of Punctuation and Truecasing for Medical ASR (aclweb.org)](https://www.aclweb.org/anthology/2020.nlpmc-1.8.pdf) 2020
4. [The Use of Prosody in a Combined System for Punctuation Generation and Speech Recognition (isca-speech.org)](https://www.isca-speech.org/archive/archive_papers/eurospeech_2001/e01_2757.pdf) Kim and Woodland 2001
5. [IEEE Xplore Full-Text PDF:](https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6423471) PROSODY-BASED SENTENCE BOUNDARY DETECTION IN CHINESE BROADCAST NEWS 2012
6. [IEEE Xplore Full-Text PDF:](https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9054337) IMPROVING PROSODY WITH LINGUISTIC AND BERT DERIVED FEATURES IN MULTI-SPEAKER BASED MANDARIN CHINESE NEURAL TTS 2020
7. [GitHub - nkrnrnk/BertPunc: SOTA punctation restoration (for e.g. automatic speech recognition) deep learning model based on BERT pre-trained model](https://github.com/nkrnrnk/BertPunc)
8. [Bidirectional Recurrent Neural Network with Attention Mechanism for Punctuation Restoration (semanticscholar.org)](https://pdfs.semanticscholar.org/8785/efdad2abc384d38e76a84fb96d19bbe788c1.pdf) 2016 INTERSPEECH
9. [download (psu.edu)](https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.28.276&rep=rep1&type=pdf) PERFORMANCE MEASURES FOR INFORMATION EXTRACTION
10. [RESTORING PUNCTUATION AND CAPITALIZATION IN TRANSCRIBED SPEECH (googleusercontent.com)](https://static.googleusercontent.com/media/research.google.com/en/pubs/archive/34562.pdf) 2009
11. [1908.02404.pdf (arxiv.org)](https://arxiv.org/pdf/1908.02404.pdf) Fast and Accurate Capitalization and Punctuation for Automatic Speech Recognition Using Transformer and Chunk Merging 2019
    1. Overlap chunk split, Transform, Overlap chunk merge (0.5 overlap)
    2. Utilised on on
12. [CMSY9 (iwslt.org)](https://workshop2015.iwslt.org/downloads/IWSLT_2015_RP_16.pdf) Punctuation Insertion for Real-time Spoken Language Translation 2015
13. Fastpunct? (no as lacking model or paper) Or bertpunc
14. [Leveraging a Character, Word and Prosody Triplet for an ASR Error Robust and Agglutination Friendly Punctuation Approach (isca-speech.org)](https://www.isca-speech.org/archive/Interspeech_2019/pdfs/2132.pdf) 2019 interspeech
15. [Evaluating an automatic speech recognition service | AWS Machine Learning Blog (amazon.com)](https://aws.amazon.com/blogs/machine-learning/evaluating-an-automatic-speech-recognition-service/) For end 2 end asr, talks about evaluation.
16. [1805.04699.pdf (arxiv.org)](https://arxiv.org/pdf/1805.04699.pdf) TEDLIUM3
17. [2012.02012.pdf (arxiv.org)](https://arxiv.org/ftp/arxiv/papers/2012/2012.02012.pdf) end 2 end ASR

Implement HMM

Does pretrained wiki bert perform on speech tasks?

Importance of punctuation in bert

Different types of punctuation, various uses of comma etc. (1 pg. 40)

Sentence punctuation (, . ! ? ellipsis) used as part of phrasing (generally succeeded by a space or quote)

Quotation marks or brackets

Sentence punctuation before quotation marks

To train sentence punct classifier:

Strip all non-sentence or word punctuation (brackets, quotation marks),

Tokenize,

assign word level sentence punctuation classes,

try to implement a cnn classifier over glove embedding of the sentence

Else, train a seq2seq

TODO:

Bert finetune punctuation classifier

[1910.13461.pdf (arxiv.org)](https://arxiv.org/pdf/1910.13461.pdf) train encoder?

Perhaps modify bert inputs, cls for valid punctuation, mask punctuations

Randomly replace words with homonyms and recover the original word (if using encoder)

Randomly replace words with homonyms or synonyms to expand dataset (Is this necessary?)

Bert understands context of punctuation, it might be very useful?

Mask and return probability of all punctuation tokens, return token if above threshold.

Dealing with quotation marks: Perhaps simply ignore or insert in a 2nd pass.

Does differentiating open and close quote matter?

Check how bert input is represented (If I’m not wrong, just tokens and labels)

Which bert model? Try to find the smallest that performs.

Should there be a nospace class?

Create punct importance function

Randomly mask punctuation to be classified? Or start from 0 or start from max? How to do this efficiently?

Preprocess tedlium with some pipeline: merge lines, split into xx length sequences, (Determine xx), if sequence contains repeated non-apos punct, pop. Is it true that a smaller length

Try to run some trained bert punct and create evaluation metric and evaluate using the nsc or tedlium

NOOOO TEDLIUM 1 Has no manual transcription!! Let me try tedlium 2 or opensubtitles

Bert final model: Input: string with apostrophes, tokenize, classify by all punct tokens, loop through model until all classes are zeros.

Drawbacks: Will need to run each classifier +1 time. Perhaps using the chunk overlap/merging can help to double check and build the subsequent punctuations? What about a 2/3 overlap – 1st 1/3 to predict base punct, 2nd 1/3 to double check and last 1/3 to add.

Training (Input: String strip punct, layer with most impt punct, or keep most impt punct and classify 2nd most impt punct (or if really common, keep 1st 2 punct and classify 3rd punct?!)

Pros: the position of all punctuation marks relative to the entire sub-word space is already defined during the bert pre-training process.

Considerations:

-Do I have to normalize the pre-trained embedding to fix the context of spoken words? Perhaps look into Robust Prediction of Punctuation and Truecasing for Medical ASR to see how they perform transfer learning?

Their Approach:

-Perform some Masked Language Model Finetuning on domain specific data.

-Domain+Task Adaptation: 50% of mask being punctuation – not very useful since punctuation is so little, and it might reduce the amount of fit of the data to the new domain??

Truecase and punct predict simultaneously?

Data format:

Punctuation mask (assign classes to words with ending with none, full-stop, comma, close quote, openquote, semicolon, colon, qmark, xmark.

Issue: double punctuation i.e. Titles with punct, abbreviations op. cit.?

Periods (***..***), question marks (***??***), and exclamation marks (***!!***) do not double up. Why? Because that would look weird.

—A period never follows an exclamation mark or question mark.

“Don’t be absurd!” said Henry. . . . “You remember what the Hatter said to her: ‘Not the same thing a bit! Why you might just as well say that “I see what I eat” is the same thing as “I eat what I see”!’ ”

Evaluation

Precision, Recall, SER ( S + D + I + H?) / N or normalised Levenshtein edit distance , F Score, AUC

Ted datasource analysis:

Contains many speaker tags and

Research papers

**Robust Prediction of Punctuation and Truecasing for Medical ASR** [2007.02025.pdf (arxiv.org)](https://arxiv.org/pdf/2007.02025.pdf) 11/07/2020

This paper focuses on the domain of Medical data.

Ideas featured:

* Effectiveness of pretrained BERT model vs model trained solely on medical data.
* Pretrained BERT consistently outperformed baseline BLSTM
* BioBERT / RoBERTa outperformed BERT base
* Subword models faired better than full-word models

Model comparison:

* 2 Recurrent: 3 LSTM, 3 BLSTM
* 2 Non-recurrent: Character level CNN Highway [Character-Aware Neural Language Models (arxiv.org)](https://arxiv.org/pdf/1508.06615.pdf), Transformer, Encoder (Attention is all you need)
* Compared various BERT models- BERT, RoBERTa, BioBERT

Joint modelling of Case and Punctuation, just using lexical information

1. Text -- Wordpiece Tokenizer --> Subword embeddings
2. Subword embeddings – pretrained BERT encoder --> BERT embedding
3. BERT encoder output – softmax (Wkhi + bk )--> punctuation labels
4. (BERT encoder output, punct. labels) -- softmax(Wl (pi ⊕ hi) + bl ) --> case labels

Punct labels: (Period, Comma, ? mark, None); Caps labels: (Upper, lower, CAPS, MixedCase)

Objective: Maximise joint probability Pr(p1:T, c1:T|x1:T).

Loss: weighted average of cross entropy loss for P. and C. (L = αLp + Lc). optimal α = 0.6

Training process:

Domain adaptation:

Finetune pretrained bert on medical data, masking 15% of tokens

Task adaptation:

Finetune model on medical data, with 50% of the masked tokens being punctuations.

Data Augmentation:

* Transcribe provided audio with some available algorithm to increase word error up to a maximum of 25% WER., training on the data with highest WER and test and dev set being the lowest and next lowest 50.
* Overlapping and Chunking – split into chunks of 200 words and overlap of 50 words. [1908.02404.pdf (arxiv.org)](https://arxiv.org/pdf/1908.02404.pdf)

I don’t understand the Robustness to ASR errors