5.1 - Inference

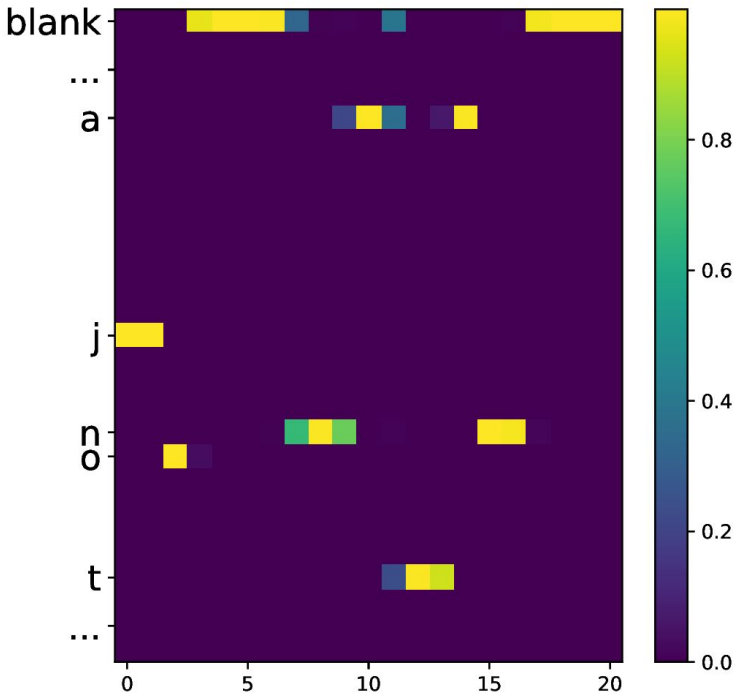
Inference

- turning model output into human-readable transcriptions
- “greedy”/best path decoding
  - take the character with the highest probability at each timestep
  - fast, doesn’t require information about text content
  - may output nonsense words

Inference

- beam search
  - consider N characters with highest probability at each timestep
  - identify sequence with highest combined probability
  - slower, may give better results than greedy search
- language models
  - feed the raw model output into a language model
  - LM has to be trained on suitable text beforehand!

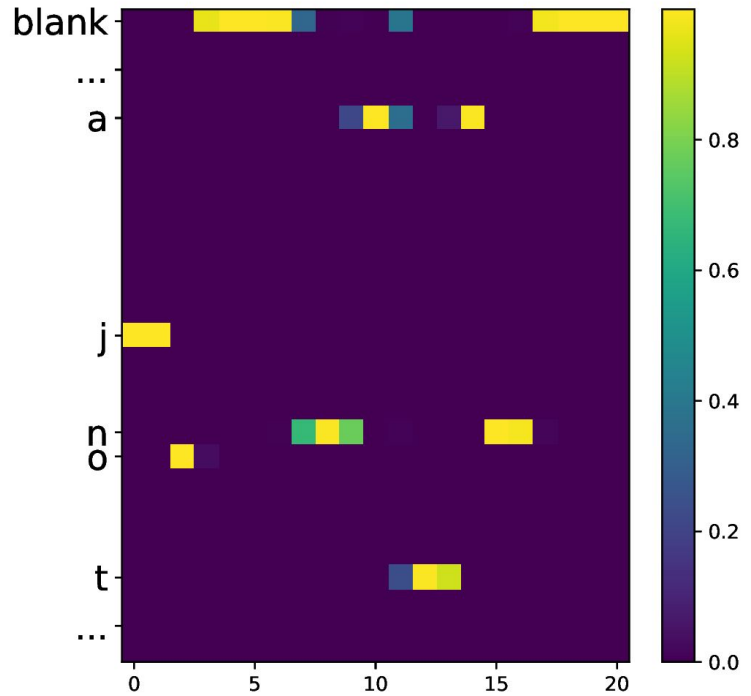
HTR Model Output



?

Transcription

HTR Model Output



1. identify max in each timestep (=column)
2. retrieve corresponding characters
   → jjo####nnnaattann####
3. collapse all repeated characters
   → jo#natan#
4. remove all blanks #
   → jonatan

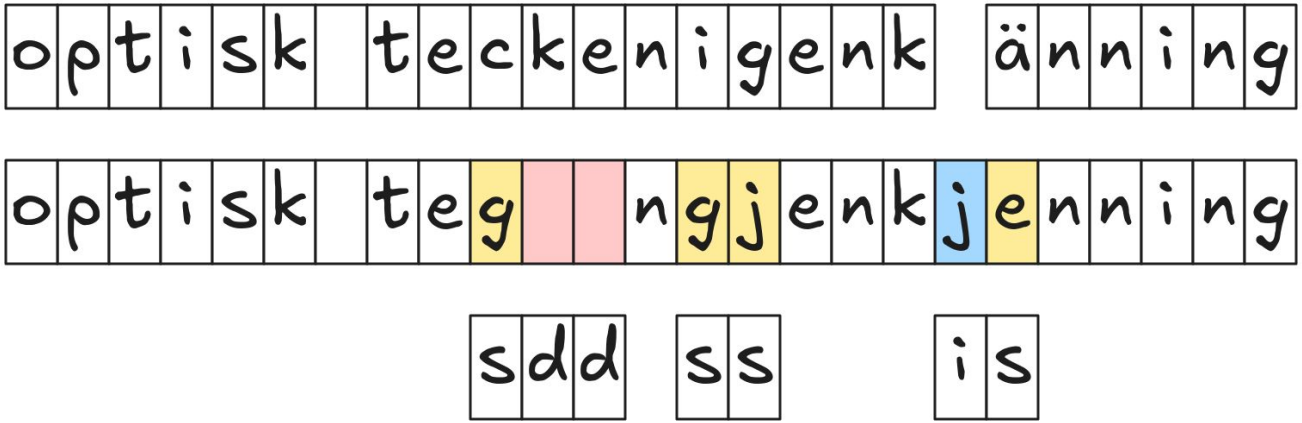
5.2 - Performance Metrics

Character and Word Error Rates

- normalised Levensthein distance between prediction and ground truth
  - at character-level
  - at word-level

$$ER = \frac{\text{Insertions} + \text{Deletions} + \text{Substitutions}}{\text{GT Char/Word Count}}$$

CER Example



$$CER = \frac{1\ \text{Insertion} + 2\ \text{Deletions} + 4\ \text{Substitutions}}{23\ \text{GT Characters}} = \frac{7}{23} \approx 30.4\%$$

WER Example

$$WER = \frac{0\ \text{Insertions} + 0\ \text{Deletions} + 1\ \text{Substitution}}{2\ \text{GT Words}} = \frac{1}{2} = 50\%$$

Hands-On Time!

Inference

Use your trained model (or the provided pre-trained checkpoint) to recognise the text lines in `ibsen/test`.

How well does your model work?
Are the automatic transcriptions (un)usable? Any mistakes that stand out? Other things that you noticed?

Feel free to also take a look at the performance for one or several of the splits in `bonnevie`.