Toward Robust Neural Machine Translation for Noisy Input Sequences

Matthias Sperber, Jan Niehues, Alex Waibel

Motivation

Real-world data is noisy
- Spelling mistakes
- Preprocessing errors
- Upstream errors, e.g. speech recognition output
  - this work
- Noisy inputs are challenging
  - How to translate errors?
  - Robustness: translate non-erroneous parts correctly
  - Train/test mismatch
  - NMT lacks robustness
    - [Chen+2016, Heigold+2017, Belinkov+2017, Ruiz+2017]

Example recognition errors:
- How much does his son decide to have a feast
- Buildings and boundaries around the location very part

Background

General-purpose regularizers
- good generalization → robustness
  - [Caramanis+2011]
- E.g. dropout

Here: Task-specific regularizers
- Randomly corrupt source-side during training
  - learn how to deal with errors, lower training/test mismatch
- Requires care: Trainability issues, explaining-away effects

Noise Model

Given:
- Noise magnitude \( \tau \in [0,1] \), sentence length \( n \), vocabulary \( V \)
- During training, for each source-side sentence
  - Sample \# \textit{ edits} → \text{TruncPoisson}(\tau \cdot n, n)
  - Sample \# substitutions, insertions deletions: \( \langle n_1, n_2, n_3 \rangle \sim \text{DiscrSimplex}(3, e) \)
  - Sample uniformly without replacement:
    - substitution, deletion positions \( \sim \{0,...,n\} \)
    - insertion positions \( \sim \{0,...,n\} \)
  - For substitutions, insertions: sample new word uniformly \( \sim V \)

Experiments

Data
- Fisher-Callhome Spanish-English speech translation corpus [Post+2013]
- Report results on Fisher/Dev speech recognition outputs (WER 41.3%)
- Model: Attentional encoder-decoder, standard settings
- Variational dropout (p=0.5), word type dropout (p=0.1)
- Pretrain on reference transcripts, fine-tune on noisy data

Findings

Main results (noisy inputs):
- Noise helps, sensitive to \( \tau \)
- Poor performance at \( \tau = 0.4 \) (close to test-time noise) → trainability issues!

Translating clean reference transcripts:
- Noise mostly does not help

Noise mostly does not help

N-gram precision (noisy inputs):
- More training noise → shorter outputs
- Del-only counteracts this, low precision

Influence of input WER
- Noisy training → output length more stable

Length control? But ideal precision/recall trade-off unclear for noisy inputs