ADAM MICKIEWICZ UNIVERSITY IN POZNAŃ
Faculty of Mathematics and Computer Science
Department of Natural Language Processing

# Automatic error correction in ASR

2019-05-23

Tomasz Ziętkiewicz

# Outline

# About me

- PhD Student at "Applied Doctorate" studies
- Work at Samsung R&D Poland
- Area of work & research:
  - Automatic Speech Recognition post-processing
  - Inverse Text Normalization
  - Automatic correction of ASR errors
  - Sequence labeling
  - Sequence-to-sequence methods

# About me

- PhD Student at "Applied Doctorate" studies
- Work at Samsung R&D Poland
- Area of work & research:
  - Automatic Speech Recognition post-processing
  - Inverse Text Normalization
  - Automatic correction of ASR errors
  - Sequence labeling
  - Sequence-to-sequence methods

# About me

- PhD Student at "Applied Doctorate" studies
- Work at Samsung R&D Poland
- Area of work & research:
  - Automatic Speech Recognition post-processing
  - Inverse Text Normalization
  - Automatic correction of ASR errors
  - Sequence labeling
  - Sequence-to-sequence methods

# About me

- PhD Student at "Applied Doctorate" studies
- Work at Samsung R&D Poland
- Area of work & research:
  - Automatic Speech Recognition post-processing
  - Inverse Text Normalization
  - Automatic correction of ASR errors
  - Sequence labeling
  - Sequence-to-sequence methods

# About me

- PhD Student at "Applied Doctorate" studies
- Work at Samsung R&D Poland
- Area of work & research:
  - Automatic Speech Recognition post-processing
  - Inverse Text Normalization
  - Automatic correction of ASR errors
  - Sequence labeling
  - Sequence-to-sequence methods

# About me

- PhD Student at "Applied Doctorate" studies
- Work at Samsung R&D Poland
- Area of work & research:
  - Automatic Speech Recognition post-processing
  - Inverse Text Normalization
  - Automatic correction of ASR errors
  - Sequence labeling
  - Sequence-to-sequence methods

# About me

- PhD Student at "Applied Doctorate" studies
- Work at Samsung R&D Poland
- Area of work & research:
  - Automatic Speech Recognition post-processing
  - Inverse Text Normalization
  - Automatic correction of ASR errors
  - Sequence labeling
  - Sequence-to-sequence methods

# About me

- PhD Student at "Applied Doctorate" studies
- Work at Samsung R&D Poland
- Area of work & research:
  - Automatic Speech Recognition post-processing
  - Inverse Text Normalization
  - Automatic correction of ASR errors
  - Sequence labeling
  - Sequence-to-sequence methods

# Outline

# Problem definition

Given corpus of hypothesis from Automatic Speech Recognition system and corresponding reference utterances, learn a transformation from the former to the latter

# Rationale

Why such transformation is needed? It can be used in ASR system as postprocessing stage as:

- re-scoring of hypothesis produced by ASR using information not present at earlier stages of processing (i.e. one directrional LM)
- adaptation of a black box, general domain ASR system to some specific domain

# Outline

# Related work

- Cucu, Horia, Andi Buzo, Laurent Besacier, and Corneliu Burileanu, 2013. „Statistical Error Correction Methods for Domain-Specific ASR Systems”
- Luis Fernando D'Haro, Rafael E. Banchs, „Automatic Correction of ASR outputs by Using Machine Translation”, Interspeech 2016
- Jinxi Guo, Tara N. Sainath, Ron J. Weiss „A Spelling Correction Model For End-to-end Speech Recognition”

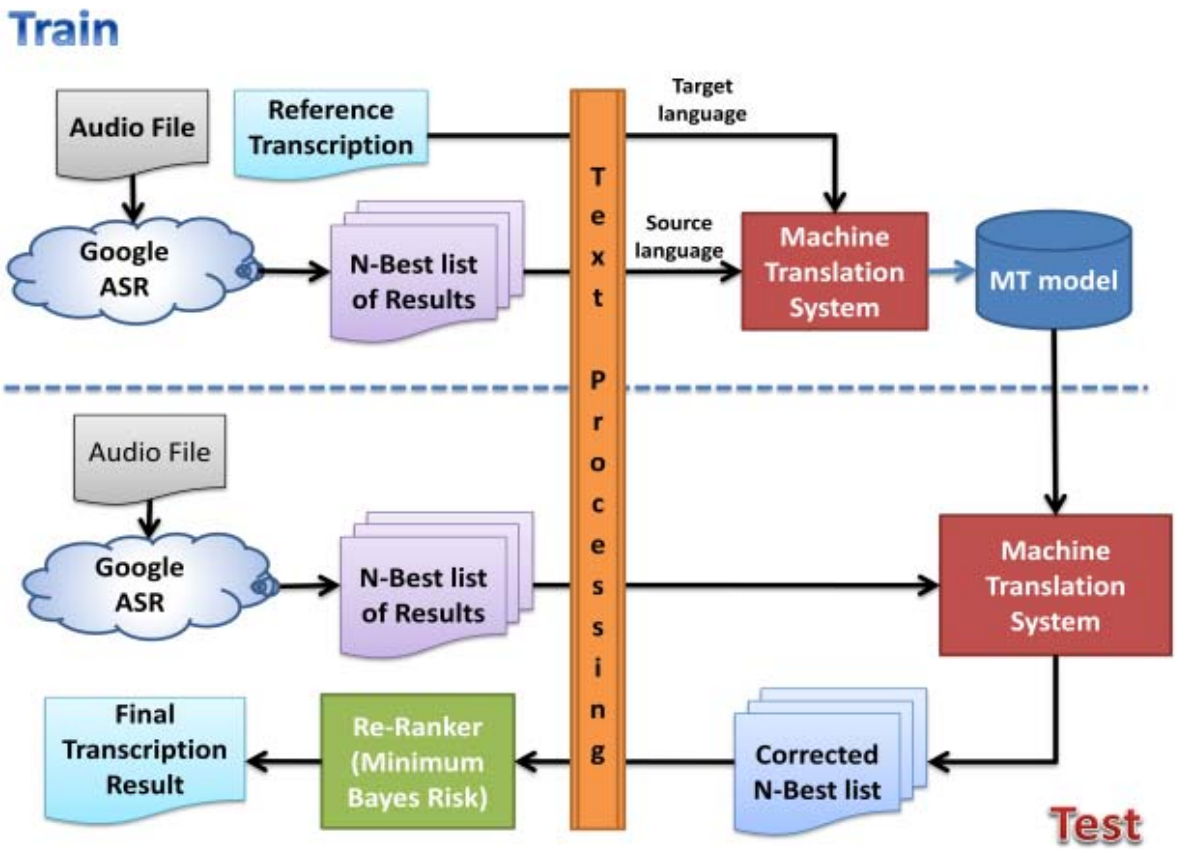
# Statistical Error Correction Methods for Domain-Specific ASR Systems

- error correction using SMT (Statistical Machine Translation) model.
- trained on relatively small parallel corpus
- 2000 ASR transcripts and their manually corrected versions.
- At evaluation time the model is used to "translate" ASR hypothesis into it's corrected form.
- Results: 10.5% relative WER improvement by reducing the baseline ASR system's WER from 11.4 to 10.332 .

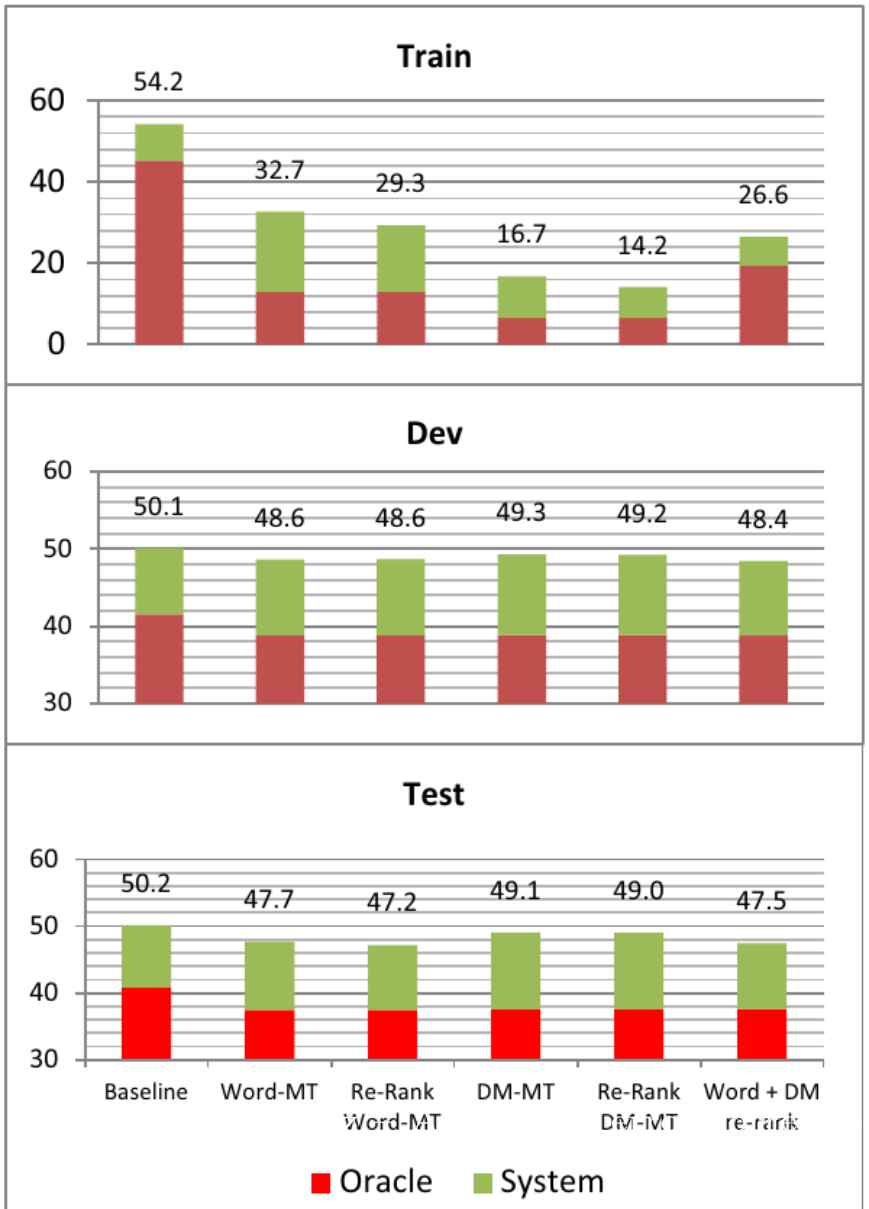
# Automatic Correction of ASR outputs by Using Machine Translation

- Phrase-based machine translation used on reference sentences and n-best list of hypothesis from ASR
- Uses Minimum Bayes Risk re-ranker to choose among translated entries from n-best list

# Automatic Correction of ASR outputs by ... System architecture



# Automatic Correction of ASR outputs by ... - Results



# Outline

# A SPELLING CORRECTION MODEL FOR END-TO-END SPEECH RECOGNITION

*Jinxi Guo*[1*] , *Tara N. Sainath*[2], *Ron J. Weiss*[2]

[1]University of California, Los Angeles, USA
[2]Google Inc., USA

lennyguo@g.ucla.edu, {tsainath, ronw}@google.com

# Motivation

- Popularity of end-to-end ASR models
- Acousting, pronunciation and language model combined in one neural network
- Problem: needs annotated audio data
- LM trained on small dataset compared with "traditionalapproach
- Worse performance on rare words

# Motivation

- Popularity of end-to-end ASR models
- Acousting, pronunciation and language model combined in one neural network
- Problem: needs annotated audio data
- LM trained on small dataset compared with "traditionalapproach
- Worse performance on rare words

# Motivation

- Popularity of end-to-end ASR models
- Acousting, pronunciation and language model combined in one neural network
- Problem: needs annotated audio data
- LM trained on small dataset compared with "traditionalapproach
- Worse performance on rare words

# Motivation

- Popularity of end-to-end ASR models
- Acousting, pronunciation and language model combined in one neural network
- Problem: needs annotated audio data
- LM trained on small dataset compared with "traditionalapproach
- Worse performance on rare words

# Motivation

- Popularity of end-to-end ASR models
- Acousting, pronunciation and language model combined in one neural network
- Problem: needs annotated audio data
- LM trained on small dataset compared with "traditionalapproach
- Worse performance on rare words

# Possible solutions

- Incorporating external LM trained on text-only data
  - Rescoring n-best decoded hypothesis from end-to-end ASR
  - Incorporate RNN-LM into first-pass beam search by shallow, cold or deep fusion
- Rare words and proper nouns are still problematic with this approach
- Why?
- LM trained with other objective then correcting e2e model's errors

# Possible solutions

- Incorporating external LM trained on text-only data
  - Rescoring n-best decoded hypothesis from end-to-end ASR
  - Incorporate RNN-LM into first-pass beam search by shallow, cold or deep fusion
- Rare words and proper nouns are still problematic with this approach
- Why?
- LM trained with other objective then correcting e2e model's errors

# Possible solutions

- Incorporating external LM trained on text-only data
  - Rescoring n-best decoded hypothesis from end-to-end ASR
  - Incorporate RNN-LM into first-pass beam search by shallow, cold or deep fusion
- Rare words and proper nouns are still problematic with this approach
- Why?
- LM trained with other objective then correcting e2e model's errors

# Possible solutions

- Incorporating external LM trained on text-only data
  - Rescoring n-best decoded hypothesis from end-to-end ASR
  - Incorporate RNN-LM into first-pass beam search by shallow, cold or deep fusion
- Rare words and proper nouns are still problematic with this approach
- Why?
- LM trained with other objective then correcting e2e model's errors

# Possible solutions

- Incorporating external LM trained on text-only data
  - Rescoring n-best decoded hypothesis from end-to-end ASR
  - Incorporate RNN-LM into first-pass beam search by shallow, cold or deep fusion
- Rare words and proper nouns are still problematic with this approach
- Why?
- LM trained with other objective then correcting e2e model's errors

# Possible solutions

- Incorporating external LM trained on text-only data
  - Rescoring n-best decoded hypothesis from end-to-end ASR
  - Incorporate RNN-LM into first-pass beam search by shallow, cold or deep fusion
- Rare words and proper nouns are still problematic with this approach
- Why?
- LM trained with other objective then correcting e2e model's errors

# Solution

Proposed solution: spelling corrector model on text-to-text (hypothesis-to-refrerence) pairs.

# Dataset

- LibriSpeech
- Large-scale (1000 hours) corpus of read English speech
- audiobooks from the LibriVox project
- carefully segmented and aligned
- http://www.openslr.org/12/
- License: CC BY 4.0

# Dataset

- LibriSpeech
- Large-scale (1000 hours) corpus of read English speech
- audiobooks from the LibriVox project
- carefully segmented and aligned
- http://www.openslr.org/12/
- License: CC BY 4.0

# Dataset

- LibriSpeech
- Large-scale (1000 hours) corpus of read English speech
- audiobooks from the LibriVox project
- carefully segmented and aligned
- http://www.openslr.org/12/
- License: CC BY 4.0

# Dataset

- LibriSpeech
- Large-scale (1000 hours) corpus of read English speech
- audiobooks from the LibriVox project
- carefully segmented and aligned
- http://www.openslr.org/12/
- License: CC BY 4.0

# Dataset

- LibriSpeech
- Large-scale (1000 hours) corpus of read English speech
- audiobooks from the LibriVox project
- carefully segmented and aligned
- http://www.openslr.org/12/
- License: CC BY 4.0

# Dataset

- LibriSpeech
- Large-scale (1000 hours) corpus of read English speech
- audiobooks from the LibriVox project
- carefully segmented and aligned
- http://www.openslr.org/12/
- License: CC BY 4.0

# Baseline

- LAS - Listen Attend and Spell
- Encoder-decoder with attention

# Spelling correction model

- attention-based encoder-decoder sequence-to-sequence

# Architecture

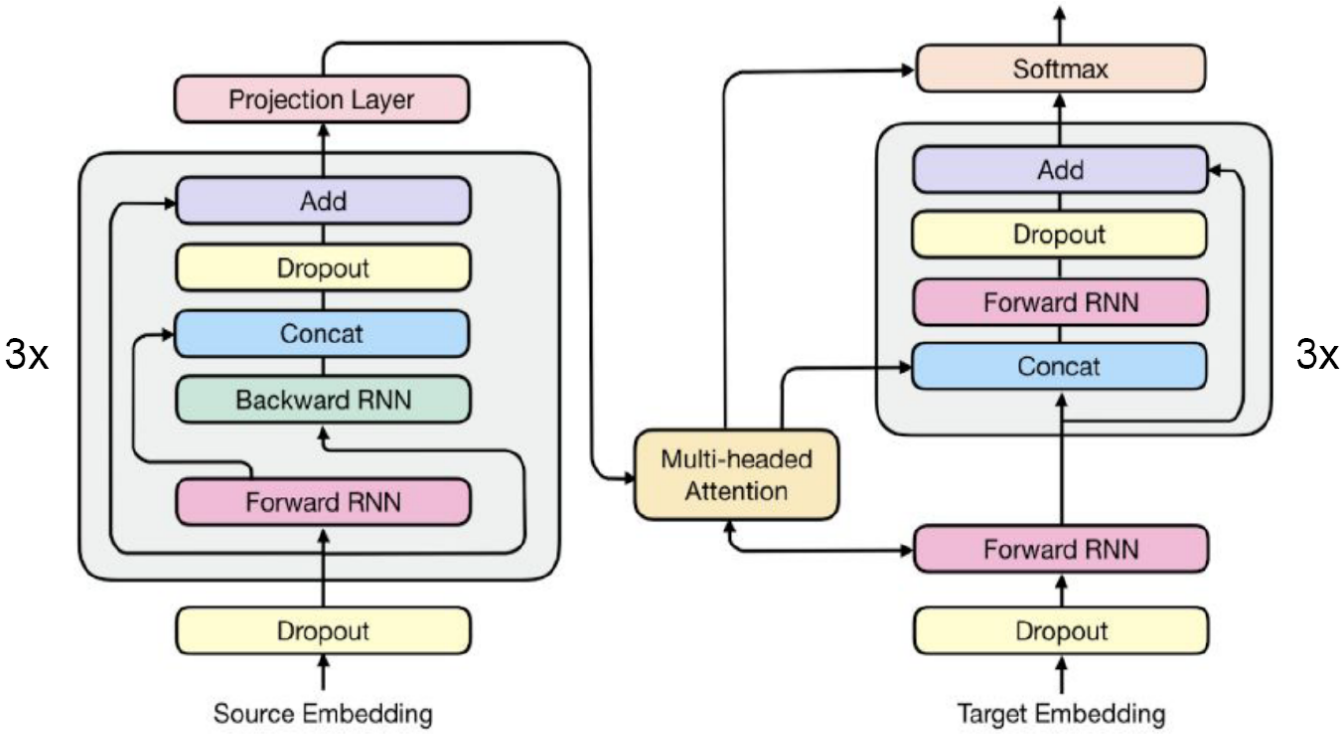



**Fig. 1**. Spelling Correction model architecture.

# Results

| System | Dev-clean | Test-clean |
|---|---|---|
| LAS | 5.80 | 6.03 |
| LAS $\rightarrow$ LM (8) | 4.56 | 4.72 |
| LAS-TTS | 5.68 | 5.85 |
| LAS-TTS $\rightarrow$ LM (8) | **4.45** | **4.52** |
| LAS $\rightarrow$ SC (1) | 5.04 | 5.08 |
| LAS $\rightarrow$ SC (8) $\rightarrow$ LM (64) | **4.20** | **4.33** |
| LAS $\rightarrow$ SC-MTR (1) | 4.87 | 4.91 |
| LAS $\rightarrow$ SC-MTR (8) $\rightarrow$ LM (64) | **4.12** | **4.28** |

**Table 1**. Word error rates (WERs) on LibriSpeech "clean" sets comparing different techniques for incorporating text-only training data. Numbers in parentheses indicate the number of input hypotheses considered by the corresponding model.

# Results

| System | Dev-clean | Test-clean |
|---|---|---|
| LAS | 3.11 | 3.28 |
| LAS $\rightarrow$ SC (1) | 3.01 | 3.02 |
| LAS $\rightarrow$ SC (8) | **1.63** | **1.68** |

**Table 2**. Oracle WER before and after applying the SC model.

# Results

| System | Dev-clean | Dev-TTS |
|---|---|---|
| LAS baseline | 5.80 | 5.26 |
| LAS $\rightarrow$ SC (1) | 5.04 | **3.45** |
| LAS $\rightarrow$ SC (8) $\rightarrow$ LM (64) | 4.20 | **3.11** |

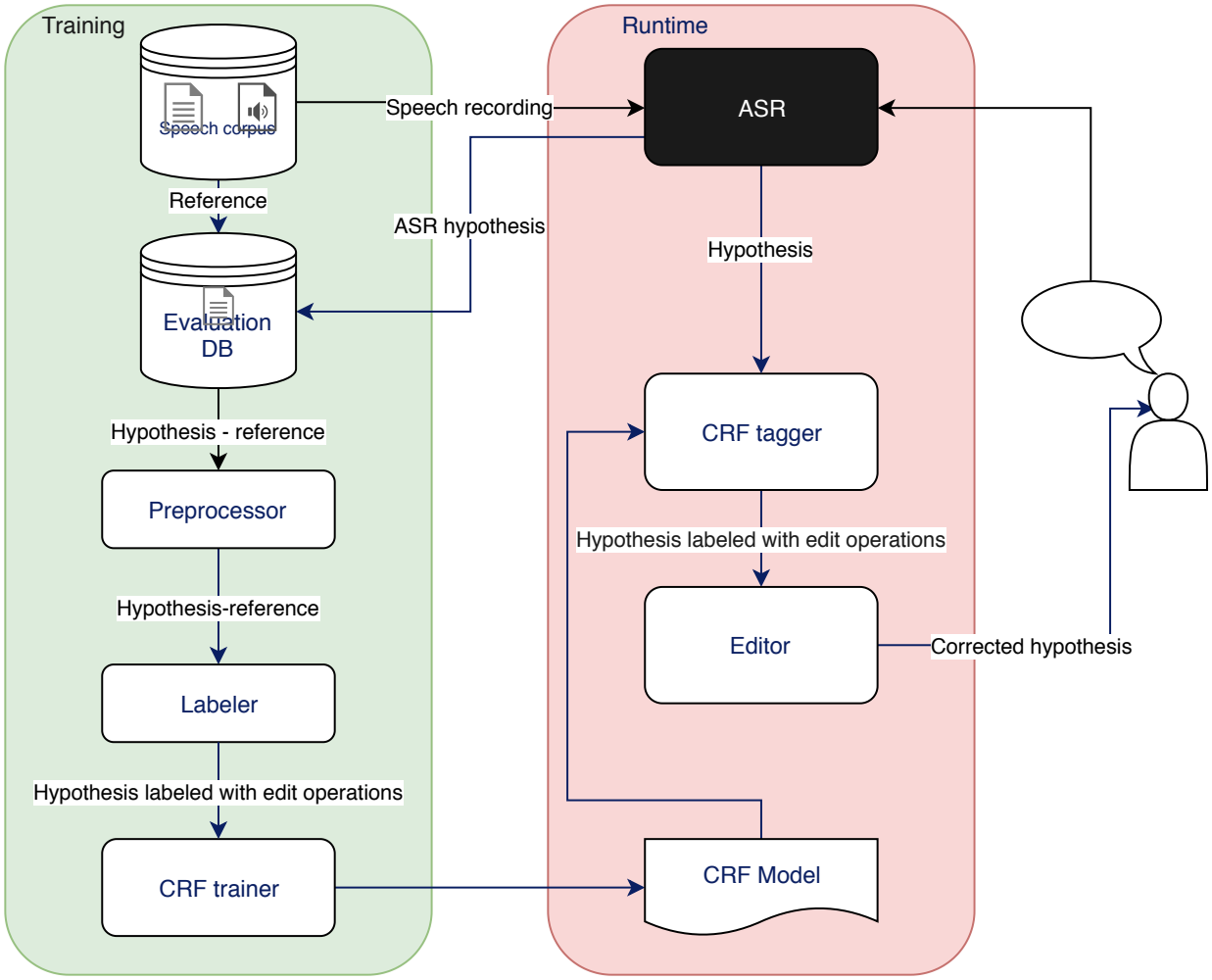
**Table 3**. WER comparison on a real audio and TTS dev sets.

# Outline

# Edit operations tagger approach

- Train
  - Compare ASR hypothesis and reference sentences from parallel corpora
  - Extract pre-defined edit operations from the comparison
  - Create corpora with ASR hypothesis tagged with edit operations labels
  - Train a tagger using this corpora
- Test
  - Use tagger on ASR hypothesis
  - Apply edit operation to the hypothesis

# Architecture



# Example

Reference: ”Multimodal distribution“
Hypothesis: ”Multi modal distribution“
Tagged hypothesis:

| Word | Tag |
|---|---|
| Multi | join |
| modal | None |
| distribution | None |

# Edit operations tagger approach

- Pros
  - Safe - Default operation - do nothing
  - Easy to control - filter operation using tag score treshold
  - Set of used edit operations can be adjusted to fix only specific kind of errors
- Cons
  - Need to manually define edit operations
  - Need to implement edit operations deducer
  - Set of operations is limited due to performance constraints

# Implementation

- Current implementation: Conditional Random Fields (CRF) tagger
  - Handcrafted features: prefix, suffix, length, left context, right context
  - CrfSuite library
  - Number of edit operations limited to 200 most popular ones
- Planned implementation: Neural tagger using RNNs/LSTMs

# Sequence-to-sequence approach

Learn seq2seq model from hypothesis to reference sentences.

- Pros
  - end-to-end - no need for handcrafted features, edit opearations etc.
- Cons
  - end-to-end - harder to control and tune
  - Needs more data
  - Higher risk for false positive errors (Chaning correct words from hypothesis)

# Thank you

Thank you for your attention!