ADAM MICKIEWICZ UNIVERSITY IN POZNAŃ

Faculty of Mathematics and Computer Science
Department of Artificial Inteligence

[2]Samsung Poland R&D Institute

Post-editing and rescoring of ASR results with edit operations tagging

PolEval 2020, Task 1: description of submitted solution

22 października 2020

Tomasz Ziętkiewicz[1,2]

Outline

1 Introduction

2 Related work

3 Data

4 Method

5 Results

Task description

Goal

"[...] create a system for converting a sequence of words from a specific automatic speech recognition (ASR) system into another sequence of words that more accurately describes the actual spoken utterance. "

`http://poleval.pl/tasks/task1/`

Data

Dataset consist of pairs:

- hypothesis generated by ASR system
- correct transcription of the utterance

Datasets contents

Each dataset contains:

- 1-best output - each utterance containing a single best transcript of the ASR output
- n-best output - each utterance containing up to 100 best alternative hypotheses of the ASR output
- lattice output - each utterance containing a list of arcs forming a lattice of the ASR output
  - each line contains the following fields: start node, end node, words, language weight, acoustic weight, list of phonetic-level states
- reference - file similar to the 1-best output, but containing the actual reference transcript

ASR

- trained on the Clarin-PL studio corpus [MKBJL15]:
  - 56 hours
  - Polish
  - Recorded in studio
- tri3b model from the ClarinStudioKaldi setup [KMBW17]

Proposed solution

- Don't learn transformation from hypothesis to reference directly
- Learn to recognize errors and how to correct them
- tagging with edit operations tags
- novel approach to the problem of correcting speech recognition errors
- used for other problems like grammatical error correction

Outline

1 Introduction

2 Related work

3 Data

4 Method

5 Results

Related work

- A spelling correction model for end-to-end speech recognition; Jinxi Guo, Tara N. Sainath, Ron J. Weiss (Google); ICASSP 2019 [GSW19]
- encoder-decoder, sequence-to-sequence
- 600M pairs after producing data with TTS

Related work

- Oleksii Hrinchuk, Mariya Popova, and Boris Ginsburg, Correction of automatic speech recognition with transformer sequence-to-sequence model, 2019 [HPG19]
- transformer-based, encoder-decoder, sequence to sequence model
- trained on 2.5M examples
- relative WER reduction of around 12%.

Related work

- Roman Grundkiewicz and Marcin Junczys-Dowmunt, Nearhuman-level performance in grammatical error correction with hybrid machine translation, 2018 [GJD18]
- massive data augmentation ($4k \rightarrow 100M$)

Related work

- Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka,and Aliaksei Severyn, Encode, tag, realize: High-precision textediting, 2019 [MKR+19]
- edit operation tagging
- tasks:
  - sentence fusion
  - sentence splitting
  - abstractive summarization
  - grammar correction tasks
- tagger using Transformer
- competitive results for small training datasets
- very short inference times

Outline

1 Introduction

2 Related work

3 Data

4 Method

5 Results

Datasets

- Train/dev/test:
  - Clarin-PL - Dataset on which ASR was trained
  - Polish parliament corpus
- Evaluation data:
  - PINC
    - Polish Interpreting Corpus
    - Parallel, Polish-English, English-Polish corpus of European Parliament speeches and their corresponding simultaneous interpretations
    - pincproject2020.wordpress.com

Dataset normalization

- all words are lowercased
- punctuation marks are removed
- numbers and special characters are replaced by their spoken forms

Datasets statistics

	Clarin-PL studio			PPC	PELCRA PARL	PINC
	Train	Test	Dev	Train	Train	Eval
Sentences	11222	1362	1229	6752	8066	462
WER	9.59	12.08	12.39	45.57	59.95	27.6
oracle WER	3.75	4.72	4.93	30.71	-	17.7
Avg. length	22	21	21	104	12	169
Min. length	3	6	7	1	2	70
Max. length	55	53	49	341	47	435

Data augumentation

- Additional corpus:
  - Trained Kaldi ASR using Clarin-PL and Kaldi recipe
  - Decoded PELCRA-PARL corpus to get more training data for ASR-correction task
- Use up to 10-best hypotheses to get 10x more reference-hypothesis pairs
- Resulting dataset:

	Train	Test	Dev
Sentences	110 059	11 930	10 254

Outline

1 Introduction

2 Related work

3 Data

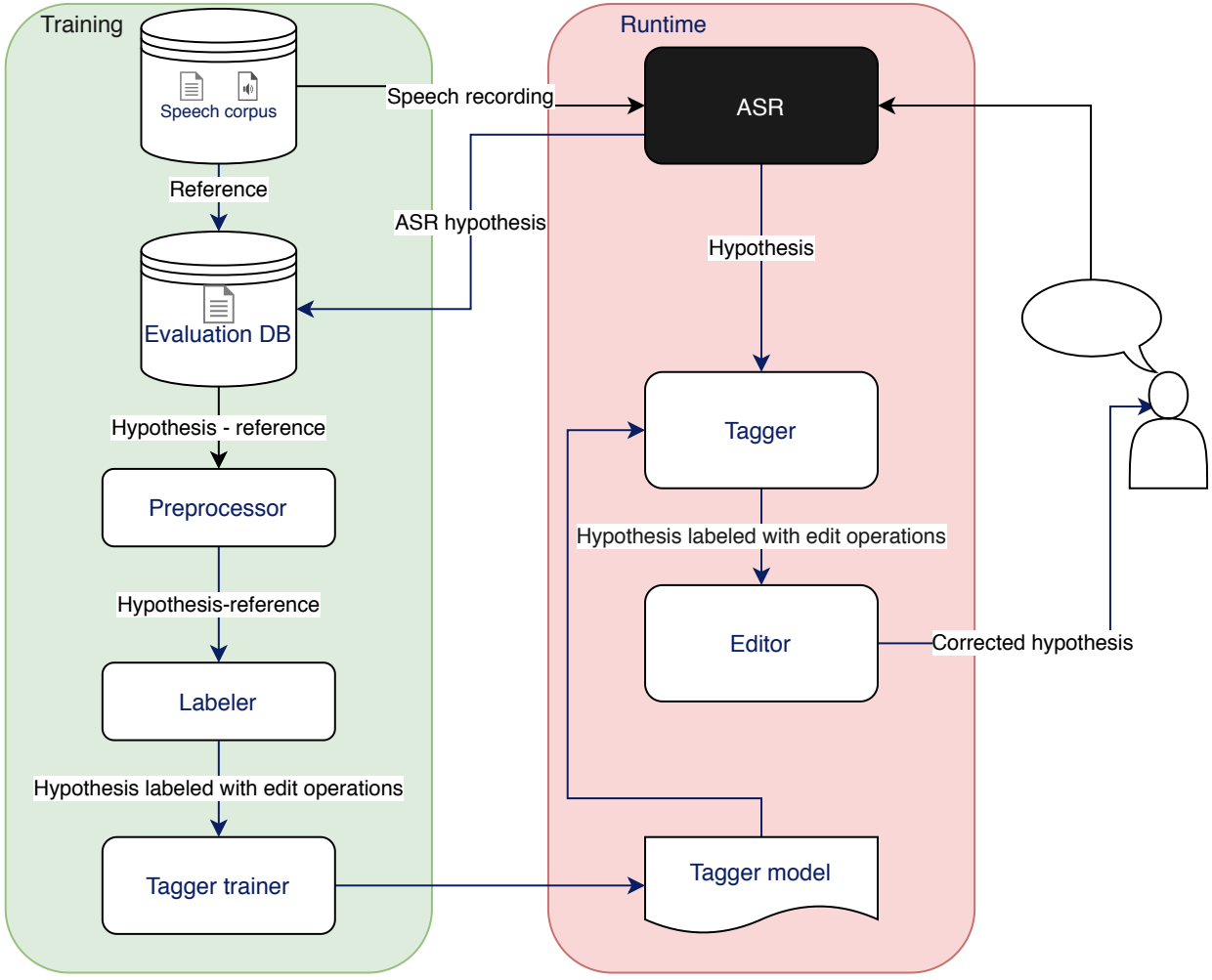
4 Method

5 Results

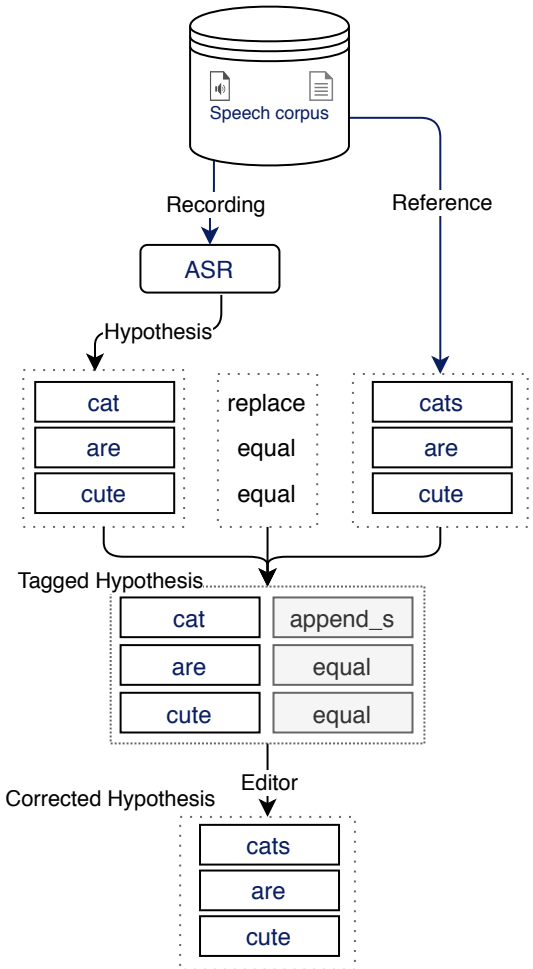
Edit operations tagger approach

- 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


Speech corpus
Recording
Reference
ASR
Hypothesis
cat
are
cute
replace
equal
equal
cats
are
cute
Tagged Hypothesis
cat
are
cute
append_s
equal
equal
Editor
Corrected Hypothesis
cats
are
cute

Examples of edit operations

name	description	example
`del`	deletes a token	cat →
`append_s`	appends given suffix to the token	cat → cats
`add_prefix_`	prepends given prefix to the token	owl → howl
`remove_suff_1`	removes 1 char from the end	cats → cat
`remove_pref_1`	removes 1 char from the beginning	howl → owl
`join`	joins token with previous one	book store → bookstore
`join_-`	joins token with previous one	long term → long-term
`replace_`	replaces token with given string	cat → hat

Examples of edit operations

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

- Polish contextual word embeddings [GG18] + BiLSTM with CRF layer
- SequenceTagger from Flair NLP library [ABB+19]

Outline

1 Introduction

2 Related work

3 Data

4 Method

5 Results

Evaluation

- Multiple submissions possible
- Evaluation metric: Word Error Rate (WER)

Word Error Rate

WER - Word Error Rate of hypothesis corrected by the proposed system, averaged over all tests sentences.

$$WER = \frac{N_{del} + N_{sub} + N_{ins}}{N_{ref}}$$

where N_{sub} = number of substitutions, N_{del} = number of deletions, N_{ins} = number of insertions, N_{ref} - length of reference sentence.

Results

	Clarin			PPC	PINC
	Train	Test	Dev	-	Eval
Raw ASR 1best	9.59	12.08	12.39	45.64	27.6
lattice oracle WER	3.75	4.72	4.93	30.71	17.7
Rel. error reduction	8.13%	4.97%	5.81%	0.37%	-
Flair tagger (from 1best)	-	10.7	-	-	24.7
Rel. error reduction	-	11.42%	-	-	10.5%

Submission	Affiliation	WER %	Changes %
KRS + spaces	UJ. AGH	25.9	3.6
KRS	UJ. AGH	26.9	1.6
Polbert	https://skok.ai/	26.9	2.1
BiLSTM-CRF edit-operations tagger	Adam Mickiewicz University	24.7	6.2
base-4g-rr	Samsung R&D Institute Poland	27.7	2.0
t-REx_k10	Uniwersytet Wrocławski	24.9	14.2
t-REx_k5	Uniwersytet Wrocławski	25.0	14.2
t-REx_fbs	**Uniwersytet Wrocławski**	**24.31**	**17.2**
PJA_CLARIN_1k	Polish-Japanese Academy of Information Technology	33.5	9.1
PJA_CLARIN_10k	Polish-Japanese Academy of Information Technology	32.0	9.6
PJA_CLARIN_20k	Polish-Japanese Academy of Information Technology	31.8	9.9
PJA_CLARIN_40k	Polish-Japanese Academy of Information Technology	31.8	10.3
PJA_CLARIN_50k	Polish-Japanese Academy of Information Technology	31.8	10.2
CLARIN_SEJM_40k	Polish-Japanese Academy of Information Technology	33.7	19.1
CLARIN_SEJM_50k	Polish-Japanese Academy of Information Technology	32.5	17.7
MLM+bert_base_polish		73.9	2.1
tR-Ex_xk	Uniwersytet Wrocławski. Instytut Informatyki	25.7	18.1
tR-Ex_fbs	Uniwersytet Wrocławski. Instytut Informatyki	24.31	17.2
tR-Ex_fx	Uniwersytet Wrocławski. Instytut Informatyki	25.0	23.3
tR-Ex_kxv2	Uniwersytet Wrocławski. Instytut Informatyki	25.5	17.1

Experiments

- Polbert:
  - For edit-operation tagging - only slight improvement - 10.7 vs 10.6 WER
  - For guessing a correct word - no improvement or degradation of results

Thank you

Thank you for your attention!

References I

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf, FLAIR: An easy-to-use framework for state-of-the-art NLP, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations) (Minneapolis, Minnesota), Association for Computational Linguistics, June 2019, pp. 54–59.

Roman Grundkiewicz and Marcin Junczys-Dowmunt, Near human-level performance in grammatical error correction with hybrid machine translation.

References II

J. Guo, T. N. Sainath, and R. J. Weiss, A spelling correction model for end-to-end speech recognition, ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 5651–5655.

Oleksii Hrinchuk, Mariya Popova, and Boris Ginsburg, Correction of automatic speech recognition with transformer sequence-to-sequence model, 2019.

Danijel Korzinek, Krzysztof Marasek, Lukasz Brocki, and Krzysztof Wolk, Polish read speech corpus for speech tools and services, CoRR abs/1706.00245 (2017).

Krzysztof Marasek, Danijel Koržinek, Łukasz Brocki, and Kamila Jankowska-Lorek, Clarin-PL studio corpus (EMU), 2015, CLARIN-PL digital repository.

References III

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn, Encode, tag, realize: High-precision text editing, 2019.

Łukasz Borchmann, Andrzej Gretkowski, and Filip Graliński, Approaching nested named entity recognition with parallel lstm-crfs, Proceedings of the PolEval 2018 Workshop (Warszawa) (Maciej Ogrodniczuk and Łukasz Kobyliński, eds.), Institute of Computer Science, Polish Academy of Science, 2018, pp. 63–73.