InCa and InDia:

Inline Casing and Diacritization Preprocessing For Robust-to-Noise Tokenization and Interpretability

Kirill Semenov, Martin Popel
kirill.semenov@uzh.ch

Universität Zürich[UZH]

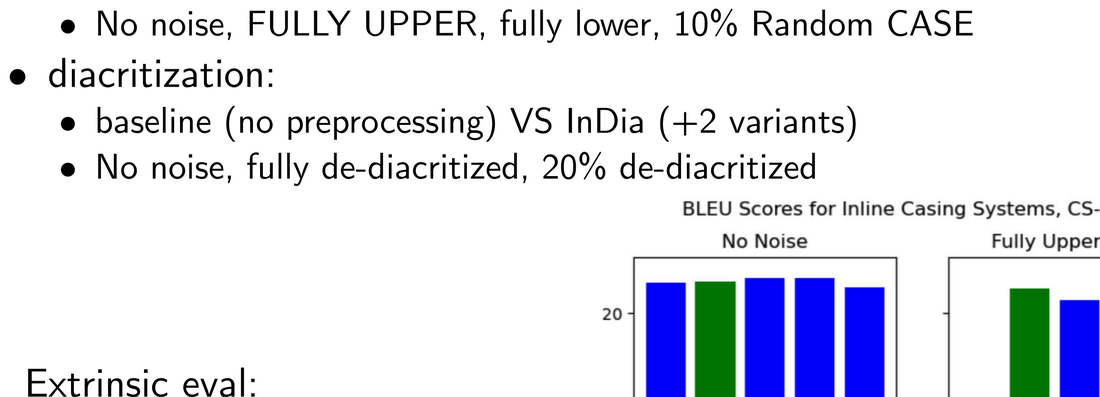
FACULTY OF MATHEMATICS AND PHYSICS
Charles University

Motivation

- Tokenizers are overly sensitive to "typographical variation":
  - "Zürich", "Zurich" and "ZURICH" will be tokenized differently
- We want to consistently tokenize such words
  - Easiest solution: de-diacritize, lowercase - **losing information!**
- Inline approach: allocate information about casing on a nearby **flag**:
  - "Hello" → "**T** hello", "HELLO" → "**U** hello"
- **BUT: we also want to minimize the encoded lengths**
- **BUT: never applied to diacritization**

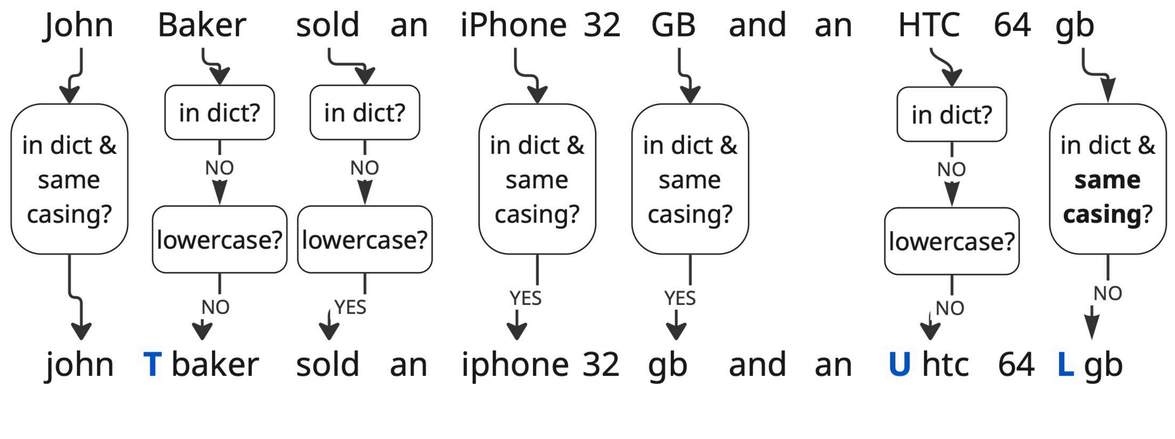
InCa: Dictionary-Based Inline Casing

I. Training: *store the most frequent casings in a dictionary*



II. Encoding: *mark only non-frequent or OOV casings*

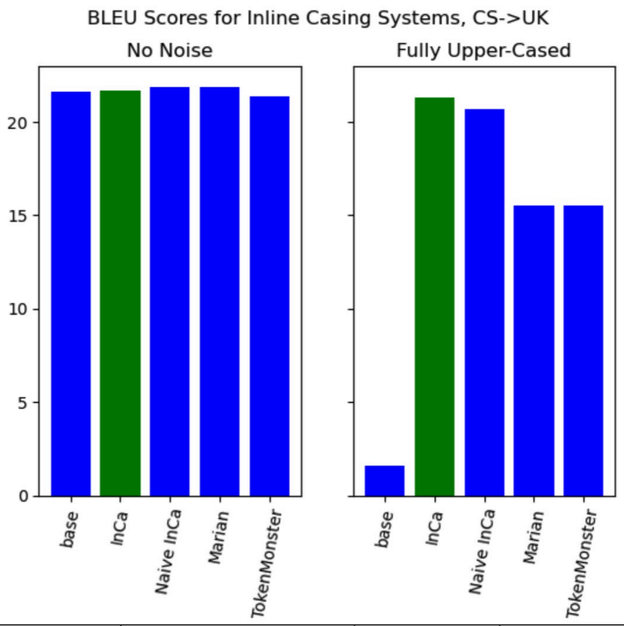
{ john: John, iphone: iPhone, gb: GB }



Shorter token sequences! Consistent base encoding! Sentence-level flags! Fully reversible!

Case Study: CS-UK MT

- Tokenizer: UnigramLM
- Extrinsic (BLEU, chrF, COMET) & intrinsic (char/token, avg rank) eval
- casing:
  - InCa VS baseline, "naive" InCa (no dictionary) Marian, TokenMonster
  - No noise, FULLY UPPER, fully lower, 10% Random CASE
- diacritization:
  - baseline (no preprocessing) VS InDia (+2 variants)
  - No noise, fully de-diacritized, 20% de-diacritized



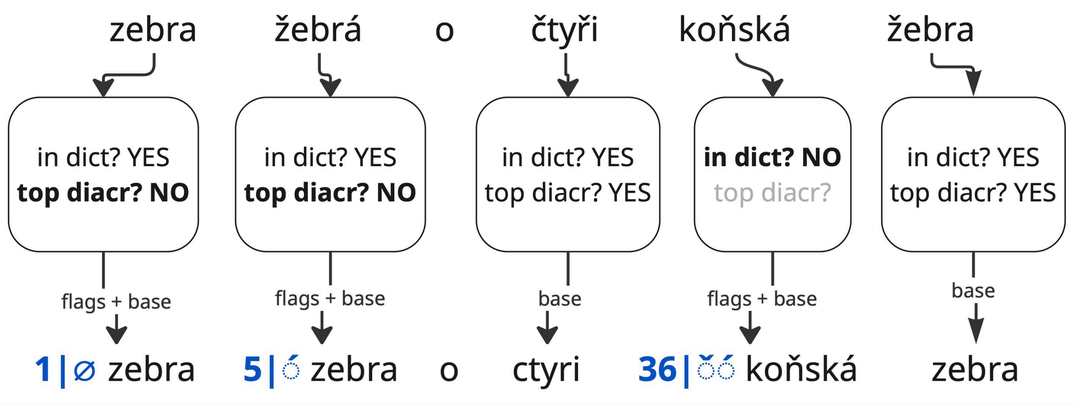
Extrinsic eval:
- most noises: on par with other systems
- fully upper-case: reaches non-noised setup

Preprocessing	Average Unique Token Length ↑	Cased Tokens ↓	Overlap with Uncased ↓
base	6.837	6169	3508
InCa	**7.119**	**4**	**0**
Naive InCa	**7.127**	**3**	**0**
Marian	6.554	2754	1049
TokenMonster	8.573*	149	92

Intrinsic eval:
- not always optimal
- most stable
- most interpretable tokens

InDia: Dictionary-Based Inline Diacritization

same principle, but flags are character-level { ctyri: čtyři, zebra: žebra }



Diacritization - extrinsic eval:
- On-par performance on non-noised data
- Significant improvements on noised texts

	BLEU		
Preprocessing	no noise	fully de-diacritized	20% de-diacritized
base	21.6	9.2	18.6
InDia	**21.7**	17.9	**21.1**
InDia$_{singleflag}$	**21.7**	**18.8**	**21.1**
InDia$_{nodict}$	21.0	**18.4**	20.5

Github link to:
- InFlags python package
- paper
- video presentation

This research was supported by the Czech Science Foundation project 25-16242S and by the Technology Agency of the Czech Republic project TQ12000040 (CZDEMOS4AI). It has been using data and tools provided by the LINDAT/CLARIAH-CZ Research Infrastructure (https://lindat.cz), supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2023062).