

Transliterate from Diagrams & Data

Mahdi Mohajeri & Wuilloud Jair

May 11, 2022



Motivation & Approach

Graphical Approach

- Empower Linguist
- Encapsulate Development
- Code generated from multiple diagrams/files

NNets

- Reduce Dependencies
- Code from Data

"Universal" Strategy/Tools



Strategy

Transliteration Code Creation

- Graphical Editing (lucidchart)
- Software Development on diagram nodes
inclusion of mappings, code snippets, NLP libraries, ...
- Code generation from csv exports
run test, tweaks & re-designs, etc, ...
- Transliteration

Train & Export NNets

- Generate transliteration data
- Train NNets & Export Onnx
- Run Ruby Prod. code



Demo



Resources

Links:

- [ISO Farsi Demo](#)

- [QuickStart](#)

- [Paper](#)



Summary & Outlook

We have presented a diagram/data based solution and discussed its potential for transliteration.

Solutions/Diagrams could be easily shared.
Code built from data for production.

Potential Improvements/Usage:

- alternatives to lucidchart or plugin
- templates for Linguists to use



Discussion / Questions

We are open to suggestions and feedbacks!

■ CODE,

[https://github.com/interscript/
transliteration-learner-from-graphs/blob/main/
docs/article.pdf](https://github.com/interscript/transliteration-learner-from-graphs/blob/main/docs/article.pdf)

