KLPT – Kurdish Language Processing Toolkit

Sina Ahmadi

Insight Centre for Data Analytics
National University of Ireland Galway

https://sinaahmadi.github.io

The 2nd Workshop for Natural Language Processing Open Source Software (NLP-OSS)
EMNLP 2020

November 2020

Table of Contents

1. Introduction

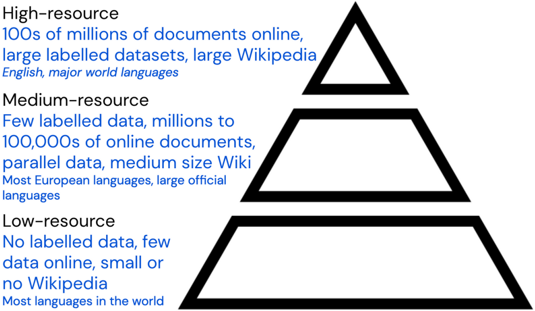
2. Kurdish Language
   - General description
   - Current state of Kurdish language processing (KLP)
   - What is wrong?

3. Kurdish Language Processing Toolkit (KLPT)
   - KLPT Structure
   - Preprocess
   - Transliterate
   - Stem
   - Tokenize

4. Conclusion

Introduction

- 7,117 languages are spoken in the world[1]
- a big proportion of these languages are endangered, minority or **less-resourced**
- recent focus on applying language-independent approaches to various tasks in natural language processing (NLP) and computational linguistics using artificial intelligence
- language-specific tools are still essential to process a language in a viable way



High-resource

100s of millions of documents online, large labelled datasets, large Wikipedia

English, major world languages

Medium-resource

Few labelled data, millions to 100,000s of online documents, parallel data, medium size Wiki

Most European languages, large official languages

Low-resource

No labelled data, few data online, small or no Wikipedia

Most languages in the world

[1] Source: `https://www.ethnologue.com/guides/how-many-languages`

* Image source: `https://ruder.io/unsupervised-cross-lingual-learning`

Table of Contents

1. Introduction

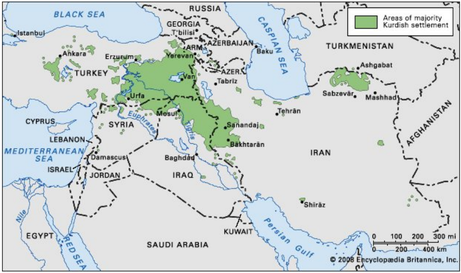
2. Kurdish Language
   - General description
   - Current state of Kurdish language processing (KLP)
   - What is wrong?

3. Kurdish Language Processing Toolkit (KLPT)
   - KLPT Structure
   - Preprocess
   - Transliterate
   - Stem
   - Tokenize

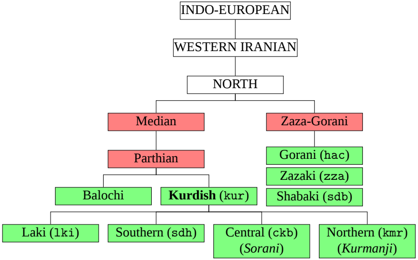
4. Conclusion

Kurdish Language

- an Indo-European language
- spoken by 20-30 million speakers
- spoken in many dialects and subdialects (*dialects* or *languages*?)
- written in many scripts, among which the Latin-based and Arabic-based ones are still widely in use



*



INDO-EUROPEAN

WESTERN IRANIAN

NORTH

Median

Zaza-Gorani

Parthian

Gorani (`hac`)

Zazaki (`zza`)

Shabaki (`sdb`)

Balochi

Kurdish (`kur`)

Laki (`lki`)

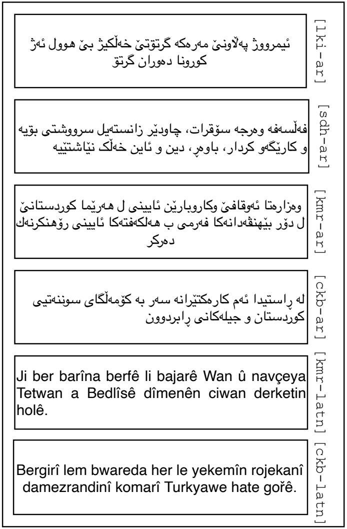
Southern (`sdh`)

Central (`ckb`) (*Sorani*)

Northern (`kmr`) (*Kurmanji*)

Source: `https://www.britannica.com/topic/Kurd`

- using more than one script for a language, not only scatters readers but also creates further challenges in text processing
- written in various orthographies following different conventions
  - *di sala 2020'an* | *2020-an* | *2020an de* "in the year 2020"
  - *hêv**i**ya*, *hêv**î**ya* or *hêvî ya* "hope of"?
  - ٠١٢٣٤٥٦٧٨٩, ۰۱۲۳۴۵۶۷۸۹ or 0123456789?
- although Kurdish orthographies are phonemic, there is not always a one-to-one relation between graphemes, particularly due to:
  - double-usage characters: ی for î/y and و for u/w
  - variations in some orthographies such as l, ll or ł for [ɫ]
  - vowel i has no equivalent in the Arabic-based orthography



ئیمرووژ پەڵاونێ مەرەکە گرتوتێ خەڵکیژ بێ هوول ئەژ
کورونا دەوران گرتو

[lki-ar]

فەڵسەفە وەرجە سۆقرات، چاودێر زانستەیل سرووشتی بۆیە
و کارێگەو کردار، باوەڕ، دین و ئاین خەڵک نیاشتێیە

[sdh-ar]

وەزارەتا ئەوقافێ وکاروبارێن ئایینی ل هەرێما کوردستانێ
ل دۆر بێهنڤەدانەکا فەرمی ب هەلکەفتەکا ئایینی رۆهنکرنەك
دەرکر

[kmr-ar]

لە ڕاستیدا ئەم کارەکتێرانە سەر بە کۆمەڵگای سوننەتیی
کوردستان و جیلەکانی ڕابردوون

[ckb-ar]

Ji ber barîna berfê li bajarê Wan û navçeya Tetwan a Bedlîsê dîmenên ciwan derketin holê.

[kmr-latn]

Bergirî lem bwareda her le yekemîn rojekanî damezrandinî komarî Turkyawe hate gořê.

[ckb-latn]

Kurdish

Current state of Kurdish language processing (KLP)

- the earliest works in the field of KLP date back to 2009
- thus far, a total number of **53** publications are published in a field directly related to KLP (as of August 2020)
- two open-source volunteer-based projects:
  - Kurdish Language Processing Project (KLPP[2]) in 2012
  - Kurdish Basic Language Resource Kit (Kurdish-BLARK[3]) in 2014
- a few number of non-scientific contributions

Open-source

Does the paper provide the discussed resource or tool under an open-source license?

Applicability

Does the paper, implicitly or explicitly, propose an approach or methodology that can be applied to solve the same problem in the other dialects of Kurdish?

[2] http://klpp.github.io/

[3] https://kurdishblark.github.io/

Current state of KLP

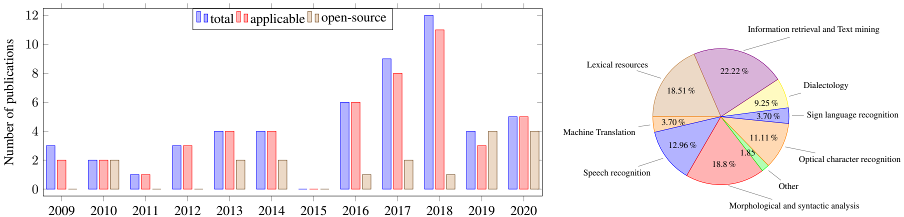
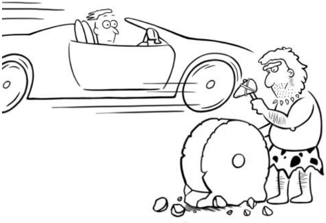


Figure: Number of scientific publications directly related to KLP per year and field

- most of these publications are applicable
- only **18** provide their resources or tools under an open-source license
- among the open-source ones, **11** are outcomes KLPP and Kurdish-BLARK
- Sorani makes up a predominant proportion of almost 90% of publications
- no publication addresses the processing of Southern Kurdish or Laki
- Kurdish still lacks basic language processing tools such as part-of-speech tagger, stemmer, lemmatizer and so on

Current state of KLP: What is wrong?

- Many projects overlap significantly, yet none of them provide a solution under any open-source license
  - Stemming is addressed at least *five* times [Jaff, 2014, Salavati and Ahmadi, 2018, Mustafa and Rashid, 2018, Saeed et al., 2018, Hawezi et al., 2019]



*

- Some are hardly integrable or inter-operable
  - A large-scale morphological lexicon and a part-of-speech tagger for Kurdish within the Alexina framework [Walther and Sagot, 2010, Walther et al., 2010]
- Released in an unorganized manner for individual tasks
  - Example: a transliteration tool for Kurdish [Ahmadi, 2019a]
- Further progress is hindered in the field
- **Kurdish *is* still a less-resourced language**

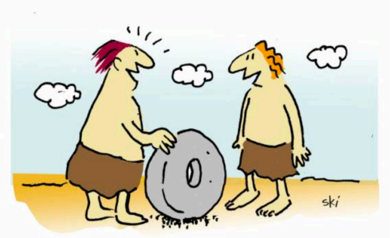
* Image source: `https://www.aic.cuhk.edu.hk/web8/Reinventingthewheel.htm`

Table of Contents

1. Introduction
2. Kurdish Language
   - General description
   - Current state of Kurdish language processing (KLP)
   - What is wrong?
3. Kurdish Language Processing Toolkit (KLPT)
   - KLPT Structure
   - Preprocess
   - Transliterate
   - Stem
   - Tokenize
4. Conclusion

Kurdish Language Processing Toolkit (KLPT)

- a basic but extendable language processing toolkit
- an effort to standardize Kurdish language with all its dialects and scripts
- implemented in Python
- inspired by the functionality of relevant NLP toolkits, e.g. NLTK and spaCy
- no external NLP library is used in this toolkit
- composed of core modules for Sorani and Kurmanji for the following tasks:
  - text preprocessing
  - stemming
  - lemmatization
  - spelling error detection and correction
  - transliteration
  - morphological analyzer and generator
  - tokenization

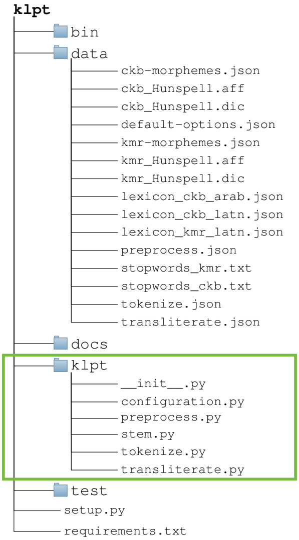


- it is open-source!

  → **https://github.com/sinaahmadi/klpt**

KLPT: Structure

- no hard-coding
- easily extendable for other dialects and tasks
- each package corresponds to a set of related tasks
- composed of four core NLP packages as follows:
    - `preprocess`
    - `transliterate` ([Ahmadi, 2019a])
    - `stem` ([Ahmadi, 2020e])
    - `tokenize` ([Ahmadi, 2020b, Ahmadi, 2020c])



```
klpt
├── bin
├── data
│   ├── ckb-morphemes.json
│   ├── ckb_Hunspell.aff
│   ├── ckb_Hunspell.dic
│   ├── default-options.json
│   ├── kmr-morphemes.json
│   ├── kmr_Hunspell.aff
│   ├── kmr_Hunspell.dic
│   ├── lexicon_ckb_arab.json
│   ├── lexicon_ckb_latn.json
│   ├── lexicon_kmr_latn.json
│   ├── preprocess.json
│   ├── stopwords_kmr.txt
│   ├── stopwords_ckb.txt
│   ├── tokenize.json
│   └── transliterate.json
├── docs
├── klpt
│   ├── __init__.py
│   ├── configuration.py
│   ├── preprocess.py
│   ├── stem.py
│   ├── tokenize.py
│   └── transliterate.py
├── test
├── setup.py
└── requirements.txt
```

KLPT Packages: Preprocess

Goal: Handle diversities in scripts and orthographies in an automatic and formalized way

1. `normalize()`: normalize text by unifying character encodings
   - Example: the grapheme ی (U+06CC, î/y), may be represented as ي (U+064A), ى (U+0649), ﻲ (U+FEF2) or ﻱ (U+FEF1)
2. `standardize()`: standardize scripts and orthographies by using writing conventions based on dialects and scripts
3. `unify_numeral()`: convert Farsi, Eastern and Western Arabic numerals

Example

```
>>> from klpt.preprocess import Preprocess
>>> preprocessor = Preprocess("Sorani", "Arabic", numeral="Latin")
>>> preprocessor.normalize("لە ســـــاڵەکانی ١٩٥٠دا")
لە ساڵەکانی 1950دا
>>> preprocessor.standardize("راستە لەو ووڵاتەدا")
ڕاستە لەو وڵاتەدا
```

KLPT Packages: Transliterate

- transliterating the Arabic-based and Latin-based scripts of Kurdish to one another, e.g. برا → *bira* ‘brother’
- based on the rule-based approach of [Ahmadi, 2019a] which
  - detects double usage characters
  - predicts the presence of the missing i, a.k.a *Bizroke*
  - finds the syllabic pattern of a given word based on Kurdish phonetics
- beneficial to many NLP tasks such as named-entity recognition

Example

```
>>> from klpt.transliterator import Transliterate
>>> transliterator = Transliterate("Kurmanji", "Latin", target_script="Arabic")
>>> transliterator.transliterate("rojhilata navîn")
'رۆژهلاتا ناڤین'
```

KLPT Packages: Stem

- an annotated lexicon + morphological rules using **Hunspell**[4] for:
  - spelling error detection and correction → also usable in text editors such as LibreOffice
  - morphological analyzer and generator
  - stemmer
- a rule-based lemmatization system
- based on [Ahmadi, 2020c, Ahmadi, 2020e]

Example

```
>>> from klpt.stem import Stem
>>> stemmer = Stem("Sorani", "Arabic")
>>> stemmer.check_spelling("سوتاندبووت")
False
>>> stemmer.correct_spelling("سوتاندبووت")
('سووتاندبووت', 'سووتاندت', 'سووتاندن', 'سووتاند')
>>> stemmer.stem("سووتاندبووت")
('سووت',)
>>> stemmer.analyze("دیتبامن")
{'pos': 'verb', 'is': 'past_intransitive', 'stem': 'دی', ‘verb_stem': 'دیت',
'terminal_suffix': 'بامن'}
```

[4] http://hunspell.github.io

KLPT Packages: Tokenize

- detect word and sentence boundaries → a non trivial task:
  - **orthographic inconsistencies**, e.g. how compounds words are separated?
  - **excessive concatenation**, e.g. لەوێشدایە (*lewêşdaye*) “(it) is also there” is written as a word but is composed of five tokens *le*, *wê*, *ş*, *da*, *ye*
- split a text into sentences or tokens
- identify compound forms such as *kar-û-bar* (word-and-load) “affaires”
- based on the [Ahmadi, 2020b]’s approach using a morphological analyzer and a lexicon

Example

```
>>> from klpt.tokenize import Tokenize
# Tokenize module
>>> tokenizer = Tokenize("Kurmanji", "Latin")
>>> tokenizer.word_tokenize("endamên encûmena wezîrên")
['_endam_ên', '_encûmen_a', '_wezîr_ên']
```

Table of Contents

1. Introduction
2. Kurdish Language
   - General description
   - Current state of Kurdish language processing (KLP)
   - What is wrong?
3. Kurdish Language Processing Toolkit (KLPT)
   - KLPT Structure
   - Preprocess
   - Transliterate
   - Stem
   - Tokenize
4. Conclusion

Conclusion

- **Lessons learned**:
  - **release your project under an open source license** → essential to ensure gradual but efficient progress in resource and technology development for a less-resourced language
  - **community-driven initiatives**: bring together users, developers, researchers, language activists and policy makers
    - ★ Vejîn Books (`https://books.vejin.net/en`)
    - ★ Vejîn Dictionaries (`https://lex.vejin.net/en`)
  - **raise awareness** by promoting good practices in content creation on the Web, particularly collaboratively-curated resources such as Wiktionary[5] and Wikipedia[6]
  - every single user is a contributor too
- **Future directions**:
  - promote the usage of KLPT in the Kurdish communities
  - create a community of developers and linguists for KLP
  - extend the current version of KLPT to include further advanced tasks

[5] https://en.wiktionary.org

[6] https://www.wikipedia.org/

Join KLPT

*

https://github.com/sinaahmadi/klpt

* Image credit: Milad Ghaderpanah

References

Sardar Jaf, Allan Ramsay (2014)
Stemmer and a POS tagger for Sorani Kurdish.
6th International Conference on Corpus Linguistics - Spain.

Shahin Salavati and Sina Ahmadi (2018)
Building a Lemmatizer and a Spell-checker for Sorani Kurdish.
arXiv preprint arXiv:1809.10763.

Mustafa, Arazo M., and Tarik A. Rashid. (2018)
Kurdish stemmer pre-processing steps for improving information retrieval
Journal of Information Science, 44.1: 15-27.

Saeed, A. M., Rashid, T. A., Mustafa, A. M., Agha, R. A. A. R., Shamsaldin, A. S., & Al-Salihi, N. K. (2018)
An evaluation of Reber stemmer with longest match stemmer technique in Kurdish Sorani text classification
Iran Journal of Computer Science, 1(2), 99-107.

Hawezi, R. S., Azeez, M. Y., & Qadir, A. A. (2019)
Spell checking algorithm for agglutinative languages Central Kurdish as an example
International Engineering Conference (IEC)(pp. 142-146). IEEE.

Sina Ahmadi (2019)
A Rule-based Kurdish Text Transliteration System
Asian and Low-Resource Language Information Processing (TALLIP) 18(2):18:1–18:8.

Sina Ahmadi (2020)
A Tokenization System for the Kurdish Language
Proceedings of the Seventh Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2020).

Sina Ahmadi (2020)
A Lemmatization System for Sorani Kurdish
Under review.

Sina Ahmadi (2020)
Building a Corpus for the Zaza–Gorani Language Family
Proceedings of the Seventh Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2020).

Sina Ahmadi (2020)
Hunspell for Sorani Kurdish Spell-checking and Morphological Analysis.
Under review.

Walther, G., & Sagot, B. (2010)

Developing a large-scale lexicon for a less-resourced language: General methodology and preliminary experiments on Sorani Kurdish.

7th SaLTMiL Workshop on Creation and use of basic lexical resources for less-resourced languages (LREC 2010 Workshop).

Géraldine Walther, Benoît Sagot, and Karën Fort. (2010)

Fast development of basic NLP tools: Towards a lexicon and a POS tagger for Kurmanji Kurdish

International conference on lexis and grammar.