Sesión 3: EDICTOR

Carlos Ugarte

01.09.2025

1 Introducción a EDICTOR[1]: una herramienta para la comparación lingüística

Por favor, entren al software PyCharm CE y repitan el proceso visto en la anterior clase hasta que su directorio de trabajo sea la carpeta clonada de este taller. Para esto, estando en la carpeta Documents, sigan los siguientes comandos:

```
mkdir proyectos
cd proyectos
mkdir taller
cd taller
git clone https://github.com/MuffinLinwist/taller_pucp25.git
```

La comparación histórica de lenguas con el propósito de identificar cognados y correspondencias de sonido en listas de palabras multilingües que luego se puedan usar para inferir árboles filogenéticos se puede llevar a cabo de manera conveniente en un marco de comparación de lenguas asistida por computadora (ver https://calclab.org), utilizando herramientas para inferencia automática, como LingPy (https://lingpy.org, List y Forkel 2016, y herramientas para anotación y curación de datos, como EDICTOR (https://edictor.org, List 2017).

2 ¿Qué necesitas como input?

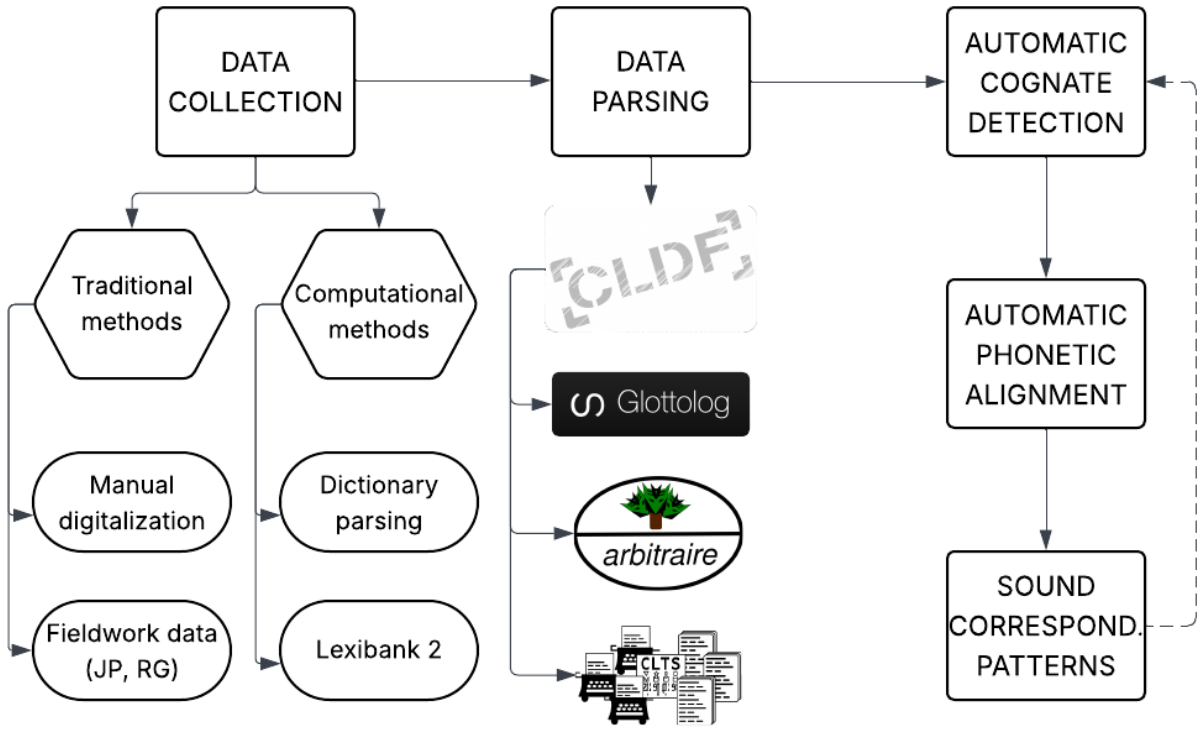
Aquí un ejemplo básico. EDICTOR espera un archivo TSV como input. Esto significa: un archivo que sea tab-separated (separado con tabs) y contenga una linea de cabecera indicando el contenido de cada columna individual. La primera columna debe proporcionar los identificadores numéricos de todas las filas de los datos. Cada fila corresponde a una palabra. Una de las columnas restantes debe llamarse DOCULECT y contener el nombre de cada idioma de la muestra. Otra columna debe llamarse CONCEPT y contener los conceptos (o glosas) de cada palabra. La forma de la palabra debe proporcionarse segmentada (el espacio se utiliza para segmentar los sonidos individuales) en la columna TOKENS.

[1] El material explicativo para esta sesión está totalmente basado en los tutoriales encontrados en la página web de EDICTOR (https://edictor.org, viñeta Help), en los del blog "Computer-Assisted Language Comparison in Practice" (https://calc.hypotheses.org/?s=edictor) así como en experiencia propia trabajando con la interface.

Visto como archivo TSV, es algo así:

ID	DOCULECT	CONCEPT	VALUE	FORM	TOKENS	BORROWING	COGID
3631	East_Futuna	above	à/luga/	luga	l u g a	0	1382
284	Wallisian	above	'o/luga/	luga	l u g a	1	1382
5391	Futuna*Aniwa*	above	weihlunga	weihlunga	w e i + ʰl u ŋ a	0	1382
761	Maori	above	i runga	i runga	i r u ŋ a	0	1382
3332	North_Marquesan	above	'una	'una	ʔ u n a	0	1382
4214	Mele-Fila	all	euči	euči	e u tʃ i	0	1115
3917	Pukapuka	all	katoa(toa)	katoa	k a + t o a	0	293
560	Proto-Polynesian	yellow	*reŋareŋa*, felo(-felo)	*reŋareŋa*	r e ŋ a + r e ŋ a	0	162
561	Proto-Polynesian	yellow	reŋareŋa, *felo(-felo)*	felo	f e l o	0	230

El workflow automático con Lingpy y CLDF, luce algo así:



3 Montando EDICTOR en tu computadora

Para comenzar, vamos a tener que instalar EDICTOR localmente en nuestras computadoras. Para ello, se necesita instalar una serie de paquetes de Python e implementar un workflow. Sin embargo, Windows es particularmente malo para hacer esto debido a ciertas peculiaridades de los sistemas operativos Windows que tienden a arrojar errores difíciles de entender al seguir las instrucciones de instalación tradicionales orientadas a usuarios con sistemas operativos Unix (para más referencia sobre esto, dirigirse a: https://calc.hypotheses.org/7852). Como trabajamos ahorita con Windows, haremos el set-up para tal sistema operativo (pero los tutoriales en línea presuponen el uso de Linux o Mac, por lo que también pueden referirse a ellos, en caso trabajen con estos sistemas operativos.) Para eso, ejecutaremos los siguientes comandos

en nuestras terminales:

```
python -m pip install virtualenv
virtualenv tallervenv
Set-ExecutionPolicy -ExecutionPolicy Unrestricted -force
.\tallervenv\Scripts\activate
```

El estándar en este tipo de trabajos es siempre partir desde un fresh venv para almacenar todos y únicamente los paquetes que son indispensable para correr las funciones que nuestro proyecto emplea y evitar colisiones entre paquetes. Los comandos de arriba crean un venv llamado TALLERVENV y lo activan. Ahora descargamos en nuestro venv los paquetes necesarios para correr EDICTOR con los siguientes comandos:

```
python -m pip install pylexibank
python -m pip install edictor
python -m pip install lingpy
python -m pip install lingrex
python -m pip install lexibase
```

Ahora descargamos el dataset que vamos a utilizar. Para eso, retrocederemos al folder de proyectos y clonamos el dataset NorthPeruLex[2] y nos metemos en la carpeta EDICTOR-REMOTE en él:

```
cd ../
git clone https://github.com/lexibank/northperulex.git
cd northperulex/edictor-remote
```

Una vez con la data ya trabajada en nuestra computadora, y con nosotros teniendo el folder con esa data en el working directory, generamos el archivo que va a ser el input para edictor con el siguiente comando:

```
edictor wordlist --name=northperulex --preprocessing=edictor-remote/to_edictor.py --
```

Generado este archivo, abrimos el servidor de EDICTOR desde nuestra línea de comando con el siguiente comando:

```
edictor server
```

Este comando abre EDICTOR en Firefox. Ahora abrimos la ventana FILES, donde subiremos el archivo output con el nombre NORTHPERULEX.TSV.

[2]También pueden descargar cualquier dataset de Lexibank (https://github.com/lexibank/) y hacer lo mismo cambiando el nombre por el del otro dataset.