

Okazuje się, jeśli tylko wierzyć pewnej popularnej stronie, że mechanizmy predykcji tekstu powstały z myślą właśnie o niepełnosprawnych ruchowo, żeby przyspieszyć im pisanie:

<http://editlib.org/p/52681/>

<http://aclweb.org/anthology//W/W10/W10-0403.pdf>

Open source'owy korektor pisowni z przeznaczeniem specjalnie dla języka polskiego i dostępny bezpośrednio w pythonie, jeśli nawet istnieje to mało prawdopodobne żeby udało się go kiedyś znaleźć. Najbardziej rozpowszechnionym, popularnym i uznanym z open source'owych narzędzi wydaje się być używany przez Open Office'a i Mozillę MySpell i jego następcę Hunspell. Pythonowy binding i inna nakładka:

<https://packages.debian.org/pl/sid/python3-hunspell>

<https://pypi.python.org/pypi/spell4py> (choć to już chyba nieaktualne bo pod ich adresem jest teraz strona dla ortodontów)

Jeśli chodzi o praktyczne pythonowe wykorzystanie to najczęściej można natrafić na bibliotekę PyEnchant, opartą na Enchant. Właściwie jest to coś bardziej jakby prosta nakładka, umożliwiająca łatwą obsługę poprzez pythona, a mogąca wykorzystywać różne słowniki i różne backendowe biblioteki sprawdzające pisownię, w tym także My i Hunspella.

PyEnchant w działaniu:

```
import enchant
```

```
> broker = enchant.Broker()
```

```
> broker.describe()
```

```
    [<Enchant: Aspell Provider>, <Enchant: Hspell Provider>, <Enchant: Myspell  
    Provider>,<Enchant: Ispell Provider>]
```

```
> broker.request_dict('pl_PL').provider
```

```
    <Enchant: Myspell Provider>
```

```
> dictionary = enchant.Dict('pl_PL')
```

```
> dictionary.check('łyżka')
```

```
    True
```

```
> dictionary.check('łyżka')
```

```
    False
```

```
> dictionary.check('ugrzeźłybyśmy')
```

```
    True
```

```
> dictionary.suggest('łyżka')
```

```
    ['łyka', 'łyczka', 'łezka', 'łyska', 'łydka', 'łózka', 'łyżka', 'kryzka']
```

```
> dictionary.suggest('łyżk')
```

```
    ['łyż', 'łyk', 'łyżka', 'łyżki', 'łyżko', 'łyżek', 'łyżką', 'łyżkę', 'łyża', 'łyżo', 'łyże', 'łyżw', 'łyży',  
    'łysk', 'łyżą']
```

```
> dictionary.suggest('bura')
```

```
    ['baru', 'Bura', 'nura', 'buta', 'burs', 'bur', 'biura', 'burza', 'buran', 'buras', 'bursa', 'burta', 'burak',  
    'burka', 'burda']
```

```
> dictionary.suggest('ygrzeźnięto')
```

```
    ['ugrzeźnięto', 'grzeźnięto', 'y grzeźnięto', 'grzeźnijmy']
```

```
> dictionary.suggest('prze')
['orze', 'pre', 'parze', 'porze', 'oprze', 'perze', 'przez', 'przen', 'pyrze', 'przed', 'przeł', 'uprze',
'przej', 'przeć', 'przeń']
```

Korektor w dokumentach Google działa mniej więcej podobnie, ma także funkcję predykcji kontekstowej ale oczywiście tylko dla j. angielskiego.

Jest też czysto pythonowy Whoosh z funkcją sprawdzania pisowni ale to głównie do innych zastosowań:

<https://pypi.python.org/pypi/Whoosh/>

Dla GTK jest już gotowy pakiet, robiący wszystkie te rzeczy, oparty o Enchant:

<https://github.com/koehlma/pygtkspellcheck>

Użycie (podpowiedzi prawym przyciskiem):

```
import sys
import locale
from gi.repository import Gtk as gtk
from gtkspellcheck import SpellChecker

def quit(*args):
    gtk.main_quit()

window = gtk.Window.new(gtk.WindowType.TOPLEVEL)
view = gtk.TextView.new()
spellchecker = SpellChecker(view, 'pl')
window.set_default_size(600, 400)
window.add(view)
window.show_all()
window.connect('delete-event', quit)
gtk.main()
```

A więc można to pewnie zrobić w Clutterze na zasadzie podobnej jak to robi ten skrypt:

<https://github.com/koehlma/pygtkspellcheck/blob/master/src/gtkspellcheck/spellcheck.py>

jeśli by połączyć PyEnchant z jakimś silnikiem uczenia się i sortowania podpowiedzi dałoby się też zrobić i predykcję i korekcję, pytanie tylko lepszą czy gorszą.

Na koniec krótka opinia eksperta na ten temat:

Savoir vivre, poprawianie błędów językowych

Czy poprawiać cudze błędy językowe? Do szału doprowadza mnie np. mówienie "dwutysięczny jedenasty" zamiast "dwa tysiące jedenasty" - czy jednak wypada, żebym zwracał innym uwagę. Poprawianie innych, jeżeli w ogóle zdecydujemy się na taki krok, wymaga nie tylko

wiedzy i doświadczenia, ale i dużo taktu. Oczywiście inny wydzźwięk będzie miało nakierowanie studenta przez profesora podczas zajęć, gdy ten odczytuje swój esej napisany w obcym języku, a inny - zwrócenie uwagi przez kolegę z pracy podczas cotygodniowego zebrania, że nie mówi się "włancza", a "włącza" - ku uciesze wszystkich zgromadzonych. Jedną z ważniejszych zasad dobrych manier jest, by poprzez swoje zachowanie nie powodować dyskomfortu u innych. Nieumiejętne zwrócenie uwagi dotyczące błędów językowych może sprawić komuś dużo przykrości, tym bardziej w dodatkowo stresujących okolicznościach. Innym aspektem jest to, kto komu może zwracać uwagę. W pracy przełożony podwładnemu - jak najbardziej, a podwładny przełożonemu? Czemu nie, w końcu niektórzy podwładni od tego są, żeby doradzać. Starszy młodszemu - jak najbardziej, ale czy młodszy starszemu? Zależy czego będzie dotyczyła ta uwaga i w jaki sposób zostanie wypowiedziana. Dlatego jeżeli faktycznie jesteśmy przekonani, że na słodki owoc cytrusowy mówimy "pomarańcza", a nie "pomarańcz", powiedzmy o tym dyskretnie i delikatnie, stosując się tym samym do starego biznesowego powiedzenia: "Chwal przy wszystkich, zwracaj uwagę w cztery oczy".

Na pytania odpowiada Adam Jarczyński, dyrektor generalny Polskiej Akademii Protokołu i Etykiety.