Bigorna

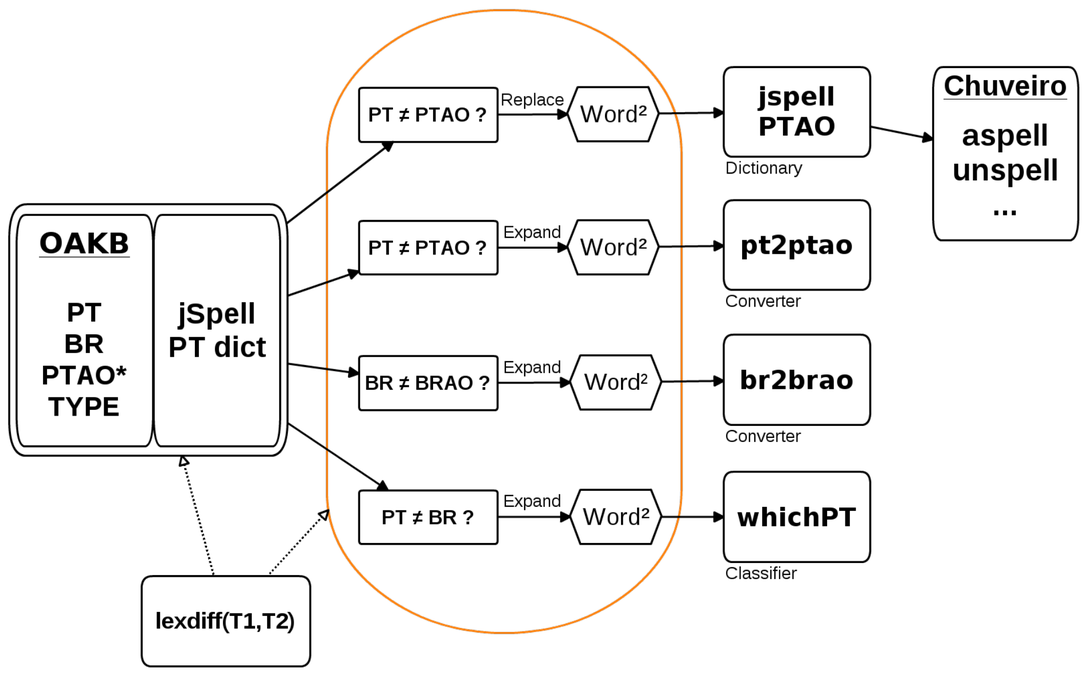
A Toolkit for Orthography Migration Challenges

José João Almeida, André Santos, Alberto Simões

Abstract

Languages are born, evolve and, eventually, die. During this evolution their spelling rules (and sometimes the syntactic and semantic ones) change, putting old documents out of use. In Portugal, a pair of political agreements with Brazil forced relevant changes on the way the Portuguese language is written, the most recent one being the Portuguese Language Orthographic Agreement (PLOA) signed in 1990.
Bigorna is a toolkit for the classification of language variants, their comparison and the conversion of texts in different language versions. As Bigorna relies on a set of conversion rules we will also discuss how to infer conversion rules from a set of documents (texts with different ages).

Contents



1 Compiling OAKB

A table containing all the information about the word changes. This table was built based on previously existing resources and proved to be crucial to the subsequent tasks performed.

OAKB structure

```
oakb = entry*
entry =
  pt_pt            : word
  pt_br            : word
  pt_oa            : word*
  preferencial_pt  : word
  preferencial_br  : word
  type     : Capit | Hyphen | Accent | Normal | Excep
```

```
adenóide :: adenóide :: adenoide :: adenoide :: adenoide :: Accent
adjecção :: adjeção  :: adjeção  :: adjeção  :: adjeção  :: Normal
Março    :: março    :: março    :: março    :: março    :: Capit
```

2 Updating the dictionary vocabulary

An existing European Portuguese spellchecker dictionary (jSpell) was updated. This dictionary was later used to generate lists to both the language conversion tools and the language classifier.

From the 2 600 words in `OAKB`, just 960 were related directly with a lemma in jSpell's dictionary. From these 960 lemmas jSpell generates a total of 11 500 words.

Update function

```
Function newdic(oakdb,dicjs)
  for ( x ∈ dom(oakb) ∧ oakb[x].type = normal
       ∧ x ≠ oakb[x].preferpt ∧ x ∈ dom(dicjs))
    {
      neww ← oakb[x].preferpt
      dicjs[neww] ← dicjs[x]
      delete dicjs[x]
    }
```

3 Language conversion tools

As many texts will need to be updated to the new spelling form, there was the need to create automated conversion tools. Due to the multiple spelling cases, two versions were created: an European Portuguese converter and a Brazilian Portuguese one.

Conversion examples

```
$ pt2ptao
A adopção do acordo implica a actualização de ferramentas.
              ↓
A adoção do acordo implica a atualização de ferramentas.
```

Conversion examples

```
$ br2brao
Ele fez um vôo rasante sobre a aréia.
              ↓
Ele fez um voo rasante sobre a areia.
```

4 Variant classifier tool

After creating the language converters became clear the need to have a language classifier, capable of detecting the variant of Portuguese in which a given text was written, allowing to automatically differentiate texts (possibly to further conversion).

To build this tool two lists were generated: one with European Portuguese-only words and another with Brazilian Portuguese ones.

Calculating the lists

```
Function calc_whichpt_lsts(dicpt,dicbr,oakb)
  for ( x ∈ dom(oakb)
     ∧ oakb[x].type = (normal or accent)
     ∧ oakb[x].pt_pt ≠ oakb[x].pt_br
     ∧ oakb[x].pt_pt ∈ dom(dicpt)
     ∧ oakb[x].pt_br ∈ dom(dicbr))
   {
    wpt ← oakb[x].pt_pt
    wbr ← oakb[x].pt_br
    justpt ← justpt ∪ {x ∈ deriv(wpt,dicpt)| x ∉ dicbr}
    justbr ← justbr ∪ {x ∈ deriv(wbr,dicbr)| x ∉ dicpt}
   }
```

Language classifier definition

```
Function classify_pt(text)
  for ( x ← text )
     if ( x ∈ justpt ) PTcount++
     if ( x ∈ justbr ) BRcount++
   compare ( PTcount, BRcount)
```

5 Lexical comparison tools

There are other situations where there are no available lists of words, only documents with different orthographic versions.

`lexdiff`, is able to compare two versions of a text with different spelling and detect (linguistic) differences. This may be used to help building tools (as the previously mentioned).

Lexdiff example: word level changes

```
$ lexdiff -wa AmPerd.ptBR AmPerd.ptPT
   32 acadêmico → académico
   14 idéia     → ideia
   12 redargüiu → redarguiu
    7 gênio     → génio
    4 refletiu  → reflectiu
    ...
```

Lexdiff example: char level changes

```
$ lexdiff -cctx AmPerd.ptBR AmPerd.ptPT
 changed  PT→BR (unchanged)    changed   BR→PT (unchan) Concl

!   36   ect→et  (9)              36    et →ect  (206)  BR →?PT
!   34   dém→dêm (1)        !!    34    dêm→dém         BR?↔ PT
    18   dei→déi (164)      !!    18    déi→dei         BR → PT
    17   gui→güi (88)       !!    17    güi→gui         BR → PT
    15   que→qüe (2417)     !!    15    qüe→que         BR → PT
!!  11   gén→gên            !     11    gên→gén  (6)    BR → PT
!!  9    món→môn            !!    9     môn→món         BR ↔ PT
!   8    act→at  (1)              8     at →act  (456)  BR ← PT
!!  7    ecç→eç                   7     eç →ecç  (77)   BR ← PT
!!  6    acç→aç                   6     aç →acç  (431)  BR ← PT
!!  6    tón→tôn            !!    6     tôn→tón         BR ↔ PT
```

Confusion matrix pt-br → pt-pt

```
   et   → {   et  → 206, ect  → 36 },
   déi  → {  dei  → 18 },
   güi  → {  gui  → 17 },
   at   → {   at  → 456, act  → 8  apt  → 1, apt  → 1},
   eç   → {   eç  → 77,  ecç  → 7, eaç  → 2, epç  → 2},
```