

Pandora

A (language independent) Tagger Lemmatizer for Latin and the Vernacular

Mike Kestemont, Jean-Baptiste Camps, Thibault Clérice,
Enrique Manjavacas

Université d'Anvers / École nationale des chartes

COSME – 6 novembre 2017

Plan

Ouvrir la boîte (ou l'amphore)

- Pandora : pourquoi faire ?

- Lemmatisation et apprentissage profond

Sous le couvercle

- Vecteurs de mots

- Réseaux de neurones

Pandora en action

- Un rapide guide d'utilisation de Pandora

- Résultats provisoires

 - Corpora en ancien français et occitan

 - Corpora en d'autres vernaculaires ?

 - Corpora latins

L'avenir de Pandora

- Plus de données !

- La pandora postcorrect app

Plan

Ouvrir la boîte (ou l'amphore)

- Pandora : pourquoi faire ?

- Lemmatisation et apprentissage profond

Sous le couvercle

- Vecteurs de mots

- Réseaux de neurones

Pandora en action

- Un rapide guide d'utilisation de Pandora

- Résultats provisoires

 - Corpora en ancien français et occitan

 - Corpora en d'autres vernaculaires ?

 - Corpora latins

L'avenir de Pandora

- Plus de données !

- La pandora postcorrect app

Problèmes spécifiques des langues anciennes (*a fortiori* vernaculaires)

Problèmes de fond :

- ▶ faiblesse (ou pluralité) de la norme graphique ;
- ▶ importance de la variation diatopique, diachronique, ...

Problèmes conjoncturels :

- ▶ outils conçus pour les langues modernes peu adaptés ;
- ▶ rareté des ressources ;
- ▶ petite dimension des corpus.

Ex. : graphies concurrentes de “cheval” au sein d’un même corpus

forme	occurrences	forme	occurrences
cheval	3907	chiuaus	11
chevaus	460	ceuaus	9
cheual	377	chevail	9
ceval	339	chiuau	9
chevals	186	chivals	8
cevaus	93	chevau	7
chival	70	kevaus	6
ceual	66	chavaus	3
cheuaus	65	cheuas	2
cevals	28	keval	2
chaval	27	cheua	1
chivaus	27	cheuau	1
chiual	23	cheva	1
chevas	20	chiuals	1
cheuals	14		

En conséquent...

Abondance de mots inconnus...

1. Loi de Zipf
2. forte variation graphique
3. variation flexionnelle

→ tout texte inconnu du corpus d'entraînement va présenter un nombre important de formes inconnues du lemmatiseur.

Fréquence des lemmes ou formes homographes

“son”, est-ce ?

- ▶ son1 (<SUMMUS), *sommet, extrémité* ;
- ▶ son2 (<SECUNDUM), *selon* ;
- ▶ son3 (<SONUS), *son, émission sonore* ;
- ▶ son4 (<SUUS), *possessif*.

Voire une forme de *sēon* (résidu du grain), du vb. *estre1*, etc.

L'approche de Pandora : l'apprentissage profond

- ▶ apprentissage en contexte sur des corpus annotés ;
- ▶ pas de lexique, ni de règles prédéfinies ;
- ▶ représentation sémantique via des vecteurs de mots ;
- ▶ réseaux de neurones (convolutifs ou récurrents LSTM).

Un lemmatiseur : plusieurs approches

Lemmes

- ▶ `label` : les lemmes connus du lemmatiseur sont ceux rencontrés dans le corpus d'entraînement, et il ne peut en créer d'autres ;
- ▶ `generate` : le réseau de neurone tente d'apprendre comment recréer les lemmes, et peut donc proposer des lemmes nouveaux qui ne figurent pas dans le corpus d'entraînement (*experimental*).

POS

- ▶ `label` seul (par ex. `PRE`, `DETdef`, etc.).

morphologie

- ▶ `label` : un ensemble donné d'étiquettes de flexion est pris comme un seul label (par ex., ``NOMB.=s | GENRE=m | CAS=n'`) ;
- ▶ `multilabel` : les étiquettes complexes sont décomposées en une série d'étiquettes (``NOMB.=s'`, ``GENRE=m'` et ``CAS=n'`).

Plan

Ouvrir la boîte (ou l'amphore)

Pandora : pourquoi faire ?

Lemmatisation et apprentissage profond

Sous le couvercle

Vecteurs de mots

Réseaux de neurones

Pandora en action

Un rapide guide d'utilisation de Pandora

Résultats provisoires

Corpora en ancien français et occitan

Corpora en d'autres vernaculaires ?

Corpora latins

L'avenir de Pandora

Plus de données !

La pandora postcorrect app

Les vecteurs de mots

Problématique :

- ▶ Comment représenter un mot ?
- ▶ Comment désambiguïser des formes homographes ?

Un mot a-t-il un sens en isolation ou uniquement dans un contexte (*sémantique distributionnelle*) ?

Représentation des mots dans un espace sémantique en fonction de leurs cooccurents.

w2vec et l'approche "skipgram"

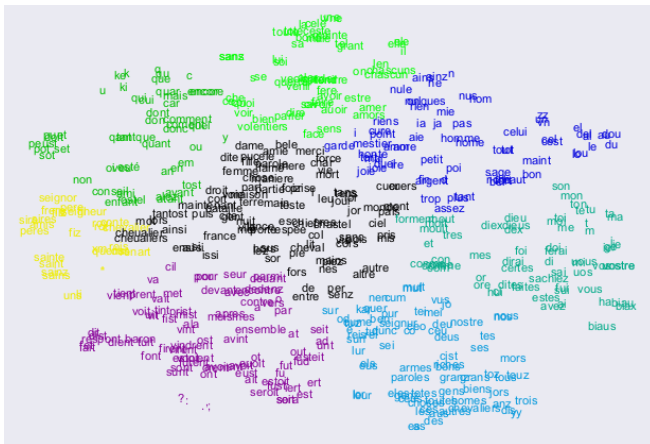
(Mikolov *et al.*, 2013)

Implémentation de w2vec utilisée : celle fournie par le module gensim.

Calcul de vecteurs pour chaque mot, en fonction du contexte, dans un espace de dimension n (par ex., 100 ou 150).

Que capturent ces vecteurs ?

Visualiser les vecteurs de mots en 2 dimensions



Visualisation en deux dimensions (algorithme *t-distributed stochastic neighbor embedding* appliqué aux vecteurs en 100 dimensions de l'analyse w2vec du *Nouveau corpus d'Amsterdam* (3 M de mots)

Calculs de similarité (proximité)

```
>>> model.wv.most_similar(positive=['cheualier'])  
[('cheualiers', 0.7197344899177551), ('chevalier',  
0.6893125176429749), ('uallet', 0.6829389929771423),  
('bacheler', 0.6448491215705872), ('vilain',  
0.6261616945266724), ('prodome', 0.6113272309303284),  
('preudome', 0.5974007844924927), ('vallet',  
0.5890029072761536), ('uallez', 0.5840432643890381),  
('ior', 0.5809659957885742)]
```

```
>>> model.wv.most_similar(positive=['dame'])  
[('damoisele', 0.7411099672317505), ('pucele',  
0.6799402236938477), ('amie', 0.6792483329772949),  
('suer', 0.6707947254180908), ('roine', 0.6304073333740234),  
('lasse', 0.6262991428375244), ('vielle', 0.6252794861793518),  
('damme', 0.6095970273017883), ('fille', 0.600243866443634),  
('chiere', 0.5854043960571289)]
```

```
>>> model.wv.most_similar(positive=['sui'])  
[('suis', 0.8027263283729553), ('serai', 0.7987678647041321),  
('fui', 0.7904558181762695), ('estoie', 0.7331922054290771),  
('fusse', 0.7213727235794067), ('seroie', 0.7185894846916199),  
('suj', 0.6672846078872681), ('voi', 0.6305763721466064),  
('estes', 0.6271989941596985), ('aim', 0.6120427846908569)]
```

(Avec des noms propres)

```
>>> model.wv.most_similar(positive=['rollant'])  
[('aoi', 0.9007091522216797), ('oliver', 0.8698867559432983),  
 ('carles', 0.8476314544677734), ('guenes', 0.8400073051452637),  
 ('capaneus', 0.826859176158905), ('saul', 0.8180047869682312),  
 ('jonathas', 0.805569589138031), ('willame', 0.8026257753372192),  
 ('david', 0.794212818145752), ('idunc', 0.7886440753936768)]
```

Excursus : Opérations un peu plus avancées

roi – home + dame

```
>>> model.wv.most_similar(positive=['roi', 'dame'],  
                           negative=['home'])  
[('roine', 0.6189830303192139), ('roïne', 0.5859614610671997), ...]
```

mon – je + tu

```
>>> model.wv.most_similar(positive=['mon', 'tu'],  
                           negative=['je'])  
[('ton', 0.7054509520530701), ('ten', 0.4551343321800232),  
 ('tun', 0.44703787565231323), ('toi', 0.42671072483062744), ...]
```

suis – estre + dire

```
>>> model.wv.most_similar(positive=['sui', 'dire'],  
                           negative=['estre'])  
[('dirai', 0.5785393714904785), ('di', 0.5722167491912842), ...]
```

Bilan d'étape

Des vecteurs qui capturent

- ▶ de la variation graphique (logiquement, des variantes graphiques du même lemme vont tendre à apparaître dans le même contexte...);
- ▶ pouvant aller jusqu'à modéliser la variation dialectale (cf. *ton* vs. *ten*, *tun*);
- ▶ de l'information sémantique;
- ▶ de l'information morpho-syntaxique ou flexionnelle.

Utilisés dans Pandora pour

- ▶ fournir la représentation initiale (les poids) affectés à chaque forme dans l'initialisation des réseaux de neurones représentant le contexte;
- ▶ être optimisés au fur et à mesure des itérations de l'apprentissage.

Des réseaux de neurones

- ▶ convolutifs ou récurrents (LSTM) [*experimental*];

pour représenter

- ▶ les occurrences (représentation de niveau mot);
- ▶ leur graphie (représentation de niveau caractère);
- ▶ le contexte (avant et après).

Implémentation : Keras (API) sur TensorFlow ou PyTorch (*experimental*).

Un réseau convolutif pour représenter la variation graphique

Application d'un réseau convolutif à la suite de caractère qui est représentée par chaque mot.

Objectif : “apprendre” des caractéristiques (*features*) de plus haut niveau dans ces séquences, qui pourront ensuite être mobilisées pour identifier des variations graphiques courantes (par ex. «ch» / «c» / «k»).

Concrètement, chaque mot est représenté comme une suite de vecteurs (un par caractère), et, pour chaque vecteur, l'index du caractère en question est de 1, celui des autres de 0.

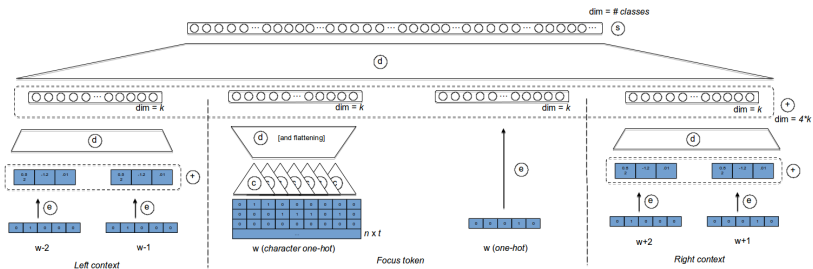
Avantages

Par rapport à une approche où chaque mot serait globalement représenté par un seul vecteur correspondant à l'indexation des formes connues, cette approche a plusieurs avantages :

- ▶ moindre dimensionnalité (nombre de modalités = nombre de caractères différents, pas de mots) ;
- ▶ possibilité de représenter des formes inconnues au moment de l'apprentissage (la forme est inconnue, mais les caractères le sont !).

À partir de là, le réseau convolutif “glisse” sur chaque mot pour en apprendre les caractéristiques, les motifs.

Architecture



Plan

Ouvrir la boîte (ou l'amphore)

Pandora : pourquoi faire ?

Lemmatisation et apprentissage profond

Sous le couvercle

Vecteurs de mots

Réseaux de neurones

Pandora en action

Un rapide guide d'utilisation de Pandora

Résultats provisoires

Corpora en ancien français et occitan

Corpora en d'autres vernaculaires ?

Corpora latins

L'avenir de Pandora

Plus de données !

La pandora postcorrect app

La préparation des données : format

Format tabulaire (tsv) à 4 colonnes.

token	lemma	POS	morph
virent	vëoir	VERcjg	ind psp 3 p
venir	venir	VERinf	
alemans	alemant	NOMpro	p m r
e	et	CONcoo	
baivers	baiver	NOMpro	p m r
e	et	CONcoo	
loerengs	loerenc	NOMpro	p m r
cels	cel	PROdem	p m r
as	a3+le	PRE.DETdef	p m r
curages	corage	NOMcom	p m r
fiers	fier	ADJqua	p m r

Répartis dans trois dossiers : train, dev et test.

Le fichier de configuration

```
[global]
nb_encoding_layers = 2
nb_dense_dims = 2000
char_embed_dim = 50
batch_size = 50
nb_left_tokens = 2
nb_right_tokens = 1
nb_embedding_dims = 100
model_dir = models/geste
postcorrect = False
include_token = True
include_context = True
include_lemma = label
include_pos = True
include_morph = label
include_dev = True
include_test = True
nb_filters = 100
min_token_freq_emb = 10
filter_length = 3
focus_repr = convolutions
dropout_level = 0.15
nb_epochs = 150
halve_lr_at = 75
max_token_len = False
max_lemma_len = False
min_lem_cnt = 1
model = Keras
```

Commandes principales

Entraînement

```
python train.py config_geste.txt --train data/geste/train/  
                                --dev data/geste/dev/  
                                --test data/geste/test/
```

Annotation de nouvelles données

```
python3 tagger.py models/geste --tokenized_input  
    --input data/geste/unseen/  
    -output data/geste/tagged/
```

Un entraînement en cours

```
-> Epoch 1 ...
Epoch 1/1
24845/24845 [=====] - 53s - loss: 11.3096 - lemma_out_loss: 5.5634 - pos_out_loss: 2.3781 - morph_out_loss: 1.3675
::: Train Scores (lemma) :::
+ all acc: 0.37593077077882875
+ kno acc: 0.37593077077882875
+ unk acc: 0.0
::: Dev Scores (lemma) :::
+ all acc: 0.35721200387221685
+ kno acc: 0.4165088981446422
+ unk acc: 0.015283842794759825
::: Test scores (lemma) :::
+ all acc: 0.36694587628865977
+ kno acc: 0.4227188081936685
+ unk acc: 0.00954653937947494
::: Train scores (pos) :::
+ all acc: 0.5799154759508955
+ kno acc: 0.5799154759508955
+ unk acc: 0.0
::: Dev scores (pos) :::
+ all acc: 0.5543723781865117
+ kno acc: 0.5925785687239682
+ unk acc: 0.33406113537117904
::: Test scores (pos) :::
+ all acc: 0.5682989690721649
+ kno acc: 0.6011173184357542
+ unk acc: 0.35799522673031026
::: Train scores (morph) :::
+ all acc: 0.4238277319380157
+ kno acc: 0.4238277319380157
+ unk acc: 0.0
::: Dev scores (morph) :::
+ all acc: 0.13068731848983542
+ kno acc: 0.13517606967057932
+ unk acc: 0.10480349344978165
::: Test scores (morph) :::
+ all acc: 0.40496134020618557
+ kno acc: 0.4324022346368715
+ unk acc: 0.22911694510739858
```


Évaluation

Pour tous les corpus dont il sera question, sur l'ensemble des données disponibles :

- ▶ 80% ont servi à l'entraînement (`train`);
- ▶ 10% ont servi aux ajustements du modèle (`dev`);
- ▶ 10% ont servi au test (`test`);

On distinguera à chaque fois :

- ▶ l'ensemble des formes (`all`);
- ▶ les formes connues, *i.e.* rencontrées durant l'entraînement (`kno`);
- ▶ les formes jamais rencontrées durant l'entraînement (`unk`);

Résultats provisoires pour l'ancien français et l'occitan

Lemmes

Corp.	Dim.	trainAll	devAll	devKno	devUnk	testAll	testKno	testUnk
Chrest	200k	98,47	94,90	96,23	54,46	94,13	95,79	46,64
Geste	50k	99,22	89,42	93,59	48,23	88,55	93,35	46,55
NCA	3M							
Montf	50k	98,96	91,52	96,23	38,17	88,08	94,61	34,66

Chrest : œuvres de Chrétien de Troyes lemmatisées (Pierre Kunstmann).

Geste : corpus de chanson de geste (J.B. Camps)

NCA : Nouveau corpus d'Amsterdam (A. Dees, A. Stein, M.D. Glessgen).

Montferrand : Comptes des consuls de Montferrand, éd. Lodge (annot. JBC, Gilles Couffignal).

Résultats provisoires pour l'ancien français et l'occitan

POS

Corp.	Dim.	trainAll	devAll	devKno	devUnk	testAll	testKno	testUnk
Chrest	200k							
Geste	50k	99,90	91,07	92,46	77,42	90,63	92,23	76,66
NCA	3M	97,58	95,84	96,17	82,28	95,75	96,07	82,39
Montf	50k							

Morph

Corp.	Dim.	trainAll	devAll	devKno	devUnk	testAll	testKno	testUnk
Chrest	200k							
Geste	50k	81,02	9,92	10,03	8,76	67,83	69,84	50,28
NCA	3M							
Montf	50k							

Corpus latin : *Capitularia*

	train		dev		test		
	all	all	kno	unk	all	kno	unk
Lemma	95.08	93.54	95.73	53.25	93.16	95.74	50.58
PoS	95.14	94.16	95.03	78.04	93.97	95.14	74.81

Capitularia (Eger et al, 2015), 400k tokens.

Plan

Ouvrir la boîte (ou l'amphore)

Pandora : pourquoi faire ?

Lemmatisation et apprentissage profond

Sous le couvercle

Vecteurs de mots

Réseaux de neurones

Pandora en action

Un rapide guide d'utilisation de Pandora

Résultats provisoires

Corpora en ancien français et occitan

Corpora en d'autres vernaculaires ?

Corpora latins

L'avenir de Pandora

Plus de données !

La pandora postcorrect app

Pistes d'amélioration

- ▶ gestion fine des différents paramètres pour les adapter à une situation ;
- ▶ nouvelles stratégies d'apprentissage ?
- ▶ implémentation possible de différents environnements d'apprentissage profond, notamment PyTorch (Enrique Manjavacas) ;
- ▶ **plus de données !** (quantité, qualité).

La pandora postcorrect app

- ▶ Développement d'une application de post-correction ;
- ▶ rendre plus rapide la correction (détection de similarité, correction par lots, etc.) ;
- ▶ maintenir l'intégrité des données (référentiels).

Création/gestion de corpus

Pandora Post-Correction Editor New Corpus Floovant ▾

Corpus Floovant

This corpus has 21 625 tokens.

[Edit tokens](#)[Export tokens](#)[History](#)[Edit tokens with unallowed lemma](#)[Edit tokens with unallowed POS](#)[Edit tokens with unallowed morph](#)

Settings :

[Edit Lemma allowed values](#)[Edit POS allowed values](#)[Edit morph allowed values](#)

Corpus SBath - List of tokens

1 2 3 4 5 ... 111 112

Form	Lemma	POS	Morph	Context	Save
Ch'	ce1	PROdem	—	Ch' est le	Save
est	être1	VERcjjg	—	Ch' est le vie	Save
le	le	DETdef	—	Ch' est le vie de	Save
vie	vie	NOMcom	—	Ch' est le vie de sainte	Save
de	de	PRE	—	est le vie de sainte Baltelt	Save
sainte	sainte	ADJqua	—	le vie de sainte Baltelt roine	Save
Baltelt	Bathilde	NOMpro	—	vie de sainte Baltelt roine .	Save
roine	reine	NOMcom	—	de sainte Baltelt roine . Beneois	Save
.	.	PONfrt	—	sainte Baltelt roine . Beneois soit	Save
Beneois	bénir	VERppe	—	Baltelt roine . Beneois soit Nostres	Save

Vérifications par lots

Corpus Floovant - Similar tokens

Match	Partial	Complete	Match at least	Lemma	POS	Morph	Different on	Lemma	POS	Morph
-------	---------	----------	----------------	-------	-----	-------	--------------	-------	-----	-------

All matches are at least a match on form.

1

Original token

Form	Context	Lemma	POS	Morph
or	SOIGNORS or escoutez que Dés	or4	ADVgen	-

Similar matching

Form	Lemma	POS	Morph	Context	Save
or	or4	ADVgen	-	enginé Maudite soit or l eure que	Save
or	or4	ADVgen	-	espées forbies Et or ai tot pordu	Save

Références

Pandora Kestemont (Mike), Jean-Baptiste Camps, Thibault Clérice, et Enrique Manjavacas, *Pandora : A (language-independent) Tagger-Lemmatizer for Latin & the Vernacular*, Anvers et Paris, 2016-...,

<https://github.com/hipster-philology/pandora/>.

Clérice (Thibault), *pandora-postcorrect-app*, Paris, 2017-...

<https://github.com/hipster-philology/pandora-postcorrect-app>.

Kestemont (M.), De Pauw (G.), Van Nie (R.) & Daelemans (W.),
'Lemmatisation for Variation-Rich Languages Using Deep Learning'.

Forthcoming in : DSH – Digital Scholarship in the Humanities.

<https://academic.oup.com/dsh/article/doi/10.1093/llc/fqw034/2669790/Lemmatization-for-variation-rich-languages-using>

Kestemont, M. & J. de Gussem, 'Integrated Sequence Tagging for Medieval Latin Using Deep Representation Learning', *Journal of Data Mining & Digital Humanities* (2017), pp. 17. Special Issue on Computer-Aided Processing of Intertextuality in Ancient Languages, ed. M. Buechler and L. Mellerin

<https://jdm.dh.episciences.org/3835/pdf>.

w2vec

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean J. (2013).
Distributed representations of words and phrases and their compositionality.
Neural Information Processing Systems, 26 : 3111–9.