Damegender: Gender detection tool

David Arroyo Menéndez

January 16, 2019

Damegender from the commands (I)

```
# Detect gender from a name
$ python3 main.py David
male
# Count gender from a csv example file
$ python3 csv2gender.py files/partial.csv
The number of males in files/partial.csv is 16
The number of females in files/partial.csv is 3
The number of gender not recognised in files/partial.csv is 2
# Count gender from a git repository
$ python3 git2gender.py https://github.com/chaoss/grimoirelab-p
The number of males sending commits is 15
The number of females sending commits is 7
```

Damegender from the commands (II)

```
# Count gender from a mailing list
$ cd files
$ wget -c http://mail-archives.apache.org/mod_mbox/httpd-annour
$ cd ..
$ python3 mail2gender.py http://mail-archives.apache.org/mod_mb
# Use an api to detect the gender
$ python3 api2gender.py David --api=genderguesser
male
# To measure success
$ python3 accuracy.py
Namsor accuracy: 0.9047619047619048
Sexmachine accuracy: 0.7619047619047619
```

Damegender from the commands (III)

```
$ python3 confusion.py
A confusion matrix C is such that Ci,j is equal to the number o
If the classifier is nice, the diagonal is high because there a
Namsor confusion matrix:
 [[ 3  0  0]
 [ 0 16  0]
 [ 0  2  0]]
Sexmachine confusion matrix:
 [[ 2  1  0]
 [ 2 14  0]
 [ 1  1  0]]

# To deploy a graph about correlation between variables
$ python3 corr.py
# To create the pickle models in files directory
$ python3 dememodels.py
```

Damegender as an exercise to practice NLTK and Perceval

```
~/git/python-examples/nlp/nltk: (dev) $ python3 sexmachine.py
What's your name?: David
What's my name?: Elena
David is male and Elena is female. Enjoy!.
The classifier has an accuracy: 0.052
Most Informative Features
 last_letter = 'a'              female : male   =     35.5 : 1.0
 last_letter = 'k'                male : female =     34.1 : 1.0
 last_letter = 'f'                male : female =     15.9 : 1.0
 last_letter = 'p'                male : female =     13.5 : 1.0
 last_letter = 'v'                male : female =     12.7 : 1.0
```

Damegender choosing features

```
$ python3 infofeatures.py
-----------------------------------------------------------------
Females with last letter a: 0.4705246078961601
Males with last letter a: 0.04867256637168141
-----------------------------------------------------------------
Females with last letter consonant: 0.2735841767750908
Males with last letter consonant: 0.635532897268180 1
-----------------------------------------------------------------
Females with last letter vocal: 0.7262612995441552
Males with last letter vocal: 0.3640823393612928
-----------------------------------------------------------------
```

Damegender coding features

```
def features_int(self, name):
# features method created to check the scikit classifiers
features_int = {}
features_int["first_letter"] = ord(name[0].lower())
features_int["last_letter"] = ord(name[-1].lower())
for letter in 'abcdefghijklmnopqrstuvwxyz':
features_int["count({})".format(letter)] = name.lower().count(l
features_int["vocals"] = 0
for letter1 in 'aeiou':
for letter2 in name:
if (letter1 == letter2):
features_int["vocals"] = features_int["vocals"] + 1
features_int["consonants"] = 0
for letter1 in 'bcdfghjklmnpqrstvwxyz':
for letter2 in name:
if (letter1 == letter2):
features_int["consonants"] = features_int["consonants"] + 1
```

Damegender the nltk standard model

```
def classifier(self):
labeled_names = ([(name, 'male') for name in names.words('male.
 [(name, 'female') for name in names.words('female.txt')])
featuresets = [(self.features(n), gender) for (n, gender) in la
train_set, test_set = featuresets[500:], featuresets[:500]
classifier = nltk.NaiveBayesClassifier.train(train_set)
return classifier

def guess(self, name, binary=False):
guess = ''
guess = super().guess(name, binary)
if ((guess == 'unknown') | (guess == 2)):
classifier = self.classifier()
guess = classifier.classify(self.features(name))
if binary:
if (guess=='male'):
guess = 1
```

Damegender building a ML model (scikit)

```
def sgd(self):
# Scikit classifier
X = np.array(self.features_list(path="files/all.csv"))
y = self.gender_list("files/all.csv")
clf = SGDClassifier(loss="log").fit(X,y)
filename = 'files/sgd_model.sav'
pickle.dump(clf, open(filename, 'wb'))
return clf

def sgd_load(self):
pkl_file = open('files/sgd_model.sav', 'rb')
clf = pickle.load(pkl_file)
pkl_file.close()
return clf
```

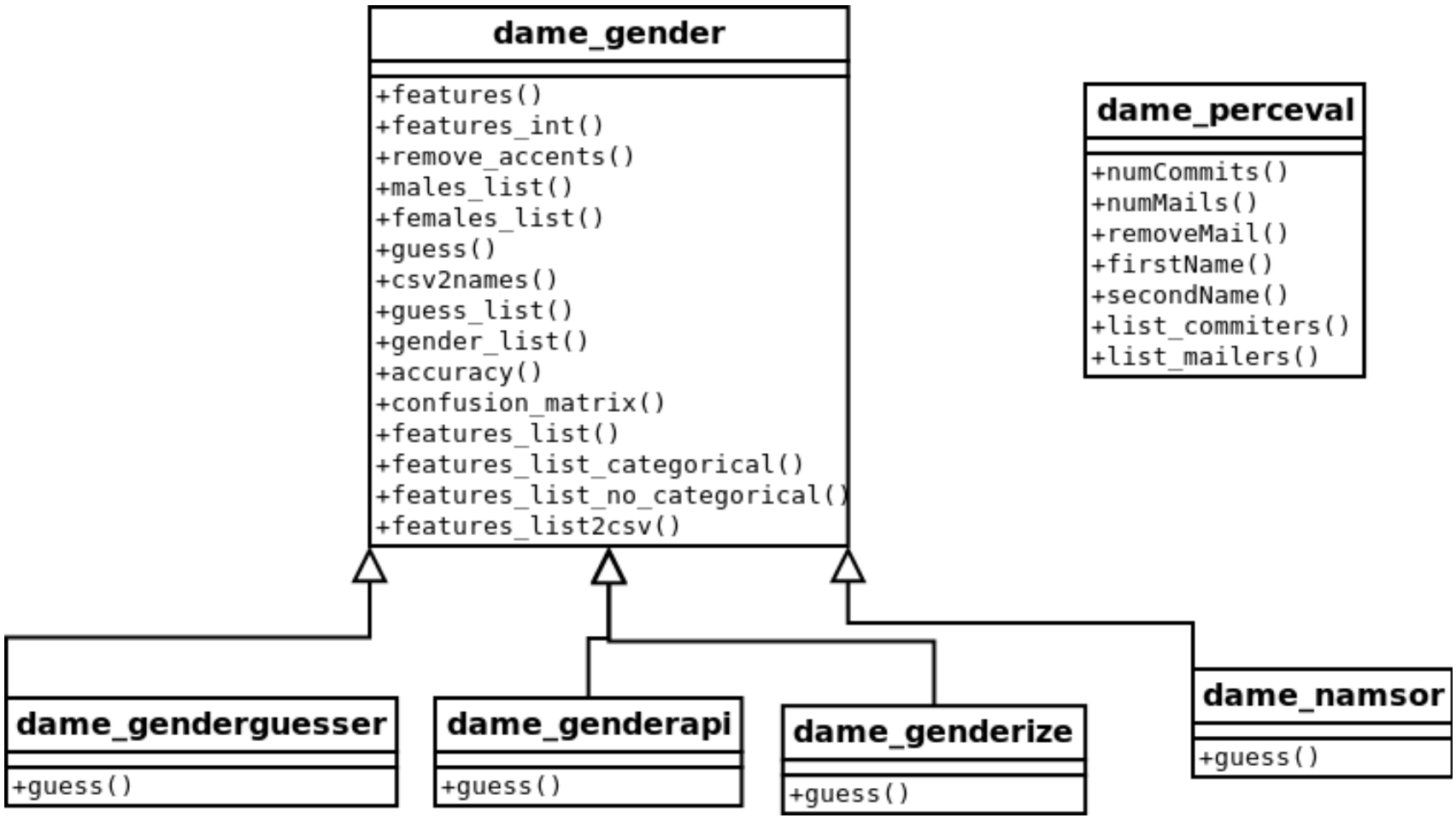
Damegender using a ML model (scikit)

```
$ cat main.py
```

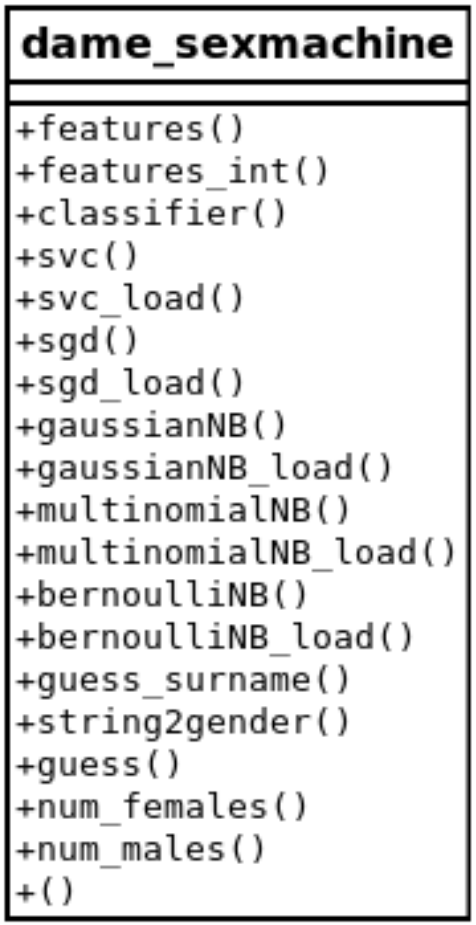
Damegender and perceval from string to gender

1. removeMail
2. string2array
3. string2gender (taking into account surnames and prefixes)

Damegender classes and methods (I)



Damegender classes and methods (II)



Comparison and benchmark of name-to-gender inference services (I)

Table 1 Comparison table showing relevant features for the gender inference services under study. Note that although Gender API does provide a specific API end point for handling surnames, our results employ the version that does not make use of them.

	Gender API	gender-guesser	genderize.io	NameAPI	NamSor
Database size (January 2018)	1,877,787	45,376	216,286	510,000	1,300,000
Regular data updates	yes	no	yes	yes	yes
Handles unstructured full name strings	yes	no	no	yes	no
Handles surnames	yes	no	no	yes	yes
Handles non-Latin alphabets	partially	no	partially	yes	yes
Implicit geo-localization	no	no	no	yes	yes
Assignment type	probabilistic	binary	probabilistic	probabilistic	probabilistic
Free parameters	accuracy, samples	–	probability, count	confidence	scale
Open source	no	yes	no	no	no
API	yes	no	yes	yes	yes
Monthly free requests	500	unlimited	30,000	10,000	1,000
Monthly subscription cost (100,000 requests/month)	79 €	Free	7 €	150 €	80 €
Provider	Gender-API.com	Israel Saeta Pérez	Casper Strømgren	Optimaize GmbH	NamSor SAS

Comparison and benchmark of name-to-gender inference services (I)

Database size	damegender
Regular data updates	yes, developing
Handles unstructured full name strings	yes
Handles surnames	yes
Handles non-Latin alphabets	no
Implicit geo-localization	no
Assingment type	probabilistic
Free parameters	ml
Free license	yes
API	future
free requests limited	unlimited

Damegender and the bussiness

- In CMS: wordpress, drupal, joomla
- In dictionaries: google translate, babylon, gnu dict, ...
- Enciclopedias: wikipedia, ...

–

- A good technical project has a good bussiness project and an interfaz for end users.
- A Free Software license and community can be a good point.

Comparison and benchmark of name-to-gender inference services (II): Assembling data

- zbMath: names from articles, labeled by humans using university websites, Wikipedia articles, etc.
- genderizeR: from full names searching in Internet biographies, personal data, etc.
- PubMed: six highest JCR
- WoS: from 2008 to 2012 included in Science Citation Index Expanded, the Social Sciences Citation Index, and the Arts and Humanities Citation Index