1 Évaluation des approches (`semeval_executable.py`)

1.1 Méthodes EXPORT

`getpredfilename` permet de s'assurer que les noms des fichiers de prédiction sont tous construits de la même manière.

```
def getpredfilename(doctree, indicator, filterspartition, methodcategory):
    return 'predictions/' + '_'.join((doctree, indicator, *filterspartition,
                      methodcategory, 'scores.pred'))
```

Un script shell est utilisé pour extraire le score MAP d'un fichier de prédiction :

```
prediction=$1

if echo $prediction | grep --quiet "2016"
then
    reference=scorer/SemEval2016-Task3-CQA-QL-test.xml.subtaskB.relevancy
else
    if echo $prediction | grep --quiet "2017"
    then
        reference=scorer/SemEval2017-Task3-CQA-QL-test.xml.subtaskB.relevancy
    else
        reference=scorer/SemEval2016-debug.relevancy
    fi
fi

python2 scorer/ev.py $reference $prediction | grep "^MAP" | sed 's/ \+/;/g' | cut -f 4 -d
 ↪  ';'
```

1.1.1 Méthodes bruteforce

Les méthodes bruteforce correspondent à tester toutes les combinaisons d'arbres de documents, de sacs de mots et de filtres. Les méthodes bruteforce sont crées en faisant le produit cartésien des dimensions envisagées.

Les méthodes précédemment générées sont exécutées et les scores produits sont écrits dans les fichiers correspondants.

```
bruteforce_methods = (doctrees, all_indicators, filters_partition)


out_of_corpus_value = max(inversedocfreqs['text_document'].values())

for doctree, indicator, filterspartition in product(*bruteforce_methods):
    wordex, sentex = all_indicators[indicator]
    customscorer = scorer(
        wordextractors[wordex],
        sentenceextractors[sentex],
        [filters[filterkey] for filterkey in filterspartition])

    scores = make_score_tree(
        doctrees[doctree],
        lambda a, b: customscorer.get_score(
```

```
            a, b,
            inversedocfreqs[wordex + '_' + sentex],
            out_of_corpus_value)
    )

    prediction_file = getpredfilename(doctree, indicator, filterspartition, 'bruteforce')
    write_scores_to_file(scores, prediction_file, verbose=True)
```

année 2016

Sac de mots	Filtres	Score MAP
Lemmes	Mots outils, ≤ 2	0.7679
Textes des entités nommées	Mots outils	0.7493
Textes des entités nommées	Mots outils, ≤ 1	0.7493
Textes des entités nommées	Mots outils, ≤ 2	0.7493
Textes des entités nommées	Mots outils, ≤ 3	0.7493
Textes des entités nommées	Mots outils, ≤ 4	0.7493
Textes des entités nommées	≤ 1	0.7493
Textes des entités nommées	≤ 2	0.7493
Textes des entités nommées	≤ 3	0.7493
Textes des entités nommées	≤ 4	0.7493
Textes des entités nommées	Pas de filtre	0.7493
Lemmes	Mots outils, ≤ 1	0.7490
Tokens	Mots outils, ≤ 2	0.7488
Tokens	Mots outils, ≤ 1	0.7446
Lemmes	Mots outils	0.7424
Tokens	Mots outils, ≤ 3	0.7423
Tokens	Mots outils, ≤ 4	0.7422
Étiquettes des entités nommées	Mots outils, ≤ 4	0.7369
Étiquettes des entités nommées	≤ 4	0.7369
Tokens	Mots outils	0.7322
Lemmes	Mots outils, ≤ 3	0.7317
Lemmes	≤ 2	0.7293
Lemmes	≤ 4	0.7292
Tokens	≤ 4	0.7275
Tokens	≤ 2	0.7274
Lemmes	≤ 1	0.7269
Lemmes	≤ 3	0.7253
Tokens	≤ 3	0.7219
Lemmes	Mots outils, ≤ 4	0.7202
Étiquettes des entités nommées	Mots outils	0.7153
Étiquettes des entités nommées	Mots outils, ≤ 1	0.7153
Étiquettes des entités nommées	Mots outils, ≤ 2	0.7153
Étiquettes des entités nommées	≤ 1	0.7153
Étiquettes des entités nommées	≤ 2	0.7153
Étiquettes des entités nommées	Pas de filtre	0.7153
Lemmes	Pas de filtre	0.7148
Tokens	≤ 1	0.7135
Tokens	Pas de filtre	0.7098
Étiquettes des entités nommées	Mots outils, ≤ 3	0.7081
Étiquettes des entités nommées	≤ 3	0.7081

année 2017

Sac de mots	Filtres	Score MAP
Tokens	Mots outils, ≤ 1	0.4705
Lemmes	Mots outils, ≤ 1	0.4678
Tokens	Mots outils, ≤ 2	0.4653
Tokens	Mots outils, ≤ 3	0.4650
Lemmes	Mots outils, ≤ 2	0.4624
Tokens	≤ 2	0.4623
Tokens	Mots outils	0.4598
Lemmes	Mots outils	0.4581
Lemmes	Pas de filtre	0.4580
Tokens	Mots outils, ≤ 4	0.4521
Lemmes	≤ 1	0.4473
Lemmes	Mots outils, ≤ 3	0.4458
Tokens	≤ 1	0.4455
Lemmes	≤ 2	0.4425
Lemmes	Mots outils, ≤ 4	0.4419
Tokens	Pas de filtre	0.4418
Tokens	≤ 3	0.4392
Lemmes	≤ 3	0.4389
Lemmes	≤ 4	0.4249
Tokens	≤ 4	0.4153
Textes des entités nommées	Mots outils, ≤ 3	0.4139
Textes des entités nommées	Mots outils, ≤ 4	0.4139
Textes des entités nommées	≤ 3	0.4139
Textes des entités nommées	≤ 4	0.4139
Étiquettes des entités nommées	Mots outils	0.4123
Étiquettes des entités nommées	Mots outils, ≤ 1	0.4123
Étiquettes des entités nommées	Mots outils, ≤ 2	0.4123
Étiquettes des entités nommées	≤ 1	0.4123
Étiquettes des entités nommées	≤ 2	0.4123
Étiquettes des entités nommées	Pas de filtre	0.4123
Étiquettes des entités nommées	Mots outils, ≤ 3	0.4104
Étiquettes des entités nommées	≤ 3	0.4104
Textes des entités nommées	Mots outils	0.4083
Textes des entités nommées	Mots outils, ≤ 1	0.4083
Textes des entités nommées	Mots outils, ≤ 2	0.4083
Textes des entités nommées	≤ 1	0.4083
Textes des entités nommées	≤ 2	0.4083
Textes des entités nommées	Pas de filtre	0.4083
Étiquettes des entités nommées	Mots outils, ≤ 4	0.4063
Étiquettes des entités nommées	≤ 4	0.4063

Année	Score MAP baseline
2016	0.7475
2017	0.4185

1.1.2 Méthodes pondérées

Le but des méthodes pondérées est d'utiliser plusieurs indicateurs au sein d'une même méthode. Un exemple d'approche de pondération est d'utiliser les lemmes pour estimer la similarité de phrases, et de donner une plus grande importance aux lemmes communs qui sont également des entités nommées.

1. Recherche des pondérations optimales
2. Pondération par entités nommées

```
ponderated_methods = (doctrees, morphologic_indicators, filters_partition)

for doctree, indicator, fltrs in product(*ponderated_methods):
    wordex, sentex = all_indicators[indicator]

    scores = make_score_tree(
        doctrees[doctree],
        lambda a, b: entityweight_scorer(
            wordextractors[wordex],
            [filters[filterkey] for filterkey in fltrs],
            a, b, inversedocfreqs[wordex + '_' + sentex],
            out_of_corpus_value
        )
    )

    prediction_file = getpredfilename(doctree, indicator, fltrs, 'nerponderation')
    write_scores_to_file(scores, prediction_file, verbose=True)
```

année 2016

Sac de mots	Filtres	Score MAP
Lemmes	Mots outils, ≤ 2	0.7663
Tokens	Mots outils, ≤ 2	0.7497
Lemmes	Mots outils, ≤ 1	0.7474
Tokens	Mots outils, ≤ 1	0.7446
Tokens	Mots outils, ≤ 3	0.7430
Tokens	Mots outils, ≤ 4	0.7422
Lemmes	Mots outils	0.7405
Tokens	Mots outils	0.7319
Lemmes	Mots outils, ≤ 3	0.7301
Lemmes	≤ 4	0.7292
Lemmes	≤ 2	0.7287
Tokens	≤ 4	0.7264
Tokens	≤ 2	0.7254
Lemmes	≤ 1	0.7239
Lemmes	≤ 3	0.7237
Tokens	≤ 3	0.7214
Lemmes	Mots outils, ≤ 4	0.7202
Lemmes	Pas de filtre	0.7167
Tokens	≤ 1	0.7142
Tokens	Pas de filtre	0.7078

année 2017

Sac de mots	Filtres	Score MAP
Tokens	Mots outils, ≤ 1	0.4725
Lemmes	Mots outils, ≤ 1	0.4707
Tokens	Mots outils, ≤ 3	0.4658
Tokens	Mots outils, ≤ 2	0.4651
Tokens	Mots outils	0.4622
Tokens	≤ 2	0.4621
Lemmes	Mots outils, ≤ 2	0.4618
Lemmes	Mots outils	0.4599
Tokens	Mots outils, ≤ 4	0.4521
Lemmes	Pas de filtre	0.4509
Lemmes	≤ 1	0.4477
Tokens	≤ 1	0.4475
Lemmes	Mots outils, ≤ 3	0.4432
Lemmes	≤ 2	0.4424
Lemmes	Mots outils, ≤ 4	0.4413
Tokens	≤ 3	0.4412
Tokens	Pas de filtre	0.4412
Lemmes	≤ 3	0.4387
Lemmes	≤ 4	0.4252
Tokens	≤ 4	0.4124