Analyse syntaxique multilingue

Traitement du Langage Naturel et Linguistique
Projet

16 octobre 2020

1 Objectif

L'objectif de ce projet est de développer un analyseur multilingue délexicalisé. Un tel analyseur prend en entrée une séquence de parties de discours correspondant à une phrase et produit un arbre de dépendances pour cette phrase. L'analyseur n'a donc pas accès à la représentation orthographique des mots, c'est en ce sens qu'il est délexicalisé. Son autre particularité est qu'il peut prendre en entrée des séquences de parties de discours de langues différentes, c'est en ce sens qu'il est multilingue. L'analyseur ne connait pas exactement la langue à laquelle appartient une phrase particulière, mais il en a une description partielle, abstraite.

Le projet est constitué de deux parties. Dans la première partie, on négligera l'aspect multilingue. Son objectif est la prise en main de l'analyseur et la réalisation d'expériences monolingues (on construit un analyseur pour chacune des langues qui nous intéressent). Dans la seconde partie, on ajoute à l'analyseur une représentation abstraite, partielle, de chaque langue, afin d'étudier les conséquences des différents aspects de cette représentation abstraite sur les performances de l'analyseur.

2 Partie I : analyse monolingue

Cette partie repose sur une implémentation qui vous est fournie d'un analyseur en transition. Le code de l'analyseur se trouve dans le dépôt `git` suivant : `https://gitlab.lis-lab.fr/alexis.nasr/tbp.git`. Les données sur lesquelles les expériences seront réalisées se trouvent à l'adresse suivante : `http://pageperso.lif.univ-mrs.fr/~alexis.nasr/Ens/TLNL/data.tgz`.

Le données sont constituées de fichiers au format `conllu`. Ces fichiers comportent des phrases enrichies de leur analyse syntaxique. Les données

couvrent 32 langues différentes. Pour chacune d'entre elles, trois fichiers sont fournis. Un fichier d'apprentissage, un fichier de développement et un fichier de test. Etant donné la langue dont le code est `fr` (il s'agit du français), ces fichiers ont respectivement pour noms `train_fr.conllu`, `dev_fr.conllu` et `test_fr.conllu`.

L'objectif de cette partie est d'entraîner 32 analyseurs syntaxiques différents (un par langue) dans exactement les mêmes conditions, c'est à dire : quasiment la même quantité de données d'apprentissage, la même fonction de décomposition (feature function) et les mêmes hyperparamètres du perceptron multicouche.

Afin de vous fournir un point de comparaison, la Table 1 reporte les résultats obtenus sur l'ensemble des langues pour un modèle appris sur 10.000 mots, avec un corpus de développement de 5.000 mots et un corpus de test de 700 mots. La structure du classifieur permettant de prédire une action à effectuer étant donné une configuration est la suivante :

```
model = Sequential()
model.add(Dense(units=128, activation='relu', input_dim=inputSize))
model.add(Dropout(0.4))
model.add(Dense(units=outputSize, activation='softmax'))
model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])

model.fit(x_train, y_train, epochs=10, batch_size=32,
                  validation_data=(x_dev,y_dev))
```

La fonction de décomposition, qui correspond au fichier `basic.fm` est la suivante :

```
W B -2 POS
W B -1 POS
W B 0 POS
W B 1 POS
W B 2 POS
W S 0 POS
W S 1 POS
```

2.1 Ce qu'il faut faire

Votre travail pour cette partie consiste à constituer votre *baseline*. Il s'agit de votre système de référence qui vous servira, dans la seconde partie du

L	LAS	UAS	L	LAS	UAS	L	LAS	UAS
ar	60.14	68.75	fa	51.26	60.92	nl	54.00	62.93
bg	68.86	80.03	fr	69.52	74.57	nno	73.74	78.93
ca	61.83	68.59	he	63.32	68.86	nob	67.25	75.55
cs	70.83	79.23	hi	63.73	72.38	pl	71.58	85.08
da	66.17	73.06	hr	57.54	67.45	pt	70.97	75.51
de	56.44	64.49	hu	58.09	68.49	ro	53.72	64.96
el	71.13	78.64	id	69.09	74.14	sl	47.91	59.19
en	67.58	72.67	it	75.25	81.00	sv	63.47	71.55
es	66.52	72.75	ja	75.11	85.39	vi	49.46	50.83
et	59.64	73.24	ko	46.36	55.90	zh	45.42	54.86
eu	48.39	59.84	lv	54.69	62.24			

TABLE 1 – Labeled Accuracy Score (LAS) et Unlabeled Accuracy Score (UAS) pour 32 langues différentes dans des conditions d'apprentissage proches.

projet, à comparer une approche monolingue et multilingue. Pour cela, vous trouverez un jeux d'hyper-paramètres raisonnable (on entend ici par hyper-paramètres, les hyper-paramètres du classifieur proprement dit, mais aussi la taille des données d'apprentissage, de développement et de test, ainsi que la fonction de décomposition).

D'un point de vue lingustique la partie la plus intéressante concerne la mise au point de la fonction de décomposition. C'est en effet elle qui permet de déterminer les variables linguistiques qui semblent intéressantes pour réaliser une analyse syntaxique.

Dans le code qui vous est fourni, seules les *Word Features* sont implémentées, c'est à vous d'implémenter les *Configurational Features*.

A l'issue de cette partie, vous devrez fournir les résultats que vous avez obtenu pour les différentes langues, les hyper-paramètres que vous avez utilisé ainsi que la fonction de décomposition que vous avez définie. Plus une description de la méthode que vous avez suivie pour arriver à vos hyper-paramètres et à votre fonction de décomposition.

3 Partie II : analyse multilingue