

IFT3335 : Introduction à l'intelligence artificielle

TP2 Utiliser la classification pour la désambiguïsation de sens de mots

Rapport

Nom : Boumediene Boukharouba Matricule : 20032279

Nom : Ahmed Mohamed Matricule : 20024603

Test avec arbre de décision :

Apprentissage supervisée :

Afin d'exécuter cette tâche, nous avons dû nous familiariser avec Weka. Les étapes pour lancer une classification (basique) avec ce dernier sont les suivantes :

D'abord charger le fichier **Interest-3-words.arff** via le bouton explorer

Pour choisir une classification de type Arbre de décision : Classify-> Choose -> Trees-> J48, avec un folds de 10. Ensuite, nous avons effectué plusieurs types de tests en variant le nombre de mots sur lesquels on se base pour faire la classification :

Test 1 : Word1_type

No.	Name
1	<input type="checkbox"/> previousWord1_type
2	<input type="checkbox"/> nextWord1_type
3	<input type="checkbox"/> class

Le résultat est le suivant :

Correctly Classified Instances 1515 64.0051 %

Incorrectly Classified Instances 852 35.9949 %

C'est-à-dire que 64% de nos classes étaient bien classifiées alors que près de 36% ne l'ont pas été

=== Confusion Matrix ===

	a	b	c	d	e	f	<-- classified as
69	0	0	3	197	92		a = 1
1	0	0	0	5	5		b = 2
3	0	0	0	54	9		c = 3
34	0	0	25	101	18		d = 4
49	0	0	3	314	134		e = 5
38	1	0	0	105	1107		f = 6

La matrice de confusion met en évidence le nombre de classe qui ont bien été classifiées et cela pour chaque catégorie. Les classes qui ont bien été classifiées sont sur la diagonale principale et on voit clairement que ce résultat n'est pas pertinent contenu du fait que la classe B et C ont totalement été mal classées.

Test avec NetxWord1 previousWord 1:

Correctly Classified Instances	1730	73.0883 %
Incorrectly Classified Instances	637	26.9117 %

Test Avec Word2_type + Word1_type :

Correctly Classified Instances	1672	70.6379 %
Incorrectly Classified Instances	695	29.3621 %

a	b	c	d	e	f	<-- classified as
177	0	10	29	103	42	a = 1
2	3	0	0	4	2	b = 2
23	0	19	4	11	9	c = 3
51	0	8	70	32	17	d = 4
129	1	10	36	274	50	e = 5
62	0	4	8	48	1129	f = 6

La matrice de confusion met en évidence une nette amélioration par rapport à Word1_type . Ce qui est logique car on se base sur deux mots de plus.

Test Avec Word2 + Word1 :

Correctly Classified Instances	1924	81.2843 %
Incorrectly Classified Instances	443	18.7157 %

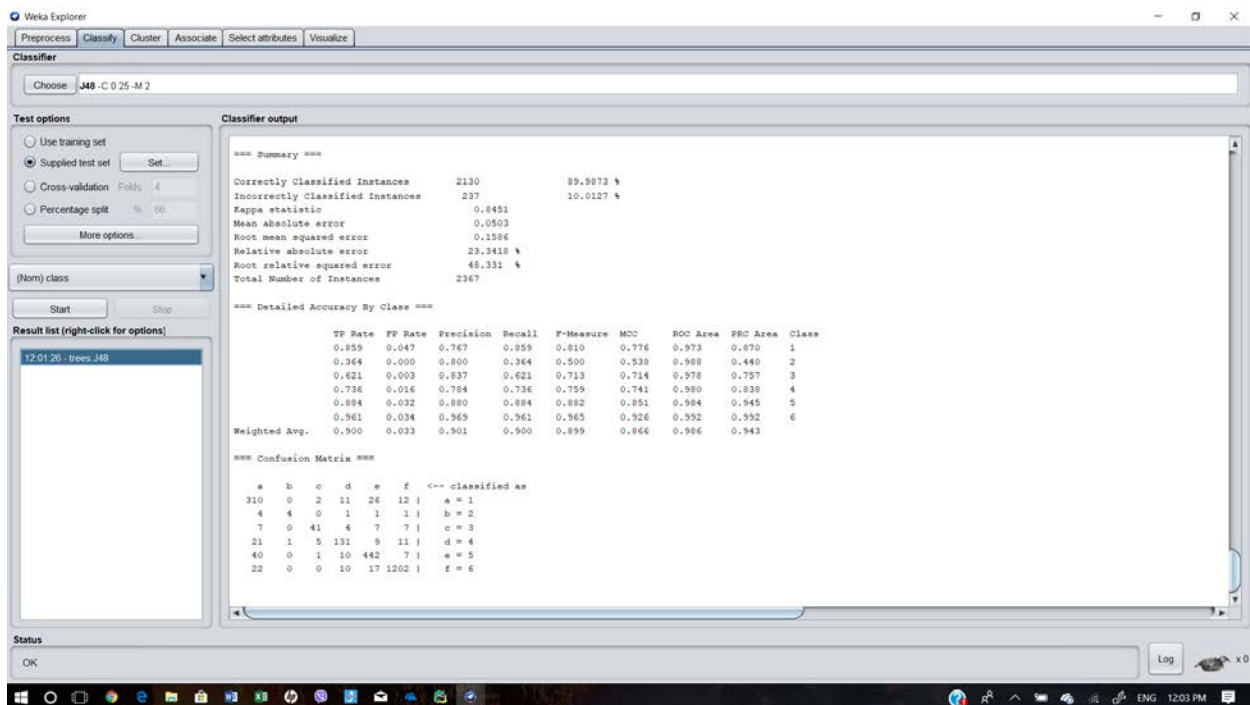
Test Avec Word 3 type + Word2_type + Word1_type :

Correctly Classified Instances	1739	73.4685 %
Incorrectly Classified Instances	628	26.5315 %

Test Avec Word 3 + Word 2 + Word 1 :

Correctly Classified Instances	1892	79.9324 %
Incorrectly Classified Instances	475	20.0676 %

Apprentissage non supervisée et stemming (1000 mots) :



Word1 + Word2 + Word3 (Next & previous) + Word1_type + Word2_type + Word3_type (Next & previous) :

Correctly Classified Instances 2130 89.9873 %
Incorrectly Classified Instances 237 10.0127 %

Naïve Bayes :

Word1 Next & previous

Correctly Classified Instances 1665 70.3422 %
Incorrectly Classified Instances 702 29.6578 %

Word1_type Next & previous :

Correctly Classified Instances 1535 64.85 %
Incorrectly Classified Instances 832 35.15 %

Word1 + Word2 (Next & previous)

Correctly Classified Instances	1796	75.8766 %
Incorrectly Classified Instances	571	24.1234 %

Word1 type + Word2 type (Next & previous) :

Correctly Classified Instances	1667	70.4267 %
Incorrectly Classified Instances	700	29.5733 %

Word1 + Word2 + Word3 (Next & previous)

Correctly Classified Instances	1749	73.891 %
Incorrectly Classified Instances	618	26.109 %

Word1 type + Word2 type + Word3 type (Next & previous) :

Correctly Classified Instances	1705	72.0321 %
Incorrectly Classified Instances	662	27.9679 %

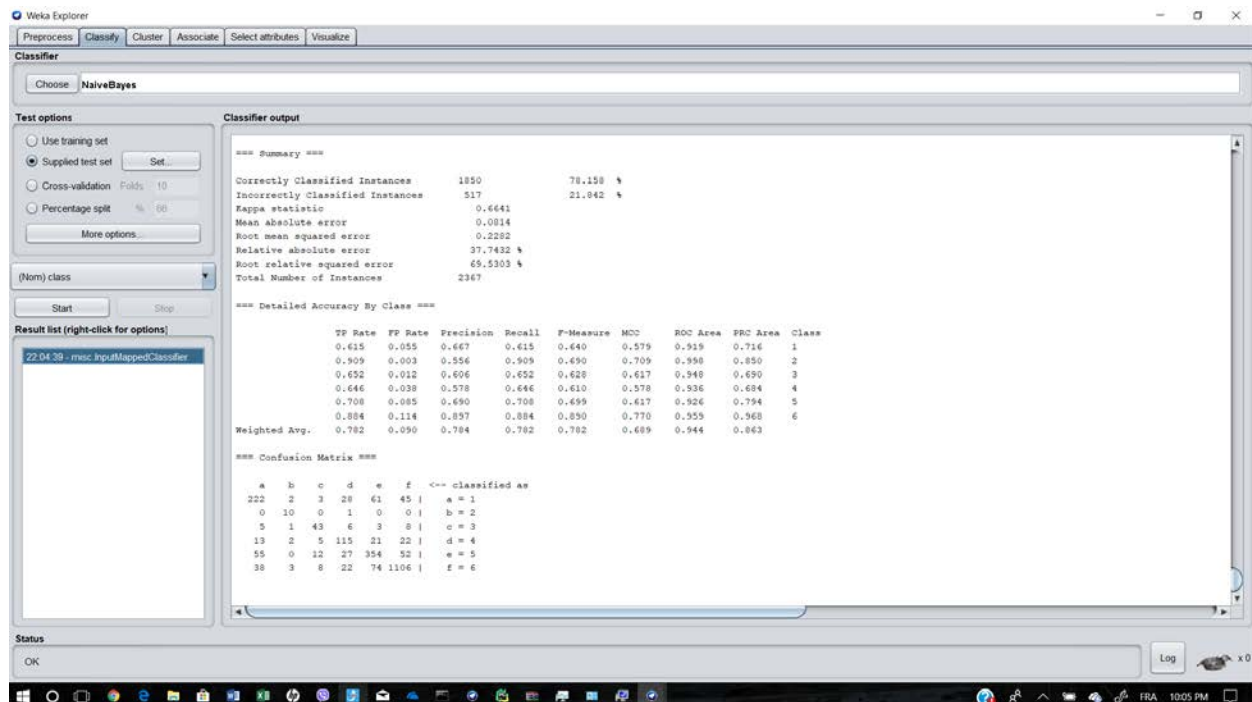
Word1 + Word2 + Word3 (Next & previous) :

Correctly Classified Instances	1749	73.891 %
Incorrectly Classified Instances	618	26.109 %

Word1 + Word2 + Word3 (Next & previous) + Word1 type + Word2 type + Word3 type (Next & previous) :

Correctly Classified Instances	1848	78.0735 %
Incorrectly Classified Instances	519	21.9265 %

Apprentissage non supervisée stemming (1000 mots) :



Word1 + Word2 + Word3 (Next & previous) + Word1_type + Word2_type + Word3_type (Next & previous) :

Correctly Classified Instances 1850 78.158 %
Incorrectly Classified Instances 517 21.842 %

Test avec SMO :

Apprentissage supervisée :

Word1 Next & previous

Correctly Classified Instances 1753 74.06 %
Incorrectly Classified Instances 614 25.94 %

=== Confusion Matrix ===

	a	b	c	d	e	f	<-- classified as
140	0	3	18	95	105		a = 1
4	0	0	1	1	5		b = 2
4	0	39	7	11	5		c = 3
23	0	5	80	31	39		d = 4
28	0	7	7	341	117		e = 5
14	1	0	5	78	1153		f = 6

On remarque que plusieurs classes ont été mal classifié. Cependant, la classes qui ont dû être dans la classe b ont tous été mal classifiés.

Word1_type Next & previous :

Correctly Classified Instances	1517	64.0896 %
Incorrectly Classified Instances	850	35.9104 %

Word1 + Word2 (Next & previous)

Correctly Classified Instances	2019	85.2978 %
Incorrectly Classified Instances	348	14.7022 %

Word1_type + Word2_type (Next & previous) :

Correctly Classified Instances	1709	72.2011 %
Incorrectly Classified Instances	658	27.7989 %

Word1 + Word2 + Word3 (Next & previous)

Correctly Classified Instances	2086	88.1284 %
Incorrectly Classified Instances	281	11.8716 %

=== Confusion Matrix ===

	a	b	c	d	e	f	<-- classified as
286	0	1	23	34	17		a = 1
5	4	0	2	0	0		b = 2
8	0	40	8	4	6		c = 3
25	0	3	125	18	7		d = 4
42	0	2	16	427	13		e = 5
24	0	0	11	12	1204		f = 6

Les erreurs de classification sont moindres que les testes précèdent. Aussi, et contrairement au teste ou on utilisait un seul mot avant et après, on voit que la classe b n'est pas totalement mal classifié et qu'il n'y a pas de zéro sur la diagonale

Word1_type + Word2_type + Word3_type (Next & previous) :

Correctly Classified Instances	1775	74.9894 %
Incorrectly Classified Instances	592	25.0106 %

Word1 + Word2 + Word3 (Next & previous) + Word1_type + Word2_type + Word3_type (Next & previous) :

Correctly Classified Instances	2061	87.0722 %
Incorrectly Classified Instances	306	12.9278 %

Apprentissage non supervisée et stemming (1000 mots) :

The screenshot shows the Weka Explorer interface with the Classifier tab selected. The classifier chosen is SMO-C 1 0 -L 0 001 -P 1 0E-12 -N 0 -V -1 -W 1 -K *weka.classifiers.functions.supportVector.PolyKernel-E 1 0 -C 250000* -calibrator *weka.classifiers.functions.Logistic-R 1 0E-8 -M -1 -num-decimal-places 4*. The Test options are set to Use training set, Supplied test set, Cross-validation (Folds: 10), and Percentage split (%: 60). The Classifier output shows the following summary:

```
=== Summary ===
Correctly Classified Instances      2282      96.409 %
Incorrectly Classified Instances      85      3.591 %
Kappa statistic                    0.9445
Mean absolute error                 0.0232
Root mean squared error             0.3116
Relative absolute error             103.5452 %
Root relative squared error         54.9422 %
Total Number of Instances          2367
```

The Detailed Accuracy By Class table is as follows:

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.936	0.019	0.897	0.936	0.916	0.901	0.901	0.980	0.863	1
0.909	0.000	1.000	0.909	0.952	0.953	0.952	0.910	0.910	2
0.939	0.001	0.969	0.939	0.954	0.953	0.972	0.914	0.914	3
0.871	0.009	0.886	0.871	0.878	0.868	0.974	0.803	0.803	4
0.962	0.009	0.968	0.962	0.965	0.956	0.988	0.948	0.948	5
0.988	0.007	0.994	0.988	0.991	0.981	0.995	0.991	0.991	6
Weighted Avg.	0.964	0.009	0.965	0.964	0.964	0.954	0.989	0.946	

The Confusion Matrix is as follows:

```
=== Confusion Matrix ===
  a  b  c  d  e  f  <-- classified as
338  0  0  9  11  3  |  a = 1
  0 10  0  1  0  0  |  b = 2
  1  0 62  1  0  2  |  c = 3
 16  0  1 155  3  3  |  d = 4
 15  0  1  3 481  0  |  e = 5
  7  0  0  6  2 1236 |  f = 6
```

Word1 + Word2 + Word3 (Next & previous) + Word1_type + Word2_type + Word3_type (Next & previous) :

Correctly Classified Instances 2282 96.409 %

Incorrectly Classified Instances 85 3.591 %

Test avec Multi-layer perceptron :

Apprentissage supervisée :

Classifier

Choose: **MultilayerPerceptron -L 0.3 -M 0.2 -N 300 -V 0 -S 0 -E 20 -H 5 -R -batch-size 70**

Test options

- ☐ Use training set
- ☐ Supplied test set
- ☒ Cross-validation Folds: **10**
- ☐ Percentage split % **66**

Classifier output

Time taken to build model: 10032.84 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	1251	52.8517 %
Incorrectly Classified Instances	1116	47.1483 %
Kappa statistic	0	
Mean absolute error	0.2149	
Root mean squared error	0.3309	
Relative absolute error	99.656 %	
Root relative squared error	100.807 %	
Total Number of Instances	2367	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,000	0,000	?	0,000	?	?	?	0,523	0,173	1
0,000	0,000	?	0,000	?	?	?	0,532	0,006	2
0,000	0,000	?	0,000	?	?	?	0,504	0,030	3
0,000	0,000	?	0,000	?	?	?	0,522	0,084	4
0,000	0,000	?	0,000	?	?	?	0,529	0,258	5
1,000	1,000	0,529	1,000	0,692	?	?	0,541	0,595	6
Weighted Avg.	0,529	0,529	?	0,529	?	?	0,533	0,403	

=== Confusion Matrix ===

Word1 + Word2 + Word3 (Next & previous) + Word1_type + Word2_type + Word3_type (Next & previous) batch size 70 training time 300 hidden layers 5:

Correctly Classified Instances 1251 52.8517 %

Incorrectly Classified Instances 1116 47.1483 %

Conclusion

Par rapport à l'arbre Nous avons que le meilleur résultat est meilleurs lorsqu'on choisit juste deux mots (2 mots prevision et deux mots next) mais le résultat l'apprentissage non supervisé et stemming donne un beaucoup meilleur résultat avec 90% des objets bien classifiés.

Pour Naïve bayes est plus performant si on prend en compte un maximum de mot avant et après ainsi que leur type. Toutefois ce dernier n'a pas donné des résultats aussi performants que l'arbre de décision et même l'apprentissage non supervisée a donné un résultat légèrement meilleur résultat que le meilleur avec l'apprentissage supervisée

Par rapport à SMO nous remarquons qu'elle met en évidence des résultats plus que pertinent et meilleures que ceux observé lors de l'utilisation du classifieur Bayes Naïves et l'arbre de décision. Aussi, nous remarquons que le meilleurs résultat obtenu est lorsque nous avons opté a utilisé Word1 + Word2 + Word3 (Next & previous) Mais le résultat avec l'apprentissage non supervisée donne le meilleur résultat en dépassant le 96% des objets bien classifiés.

Par rapport au Perceptron on l'a testé avec des options qui sont un peu faibles à cause du temps massive qu'elle prend pour faire l'inférence et la manque d'utilisation de GPU dans Weka qui pourrait réduire le temps massivement alors on l'a testé avec seulement 5 couches (hidden layers et on n'a pas pu la testé avec 10 ou plus de couches)