Atelier : Production d'un modèle affiné de reconnaissance d'écriture manuscrite avec eScriptorium et évaluation de ses performances. Évaluer son modèle HTR/OCR avec KaMI (Kraken as Model Inspector)

Lucas Terriel

@Atelier : Production d'un modèle affiné de reconnaissance d'écriture manuscrite avec *eScriptorium* et évaluation de ses performances

Évaluer son modèle HTR/OCR avec KaMI (*Kraken as Model Inspector*)

Les Futures Fantastiques - 3e Conférence Internationale sur l'Intelligence Artificielle appliquée aux Bibliothèques, Archives et Musées
#FF21

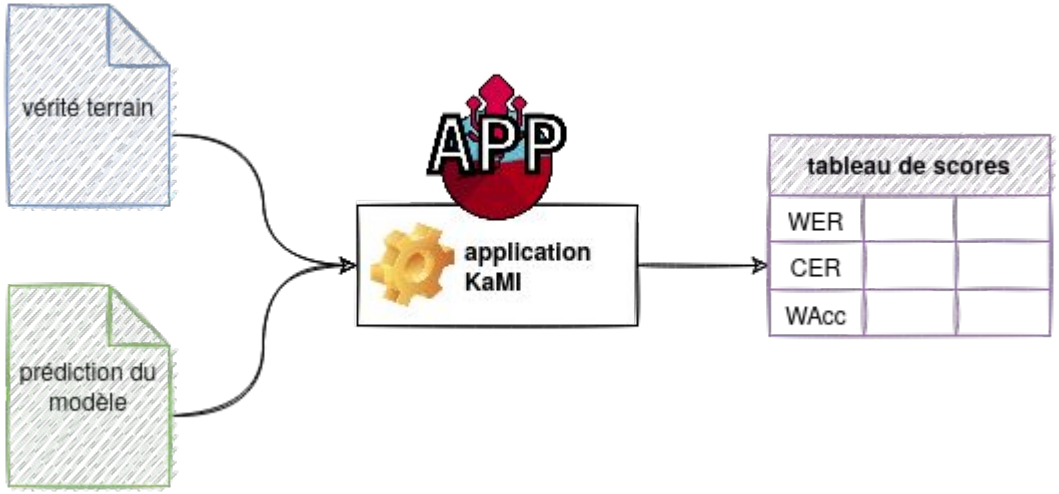
Bibliothèque nationale de France - 1er décembre 2021

Lucas Terriel
ingénieur recherche & développement au sein de l'équipe ALMAnaCH (Inria)
lucas.terriel@inria.fr
@Lucaterre

Contexte

- Librairie Python (KaMI-LIB) et application (KaMI-APP) pour évaluer des modèles HTR/OCR basés sur le moteur Kraken.
- Projet débuté à ALMANACH (INRIA) en 2020.

Fonctionnement de l'application



Objectifs

- Comprendre les métriques de performances pour évaluer vos modèles HTR/OCR
- Évaluer son modèle HTR/OCR rapidement
- Réutiliser KaMI dans votre projet

Sommaire

01

Quelques métriques indispensables : Opérations entre les chaînes de caractères, Distance de Levenshtein, WER et CER

Opérations entre chaînes de caractères

Les opérations entre les chaînes de caractères sont de 3 sortes :

- Les insertions
- Les substitutions
- Les suppressions

C1 :	E	M	M	A	G	A	S	I	N		E	R	
C2 :			M	E	G	A	S	I	N	I	E	R	S

Comparaison entre C1 (chaîne de référence) et C2 (chaîne à comparer) :

- 2 suppressions (“em” aux positions 1 et 2)
- 1 substitution (“e” à la position 4)
- 2 insertions (“i” et “s” aux positions 10 et 13)

Dans KaMI, ces opérations sont données dans le tableau de score final : insertions (Insertions), substitutions (Substitutions), suppressions (Deletions) et les signes parfaitement reconnus (Hits). Seulement sur les caractères.

Distance de Levenshtein (ou distance d'édition)

Distance mathématique qui donne l'écart entre deux chaînes de caractères et permet d'estimer une "similarité syntaxique" :

Somme des opérations entre les deux chaînes de caractères à comparer (chaque opération à un poids de 1).

C1 :	E	M	M	A	G	A	S	I	N		E	R	
C2 :			M	E	G	A	S	I	N	I	E	R	S

D(C1,C2) = 2 suppressions + 1 substitution + 2 insertions = 5

Dans KaMI, la distance de Levenshtein est donné au niveau des caractères et des mots des deux documents.

Métriques de performances : CER & WER

- Taux d'erreurs par caractères (CER) :

$$\frac{Ins.(caractères) + Subst.(caractères) + Sup.(caractères)}{nb\ total\ de\ caractères\ dans\ le\ texte\ de\ référence} = \frac{DLev_{caractères}(C_1, C_2)}{N\ total\ caractères\ C_1}$$

- Taux d'erreurs par mots (WER) :

$$\frac{Ins.(mots) + Subst.(mots) + Sup.(mots)}{nb\ total\ de\ mots\ dans\ le\ texte\ de\ référence} = \frac{DLev_{mots}(C_1, C_2)}{N\ total\ mots\ C_1}$$

Dans KaMI, le WER et le CER sont donnés dans le tableau de score final.

C1 (vérité terrain) : "Je suis à une conférence à la BnF."

C2 (prédiction) : "Jee suis une visoconférence depuis la BnFF."

CER = 14 / 34 = 0,4117 * 100 ≈ **41,17 %**

WER = 5 / 8 = 0,625 * 100 ≈ **62,5 %**

02

Utilisation de l'application KaMI

Principe de fonctionnement de l'application

1) Récupérer vos vérités terrains et vos prédictions sur *eScriptorium*

2) Pour essayer KaMI avec votre *dataset* : https://kami-app.herokuapp.com/

03

Une interprétation des scores

Comparaison modèle mixte / affiné (*finetuned*)

PAGE 1 : FRAN_0187_16402_L-0

modèle mixte (acc. 90.8 %)

	Default	Ignoring digits	Ignoring case	Ignoring punctuation	Ignoring diacritics	Combining all options
Levensthein Distance (Char.)	1158	1084	1145	1076	1003	833
Levensthein Distance (Words)	525	466	524	488	472	369
Hamming Distance	Ø	Ø	Ø	Ø	Ø	Ø
Word Error Rate (WER)	62.799	65.912	62.679	59.73	56.459	57.032
Char. Error Rate (CER)	23.758	24.436	23.491	23.3	20.57	19.918
Word Accuracy (Wacc)	37.2	34.087	37.32	40.269	43.54	42.967
Match Error Rate (MER)	23.206	23.824	22.945	22.743	20.336	19.706
Char. Information Lost (CIL)	35.679	36.429	35.242	35.022	31.1	30.071
Char. Information Preserved (CIP)	64.32	63.57	64.757	64.977	68.899	69.928
Hits	3832	3466	3845	3655	3929	3394
Substitutions	736	680	723	684	610	500
Deletions	306	290	306	279	337	288
Insertions	116	114	116	113	56	45

modèle Rigault affiné (acc. 94.6 %)

	Default	Ignoring digits	Ignoring case	Ignoring punctuation	Ignoring diacritics	Combining all options
Levensthein Distance (Char.)	733	683	724	704	557	469
Levensthein Distance (Words)	401	356	398	375	341	264
Hamming Distance	Ø	Ø	Ø	Ø	Ø	Ø
Word Error Rate (WER)	47.966	50.353	47.607	45.899	40.789	40.803
Char. Error Rate (CER)	15.038	15.396	14.854	15.244	11.423	11.214
Word Accuracy (Wacc)	52.033	49.646	52.392	54.1	59.21	59.196
Match Error Rate (MER)	14.668	14.981	14.488	14.871	11.328	11.113
Char. Information Lost (CIL)	22.59	22.942	22.263	23.111	17.559	17.132
Char. Information Preserved (CIP)	77.409	77.057	77.736	76.888	82.44	82.867
Hits	4264	3876	4273	4030	4360	3751
Substitutions	432	396	423	428	328	271
Deletions	178	164	178	160	188	160
Insertions	123	123	123	116	41	38

Comparaison modèle mixte / affiné (*finetuned*)

PAGE 3 : FRAN_0187_16419_L-1

modèle mixte (acc. 90.8 %)

	Default	Ignoring digits	Ignoring case	Ignoring punctuation	Ignoring diacritics	Combining all options
Levensthein Distance (Char.)	665	629	649	625	483	389
Levensthein Distance (Words)	429	393	424	411	373	308
Hamming Distance	Ø	Ø	Ø	Ø	Ø	Ø
Word Error Rate (WER)	66.822	76.908	66.043	64.319	58.099	64.435
Char. Error Rate (CER)	15.028	15.677	14.666	14.754	10.907	10.167
Word Accuracy (Wacc)	33.177	23.091	33.956	35.68	41.9	35.564
Match Error Rate (MER)	14.365	14.933	14.02	14.054	10.615	9.853
Char. Information Lost (CIL)	21.175	21.935	20.538	20.652	15.43	14.054
Char. Information Preserved (CIP)	78.824	78.064	79.461	79.347	84.569	85.945
Hits	3964	3583	3980	3822	4067	3559
Substitutions	337	316	321	313	228	171
Deletions	124	113	124	101	133	96
Insertions	204	200	204	211	122	122

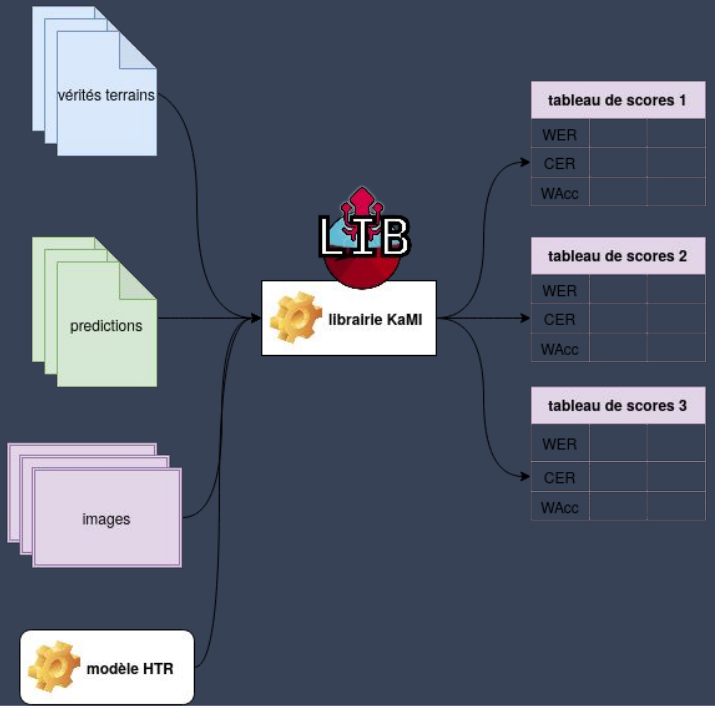
modèle Rigault affiné (acc. 94.6 %)

	Default	Ignoring digits	Ignoring case	Ignoring punctuation	Ignoring diacritics	Combining all options
Levensthein Distance (Char.)	460	436	449	436	263	204
Levensthein Distance (Words)	261	234	256	250	193	147
Hamming Distance	Ø	Ø	Ø	Ø	Ø	Ø
Word Error Rate (WER)	40.654	45.792	39.875	39.123	30.062	30.753
Char. Error Rate (CER)	10.395	10.867	10.146	10.292	5.939	5.331
Word Accuracy (Wacc)	59.345	54.207	60.124	60.876	69.937	69.246
Match Error Rate (MER)	10.083	10.508	9.842	9.972	5.878	5.269
Char. Information Lost (CIL)	15.083	15.591	14.627	15.007	8.569	7.461
Char. Information Preserved (CIP)	84.916	84.408	85.372	84.992	91.43	92.538
Hits	4102	3713	4113	3936	4211	3667
Substitutions	239	221	228	231	123	86
Deletions	84	78	84	69	94	73
Insertions	137	137	137	136	46	45

Aller plus loin avec KaMI

Code source du package Python KaMI : https://gitlab.inria.fr/dh-projects/kami/kami-lib

Notebook-Tutoriel pas-à-pas pour utiliser le *package* KaMI : https://cutt.ly/WT3Ahx1



Merci de votre attention