Trois recettes d'apprentissage automatique pour un système d'extraction d'information et de classification de recettes de cuisine *

Eric Charton[1] Ludovic Jean-Louis[1] Marie-Jean Meurs[2] Michel Gagnon[1]
(1) WeST, Ecole Polytechnique de Montréal
(2) Centre for Structural and Functional Genomics, Concordia University
Montréal, QC, Canada
{eric.charton, michel.gagnon, ludovic.jean-louis}@polymtl.ca,
marie-jean.meurs@concordia.ca

RÉSUMÉ

Cet article présente la participation du Wikimeta Lab au Défi Fouille de Texte (DeFT) 2013. En 2013, le défi s'intéresse à la fouille de recettes de cuisine en langue française et se décline en trois tâches de classification et une tâche d'extraction d'information. Les recettes de cuisine sont issues d'un site internet collaboratif et sont donc conçues par des internautes, avec très peu de contraintes quant à la définition des différents paramètres, ingrédients et commentaires qui les composent. Cette particularité rend certaines caractéristiques des corpus difficiles à modéliser dans un système d'apprentissage automatique, faute de régularité dans les données fournies. Nous expliquons dans cette communication notre démarche pour élaborer malgré cette contrainte, plusieurs systèmes ayant obtenus des résultats intéressants dans les tâches 1, 2 et 4.

ABSTRACT

Wikimeta Lab participation in DeFT 2013 - Machine Learning for Information Extraction and Classification of Cooking Recipes

This paper presents Wikimeta Lab participation in the Défi Fouille de Texte (DeFT) 2013. In 2013, this evaluation campaign is focused on mining cooking recipes in French. The campaign consists of three classification tasks and an information extraction task. The corpus is composed of recipes from a collaborative web site, thus they are written by users with almost no constraints on labels, ingredients, and comments. This very context makes some of the corpus specificities difficult to model for a machine learning based system. In this paper, we explain our approach for buiding relevant and effective systems dealing with such a corpus.

MOTS-CLÉS : classification, extraction d'information, DeFT2013, recette de cuisine.

KEYWORDS: classification, information extraction, DeFT2013, cooking recipe.

*. Les auteurs remercient Wikimeta Technologies Inc d'avoir mis ses moyens, son laboratoire et sa cuisine à leur disposition lors de ce défi.

1 Introduction

Cet article présente les systèmes que nous avons développés pour participer aux tâches 1, 2 et 4 du Défi Fouille de Texte (DeFT) 2013. Cette année, le défi s'intéresse à la fouille de recettes de cuisine en langue française. Le corpus est composé de recettes extraites du site Marmiton.org (`http://www.marmiton.org/`). Le défi propose trois tâches de classification (tâches 1 à 3) et une tâche d'extraction d'information (tâche 4). La tâche 1 consiste à identifier à partir du titre et du texte de la recette son niveau de difficulté sur une échelle à 4 niveaux (très facile, facile, moyennement difficile, difficile). La tâche 2 consiste à identifier le type de plat préparé (entrée, plat principal, dessert) à partir du titre et du texte de la recette. La tâche 3 consiste à apparier le texte d'une recette à son titre. Enfin, la tâche 4 consiste à extraire du titre et du texte d'une recette la liste de ses ingrédients. Notre système développé pour la tâche 1 (détection du niveau de difficulté) est classé premier sur six. Nos systèmes développés pour les tâches 2 (détection du type de plat) et 4 (extraction des ingrédients) sont classés seconds sur cinq.

Cet article est organisé comme suit : La section 2 décrit tout d'abord les corpus utilisés pour les tâches 1, 2 et 4 du défi. Les sections 3, 4 et 5 présentent ensuite les systèmes que nous avons développés pour traiter respectivement la première, la seconde et la quatrième tâche, ainsi que les résultats expérimentaux obtenus par ces systèmes sur le corpus d'entraînement. Enfin, la section 6 conclut cet article et propose des pistes d'investigations complémentaires.

2 Analyse du corpus

Le corpus utilisé pour le défi 2013 est composé de recettes de cuisine extraites du site Marmiton. Marmiton.org est l'un des sites culinaire francophone de large audience avec plus de 300.000 visiteurs par jour (chiffres Smart AdServer mars 2010, source : `http://www.marmiton.org/`). Les recettes rassemblées dans la base de données de Marmiton.org depuis 1999 sont proposées par les internautes via un formulaire validé pour publication par l'équipe de Marmiton. Lors de la soumission d'une recette, les internautes doivent indiquer le type de plat, le niveau de difficulté, le coût et le type de cuisson en sélectionnant des valeurs parmi des listes de choix pré-établies. Les paramètres numériques de la recette tels les temps de préparation et de cuisson, et le nombre de convives, sont à renseigner dans des champs contraints. Les ingrédients, les consignes de préparation et la boisson conseillée sont recueillis dans des champs en texte libre.

Le corpus d'entraînement contient 13.864 recettes pour un volume de données de 19,2 MB. Les corpus de test pour les tâches 1, 2 et 4 sont composés respectivement de 2309, 2307 et 2306 recettes pour des volumes de données de 3MB, 2,9MB et 2,2MB. Les recettes sont fournies au format XML. Chaque fichier du corpus d'entraînement contient le titre de la recette, son type, son niveau, son coût, la liste non normalisée de ses ingrédients et leur quantité d'usage, ainsi que les indications de préparation en texte libre.

2.1 Corpus associé à la tâche 1 : classification par niveau de difficulté

Le tableau 1 détaille le nombre de recettes du corpus d'entraînement selon leur niveau. Les quatre niveaux de difficulté possibles pour chaque recette sont présents de manière très inégale dans le

corpus d'apprentissage. On constate un fort déséquilibre entre les classes "Très facile" et "Facile" qui contiennent à elles seules plus de 90% des recettes, et les classes "Moyennement difficile" et "Difficile" qui représentent respectivement moins de 10% et 1% du corpus d'apprentissage.

Niveau	corpus d'entraînement		corpus de test	
	#recettes	% du corpus	#recettes	% du corpus
Très facile	6962	50,2	1132	49,0
Facile	5752	41,5	968	41,9
Moyennement difficile	1068	7,7	189	8,2
Difficile	80	0,6	20	0,9
Total	13862*		2309	

* Deux recettes sont incorrectement étiquetées *tres-facile*.

TABLE 1 – Répartition des recettes selon leur niveau de difficulté

2.2 Corpus associé à la tâche 2 : classification par type de plat

Le tableau 2 détaille le nombre de recettes du corpus d'entraînement selon leur type.

La répartition des classes de la tâche 2 dans le corpus d'apprentissage est plus homogène que pour la tâche 1. On constate cependant que presque une recette sur deux du corpus d'apprentissage décrit la réalisation d'un plat principal.

Type de plat	corpus d'entraînement		corpus de test	
	#recettes	% du corpus	#recettes	% du corpus
Entrée	3246	23,4	562	24,4
Plat principal	6449	46,5	1084	47,0
Dessert	4169	30,1	661	28,6
Total	13864		2307	

TABLE 2 – Répartition des recettes selon le type de plat préparé

2.3 Corpus associé à la tâche 4 : extraction des ingrédients

Pour cette tâche, la liste normalisée des ingrédients à extraire contient 960 ingrédients parmi lesquels 102 n'apparaissent pas dans le corpus d'entraînement et 298 n'apparaissent pas dans le corpus de test. Parmi ceux effectivement présents dans le corpus d'entraînement, 348 apparaissent dans 1 à 10 recettes, 355 apparaissent dans 10 à 100 recettes, 138 apparaissent dans 100 à 1000 recettes et enfin, 17 apparaissent dans plus de 1000 recettes.

Le tableau 3 montre la liste des 20 ingrédients les plus fréquents dans les corpus d'entraînement et de test, ainsi que le nombre de recettes dans lesquelles ils sont utilisés. Les données du tableau 3 sont en accord avec celles du tableau 2 et confirment, sans surprise, que les ingrédients prépondérants sont des éléments de base pour l'élaboration de plats principaux et de desserts (*sel, poivre, oignon, sucre, lait, farine, etc.*). On relève aussi que les ingrédients les plus fréquents sur le corpus de test sont identiques à ceux du corpus d'entraînement (les rangs sont quasi similaires).

Rang	corpus d'entraînement			corpus de test		
	ingrédient	# recettes	% du corpus	ingrédient	# recettes	% du corpus
1	sel	6981	50,4	sel	1122	48,7
2	poivre	6109	44,1	poivre	1009	43,8
3	oeuf	5570	40,2	oeuf	895	38,8
4	beurre	4237	30,6	beurre	733	31,8
5	oignon	3452	24,9	farine	577	25,0
6	farine	3249	23,4	oignon	568	24,6
7	huile-d-olive	2631	19,0	huile-d-olive	438	19,0
8	ail	2561	18,5	ail	430	18,6
9	sucre	2533	18,3	sucre	429	18,6
10	lait	2031	14,6	lait	327	14,2
11	tomate	1554	11,2	tomate	256	11,1
12	huile	1299	9,4	citron	236	10,2
13	persil	1267	9,1	huile	210	9,1
14	citron	1253	9,0	eau	202	8,8
15	eau	1236	8,9	persil	202	8,8
16	echalote	1083	7,8	carotte	183	7,9
17	carotte	1073	7,7	echalote	182	7,9
18	pomme-de-terre	961	6,9	pomme-de-terre	150	6,5
19	pomme	893	6,4	pomme	142	6,2
20	sucre-en-poudre	814	5,9	sucre-en-poudre	133	5,8

TABLE 3 – Liste des 20 ingrédients les plus fréquents dans les corpus d'entraînement et de test

2.4 Quelques perles

Le corpus d'apprentissage contient quelques recettes surprenantes, tant par leurs ingrédients que par leurs conditions de réalisation. Ainsi, la recette 17129 propose de cuisiner un *"gigôt bitume"* (plus exactement six) et requiert *"1 chantier de bâtiment ou de travaux publics"* :

```
<recette id="17129">
<titre>Gigot bitume</titre>
<type>Plat principal</type>
<niveau>Difficile</niveau>
<cout>Moyen</cout>
<ingredients>
  <p>6 gigots d'agneau</p>
  <p>fines herbes, dont thym</p>
  <p>sel, poivre</p>
</ingredients>
<preparation>
Matériel : 1 chantier de bâtiment ou de travaux publics
Envelopper les gigots assaisonnés dans plusieurs couches serrées de papier
kraft-aluminium utilisé en bâtiment. Entourer généreusement de fil de fer pour assurer
que l'ensemble ne se défasse pas. Plonger pendant 25 mn les gigots ainsi préparés dans
le baril de bitume brûlant destiné à l'étanchéité de la terrasse du bâtiment ou au
revêtement de la route en construction. Retirer et défaire avec précaution.
Excellent plat traditionnel de BTP mais qui ne relève pas de la cuisine
familiale. Je le donne pour information suite à l'appel aux gourmands. Rouge
</preparation>
</recette>
```

On citera également la recette 10635 du *Gâteau au fromage* dont la réussite impose de *"psalmodier gentiment des incantations magiques"* telles "*abracadabragrandmèraidemoi, oulalajamaisjepourraijamaismangertoutçajenaidéjàpleinlesdoigts...*".

3 Tâche 1 - Identification du niveau de difficulté d'une recette

Les systèmes de classification élaborés pour les tâches 1 et 2 sont à base d'apprentissage automatique. Leurs performances sont donc conditionnées par les qualités discriminantes des caractéristiques sélectionnées pour leur entraînement. Pour résoudre les deux tâches de classification de ce défi, nous avons élaboré un ensemble de paramètres discriminants. Ces paramètres peuvent être aussi bien de type classique en fouille de texte (n-grammes, éléments de champs lexicaux), mais aussi plus spécifiques à la tâche, tels les noms d'ingrédients normalisés, le nombre de mots contenus dans une section (le titre, le descriptif de recette) ou encore les quantités d'ingrédients.

3.1 Caractéristiques discriminantes pour la tâche 1

La tâche 1 concerne l'identification d'un niveau de difficulté dans une recette. La particularité de cette tâche est que l'étiquette de difficulté fournie dans le corpus de référence ne provient pas d'un processus d'annotation classique (avec guide fourni aux annotateurs) mais d'un procédé collaboratif : chaque lecteur qui soumet sa recette sur le site Marmiton est laissé entièrement libre quant à la détermination de son degré de difficulté. Ce critère étant hautement subjectif et relié à l'expertise de l'auteur, il en résulte une définition de difficulté extrêmement variable selon les fiches et difficile à modéliser. Les 78 caractéristiques finalement retenues pour l'entraînement des classifieurs pour la tâche 1 sont les suivantes :

- le nombre de mots du titre,
- le nombre de mots des consignes de préparation,
- le nombre d'ingrédients,
- le coût,
- un sous-ensemble de mots discriminants du vocabulaire spécifique de la classe *Moyennement difficile*,
- un sous-ensemble de trigrammes de mots discriminants,
- le nombre de verbes utilisés dans les consignes pour trois familles de verbes.

Mots discriminants du vocabulaire spécifique de la classe *Moyennement difficile*.
Les mots discriminants du vocabulaire spécifique de la classe *Moyennement difficile* sont obtenus en extrayant tout d'abord les mots du titre et des consignes de préparation pour chaque classe. Ces listes sont ensuite filtrées à l'aide d'un anti-dictionnaire permettant de retirer les mots vides. On obtient alors M_{TF}, M_F, M_{MD} et M_D les vocabulaires associés respectivement aux classes *Très facile* (TF), *Facile* (F), *Moyennement difficile* (MD) et *Difficile* (D).

Pour chaque mot m dans M_c où $c \in \{TF, F, MD, D\}$, on calcule les fréquences d'apparition par classe F^m_{TF}, F^m_F, F^m_{MD} et F^m_D.

La classe *Difficile* étant très peu représentée dans le corpus, nous avons choisi de ne pas la considérer et d'orienter notre sélection vers le vocabulaire spécifique de la classe *Moyennement difficile*.

Ainsi, pour chaque mot m, nous avons calculé Δ^m_{MD}, la différence moyenne de fréquence d'apparition entre la classe *Moyennement difficile* et les classes *Très facile* et *Facile* :

$$\Delta^m_{MD} = \frac{2F^m_{MD} - F^m_{TF} - F^m_F}{2}$$

On obtient alors l'ensemble M^{disc}_{MD}, des mots que nous nommons ici "mots discriminants du vocabulaire spécifique de la classe *Moyennement difficile*", en sélectionnant tous les mots pour lesquels cette différence moyenne est supérieure à 2% :

$$M^{disc}_{MD} = \{m \in M_{MD}, \quad \Delta^m_{MD} \geq 2\%\}$$

Enfin, le sous-ensemble $M^{car}_{MD} \subset M^{disc}_{MD}$ est construit en utilisant l'algorithme CFS (Correlation based Feature Selection) (Hall, 1999) associé à un algorithme de recherche glouton. CFS évalue la valeur d'un sous-ensemble de caractéristiques en considérant leurs capacités de prédiction et leurs degrés de redondance.

Trigrammes de mots discriminants.
L'ensemble de tous les trigrammes présents dans les consignes de préparation du corpus d'apprentissage est extrait. Un sous-ensemble de trigrammes discriminants est ensuite construit en utilisant l'algorithme CFS mentionné précédemment.

Familles de verbes.
Les verbes contenus dans les recettes ont été regroupés en deux ensembles : d'une part ceux qui se retrouvent dans une recette *Facile* ou *Très facile* (ensemble TFF), d'autre part ceux qui se retrouvent dans une recette *Difficile* ou *Moyennement difficile* (ensemble MDD). Pour chaque verbe de chacun de ces deux ensembles, on calcule son nombre total d'apparitions dans les recettes correspondant aux deux niveaux de difficulté. Puis, pour chaque verbe de chaque ensemble, on normalise sa fréquence en la divisant par la somme totale de fréquences pour tous les verbes. Le même procédé a été appliqué en ne prenant que l'ensemble des recettes *Très facile* d'une part (ensemble TF), et l'ensemble MDD d'autre part.

Par la suite, trois ensembles de verbes discriminants ont été obtenus selon leur ratio de fréquence normalisée. Pour chaque ratio α on a un ensemble de verbes discriminants $VD(\alpha)$ tel que :

$$VD(\alpha) = \left\{ \quad v \in MDD \cup TFF \quad \middle| \begin{array}{l} \alpha \times FN_{MDD}(v) \leq FN_{TFF}(v) \leq (1/\alpha) \times FN_{MDD}(v) \\ \vee \\ \alpha \times FN_{MDD}(v) \leq FN_{TF}(v) \leq (1/\alpha) \times FN_{MDD}(v) \end{array} \right\}$$

où FN_i est la fréquence normalisée du verbe v dans l'ensemble i.

En retirant les verbes déjà contenus dans les ensembles extraits pour un α supérieur, on obtient les trois ensembles suivants :

$VD(20) =$ { tripler, peser }
$VD(14) =$ { accorder, étirer, manipuler, redéposer, redevenir, tempérer }
$VD(8) =$ { aérer, braiser, détendre, échapper, effectuer, masquer, renverser, reprendre }

3.2 Système 1T1 pour la tâche 1

Le système 1T1 est basé sur un réseau bayésien (Pearl, 1986, 1998) dont la structure est apprise sur l'ensemble de caractéristiques décrit en 3.1 (78 variables discrètes) en utilisant l'algorithme K2 (Cooper et Herskovits, 1992), et dont la distribution de probabilités est calculée sur le corpus d'apprentissage par un estimateur simple.

Soit $X_1, ..., X_n$, variables aléatoires discrètes définies par leur loi jointe P. Ces variables sont représentées par les nœuds $v_i \in V$ du graphe orienté $G(V,E)$ associé au réseau bayésien.

Les arcs $e_i \in E$ du graphe associé au réseau bayésien représentent les dépendances entre les variables. On dit que $u \in V$ est un parent de $v \in V$ si $(u,v) \in E$. L'ensemble des nœuds parents d'un nœud v est noté $pa(v)$. Les fils de v sont les nœuds dont v est un parent. Les descendants de v sont les nœuds fils de fils et leurs descendants.

La structure graphique d'un réseau bayésien satisfait le critère de *d-séparation* : toute variable est indépendante de tout sous-ensemble de ses non-descendants, conditionnellement à ses parents. Cette propriété permet d'écrire la loi jointe sous la forme : $P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i|pa(X_i)$ qui définit complètement le réseau bayésien.

Les expériences ont été réalisées en utilisant les classes BayesNet dans Weka (Hall *et al.*, 2009).

3.2.1 Résultats

Sur le corpus d'entraînement, avec une validation croisée de pas 5, le système 1T1 obtient les résultats présentés dans les tableaux 4 et 5.

Classe	Précision	Rappel	F-mesure
Très facile	0,675	0,753	0,712
Facile	0,575	0,533	0,554
Moyennement difficile	0,389	0,248	0,303
Difficile	0,207	0,207	0,207
Moyenne pondérée	0,609	0,620	0,612

TABLE 4 – Résultats obtenus par le système 1T1 sur le corpus d'entraînement pour la tâche 1

	Classe estimée			
Classe réelle	Très facile	Facile	Moyennement difficile	Difficile
Très facile	5244	1652	59	7
Facile	2336	3068	331	17
Moyennement difficile	181	581	265	41
Difficile	8	31	26	17

TABLE 5 – Matrice de confusion du système 1T1 sur le corpus d'entraînement pour la tâche 1

Sur le corpus de test, le système 1T1 a obtenu les résultats présentés dans le tableau 6 :

Macro évaluation			Micro évaluation		
Précision	Rappel	F-mesure	Précision	Rappel	F-mesure
0,460	0,419	0,438	0,609	0,609	0,609

TABLE 6 – Résultats obtenus par le système 1T1 sur le corpus de test pour la tâche 1

3.3 Système 2T1 pour la tâche 1

Le système 2T1 est basé sur l'algorithme Logistic Model Tree (LMT) (Landwehr *et al.*, 2005) qui combine arbres de décision et modèles de régression logistique.

Un LMT est un arbre composé d'une structure d'arbre de décision standard de taille réduite, avec des fonctions de régression logistique (Collins *et al.*, 2002) au niveau des feuilles. Les paramètres des fonctions de régression logistique sont calculés pour maximiser les probabilités sur les données observées. Le LMT est un classifieur probabiliste dont les résultats sont généralement pertinents lorsque l'on dispose de peu de données d'apprentissage. La structure d'arbre permet de minimiser les erreurs d'entraînement tandis que la régression logistique évite le sur-apprentissage en limitant la taille de l'arbre.

Les expériences ont été réalisées en utilisant les classes LMT (Landwehr *et al.*, 2005; Sumner *et al.*, 2005) dans Weka (Hall *et al.*, 2009).

3.3.1 Résultats

Sur le corpus d'entraînement, avec une validation croisée de pas 5, le système 2T1 obtient les résultats présentés dans les tableaux 7 et 8.

Classe	Précision	Rappel	F-mesure
Très facile	0,671	0,777	0,720
Facile	0,587	0,561	0,574
Moyennement difficile	0,549	0,147	0,232
Difficile	0,400	0,073	0,124
Moyenne pondérée	0,625	0,635	0,618

TABLE 7 – Résultats obtenus par le système 2T1 sur le corpus d'entraînement pour la tâche 1

	Classe estimée			
Classe réelle	Très facile	Facile	Moyennement difficile	Difficile
Très facile	5406	1534	22	0
Facile	2446	3228	76	2
Moyennement difficile	197	707	157	7
Difficile	12	33	31	6

TABLE 8 – Matrice de confusion du système 2T1 sur le corpus d'entraînement pour la tâche 1

Sur le corpus de test, le système 2T1 a obtenu les résultats présentés dans le tableau 9 :

Macro évaluation			Micro évaluation		
Précision	Rappel	F-mesure	Précision	Rappel	F-mesure
0,682	0,375	0,484	0,625	0,625	0,625

TABLE 9 – Résultats obtenus par le système 2T1 sur le corpus de test pour la tâche 1

4 Tâche 2 - Identification du type de plat préparé

Le corpus d'entrainement proposé pour la tâche de détection du type de plat est beaucoup plus homogène que celui de détection de difficulté. L'analyse multivariée des distributions de n-grammes ou des noms d'ingrédients laisse apparaître des marqueurs très spécifiques tels que les noms de fruits pour les desserts, ou les noms de viande pour les plats principaux. Nous avons donc décidé de faire reposer notre approche de classification principalement sur des paramètres de type ingrédients (qui représentent 1.200 paramètres sur les 1.287 sélectionnés).

4.1 Caractéristiques discriminantes

Les 1.287 caractéristiques retenues pour l'entraînement du classifieur pour la tâche 2 sont les suivantes :
– le nombre de mots du titre,
– le nombre de mots des consignes de préparation,
– le nombre d'ingrédients,
– le coût,
– les ingrédients normalisés et sélectionnés par analyse en composantes principales (ACP),
– un sous-ensemble de trigrammes de mots discriminants,
– le nombre de verbes utilisés dans les consignes pour trois familles de verbes.

Sélection de marqueurs d'ingrédients par ACP.
Notre approche de sélection d'une liste d'ingrédients consiste à collecter la totalité des noms d'ingrédients fournis dans la section dédiée du corpus, puis à conserver les plus fréquents. Nous collectons 3.860 unités lexicales telles que *daurade* ou *jus de citron*. Puis nous utilisons chacune de ces unités en tant que détecteur binaire à l'aide d'expressions régulières que nous appliquons sur chacun des textes des recettes. Nous obtenons ainsi 13.864 vecteurs (un par recette) de 3.860 paramètres. Nous soumettons ces 13.864 vecteurs à une analyse en composantes principales configurée pour regrouper les paramètres par groupes de 5 et les ordonner par potentiel discriminant. Il est ainsi possible d'identifier 1.232 paramètres fortement discriminants qui sont conservés en tant que paramètres d'ingrédients normalisés et utilisés pour construire le corpus d'entraînement final.

4.2 Système pour la tâche 2

Le système développé pour traîter la tâche 2 est basé sur un Séparateur à Vaste Marge (SVM, Support Vector Machine) (Vapnik, 1995) à noyau linéaire. Ce choix est motivé par l'aptitude des méthodes à base de SVM à traiter des problèmes de grande dimension. Essentiellement dépendantes des vecteurs supports, elles produisent des résultats pertinents même si les données d'apprentissage sont peu nombreuses. Elles offrent ainsi un bon compromis entre capacité de généralisation et complexité.

On dispose d'un ensemble X de n données étiquetées et d'un ensemble fini U de k classes . Dans le contexte de la tâche 2, $n = 13.864$ et $k = 3$. Chaque donnée $x_{i\in[\![1,n]\!]}$ est caractérisée par p caractéristiques et par sa classe $u_i \in U$. Les données sont décrites vectoriellement dans un espace de dimension p. Pour notre système, $p = 1287$, le classifieur est basé sur un SVM multiclasses (3 classes, une par type de plat) avec un l'approche un-contre-un.

Les expériences ont été réalisées en utilisant les classes libSVM (Chang et Lin, 2011; El-Manzalawy et Honavar, 2005) dans Weka (Hall *et al.*, 2009).

4.3 Résultats

Sur le corpus d'entraînement, avec une validation croisée de pas 5, le système T2 obtient les résultats présentés dans les tableaux 10 et 11.

Classe	Précision	Rappel	F-mesure
Plat principal	0,834	0,854	0,844
Dessert	0,967	0,982	0,974
Entrée	0,694	0,648	0,670
Moyenne pondérée	0,841	0,844	0,842

TABLE 10 – Résultats obtenus par le système T2 sur le corpus d'entraînement pour la tâche 2

	Classe estimée		
Classe réelle	Plat principal	Dessert	Entrée
Plat principal	5507	60	882
Dessert	29	4094	46
Entrée	1064	80	2102

TABLE 11 – Matrice de confusion du système T2 sur le corpus d'entraînement pour la tâche 2

Sur le corpus de test, le système T2 a obtenu les résultats présentés dans le tableau 12 :

Macro évaluation			Micro évaluation		
Précision	Rappel	F-mesure	Précision	Rappel	F-mesure
0,850	0,843	0,847	0,856	0,856	0,856

TABLE 12 – Résultats obtenus par le système T2 sur le corpus de test pour la tâche 2

5 Tâche 4 - Extraction des ingrédients

Parmi les ingrédients à extraire, certains ne se retrouvent pas explicitement dans les consignes de préparation : ils peuvent être remplacés par des formes verbales (*saler*, au lieu de *sel*), nominales (*légume* pour *carotte*, ou *viande* pour *escalopes de poulet*), ou encore omis par les utilisateurs. Dans quelques rares cas, les préparations sont très courtes et ne mentionnent pas d'ingrédients : *"Mélangez tous les ingrédients et faire chauffer dans une poêle à crêpes."* (recette *20827*). A l'opposé certains ingrédients présents dans la description peuvent ne pas faire partie de la recette, par exemple lorsqu'il s'agit de suggestions d'accompagnement pour le plat ou d'une proposition alternative (*"glace à la vanille ou une crème anglaise"*).

En analysant les recettes du corpus d'entraînement et la liste des ingrédients à extraire, nous avons mesuré que dans 39,8% des cas un ingrédient à extraire n'était pas mentionné explicitement dans la préparation. Ce chiffre est une moyenne sur tous les ingrédients, et varie selon les ingrédients.

Notre méthode pour l'extraction des ingrédients s'appuie sur le titre et la description de chaque recette et se compose de trois étapes : (i) la normalisation des recettes, (ii) la détection des ingrédients, (iii) la sélection des ingrédients.

Nous avons soumis deux jeux de résultats pour cette tâche 1T4 et 2T4. Les deux premières étapes sont identiques pour les deux soumissions ; seule la phase de sélection des ingrédients diffère. Nous détaillons ci-après chacune des étapes et les soumissions.

5.1 Normalisation des recettes

La phase de normalisation vise à faire une analyse linguistique du contenu textuel des recettes. Précisément, on utilise le lemmatiseur du TreeTagger [1] pour extraire les lemmes de tous les termes d'une recette. Les lemmes sont ensuite désaccentués.

5.2 Détection des ingrédients

La phase de détection vise à identifier tous les ingrédients de la liste normalisée qui sont contenus dans une recette. En pratique, on utilise des expressions régulières pour rechercher chaque ingrédient de cette liste dans les versions normalisées des recettes (sections titre et préparation). En complément, nous avons ajouté des expressions régulières pour substituer à certaines formes verbales ou nominales les ingrédients correspondants : par exemple, "beurrée|beurrez|beurrer" sont remplacés par "beurre". Au total, la détection s'appuie sur une trentaine d'expressions régulières.

5.3 Sélection des ingrédients

La phase de sélection vise à choisir parmi tous les ingrédients détectés dans la phase précédente ceux effectivement retenus pour la recette. Nous avons proposé deux méthodes pour cette sélection. La première, utilisée par le système 1T4, est fondée sur deux critères : la fréquence et la position de la première occurrence de chaque ingrédient dans la recette. Précisément, les ingrédients sont triés par ordre décroissant de fréquence et ordre croissant d'apparition dans la recette.

La seconde méthode, utilisée par le système 2T4, est à base d'apprentissage statistique. La décision d'attribuer ou non un ingrédient à une recette est prise par un classifieur. Le classifieur s'appuie sur les caractéristiques suivantes pour prendre sa décision :
– la présence/absence de l'ingrédient dans le titre de la recette
– la forme normalisée de l'ingrédient
– le nombre d'occurrences de l'ingrédient dans la recette
– la position de la première occurrence de l'ingrédient dans le document : pos_{first}
– la position de la dernière occurrence dans le document : pos_{last}
– le taux de recouvrement du document : $rec_{last} = (pos_{last} - pos_{first})/taille(recette)$
– la profondeur : $prof = (1 - pos_{first})/taille(recette)$
Pour entraîner le modèle de sélection des ingrédients, nous avons testé plusieurs algorithmes d'apprentissage proposés dans l'outil Weka (Hall *et al.*, 2009) : arbres de décisions, Naïve Bayes, BayesNet et Meta Bagging. Les meilleurs résultats ont été obtenus avec les arbres de décisions et sont reportés dans le tableau 13.

1. http ://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

Classe	Précision	Rappel	F-Mesure
Sélectionné	0,806	0,846	0,825
Non sélectionné	0,806	0,759	0,782
Moyenne pondérée	0,806	0,806	0,806

TABLE 13 – Résultats de l'entraînement du modèle de sélection des ingrédients de la tâche 4

Nous reportons dans le tableau 14, les résultats globaux de l'extraction d'ingrédients, c'est à dire en appliquant nos deux approches de sélection d'ingrédients sur les corpus d'entraînement et de test. Les résultats sont reportés en terme de MAP (Mean Average Precision) et obtenus à partir du script d'évaluation officiel.

Système	corpus d'entraînement	corpus de test
1T4	0,5659	0,5678
2T4	0,6702	0,6462

TABLE 14 – Résultats sur l'extraction des ingrédients pour les deux approches proposées (MAP)

6 Conclusion

Nous avons présenté dans cet article trois systèmes dédiés à la résolution des tâches 1, 2 et 4 du Défi Fouille de Texte (DeFT) 2013. La tâche de fouille de texte proposée a été résolue par des approches par apprentissage automatique, impliquant néanmoins une recherche et une sélection poussée des caractéristiques. La particularité de la tâche 1, dont l'étiquette de difficulté était apposée de manière collaborative et non normalisée, rend l'apprentissage délicat. Nous avons tenté de résoudre ce problème en utilisant une méthode de classification originale qui repose sur l'algorithme Logistic Model Tree (LMT) combinant arbres de décision et modèles de régression logistique. Cette méthode a fourni des résultats intéressants. A l'issue de ce défi, l'utilisation des bases documentaires collaboratives comme ressource d'apprentissage, ainsi que la recherche d'algorithmes de classification appropriés pour traiter leurs données, sont des questions qui nous semblent appeler de plus amples investigations. Les systèmes que nous avons développés pour cette campagne, conçus en langage Java et exploitant la plateforme Weka, sont disponibles en version source libre sur `http://www.wikimeta.org/deft2013`. Ils permettent de reproduire la totalité des expériences de classification décrites dans cette communication.

Références

CHANG, C.-C. et LIN, C.-J. (2011). Libsvm : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

COLLINS, M., SCHAPIRE, R. E. et SINGER, Y. (2002). Logistic regression, adaboost and bregman distances. *Machine Learning*, 48(1-3):253–285.

COOPER, G. F. et HERSKOVITS, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347.

EL-MANZALAWY, Y. et HONAVAR, V. (2005). Wlsvm : Integrating libsvm into weka environment. *Software available at http ://www. cs. iastate. edu/yasser/wlsvm*.

HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P. et WITTEN, I. H. (2009). The weka data mining software : an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

HALL, M. A. (1999). *Correlation-based feature selection for machine learning*. Thèse de doctorat, The University of Waikato.

LANDWEHR, N., HALL, M. et FRANK, E. (2005). Logistic model trees. *Machine Learning*, 59(1-2):161–205.

PEARL, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3):241–288.

PEARL, J. (1998). *Bayesian networks*. MIT Press, Cambridge, MA, USA.

SUMNER, M., FRANK, E. et HALL, M. (2005). Speeding up logistic model tree induction. *In Knowledge Discovery in Databases : PKDD 2005*, pages 675–683. Springer.

VAPNIK, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag.