DEFT2013

Actes du neuvième DÉfi Fouille de Textes

Proceedings of the Ninth DEFT Workshop

21 juin 2013

Les Sables-d'Olonne, France

DEFT2013

Actes du neuvième DÉfi Fouille de Textes

21 juin 2013
Les Sables-d'Olonne, France

Comités

Comité de programme

Badin, Flora (INIST, Nancy)

Camelin, Nathalia (LIUM, Le Mans)

Daille, Béatrice (LINA, Nantes)

Forest, Dominic (EBSI, Université de Montréal)

Grouin, Cyril (LIMSI–CNRS, Orsay)

Paroubek, Patrick (LIMSI–CNRS, Orsay)

Zweigenbaum, Pierre (LIMSI–CNRS, Orsay)

Comité d'organisation

Badin, Flora (INIST, Nancy)

Forest, Dominic (EBSI, Université de Montréal)

Grouin, Cyril (LIMSI–CNRS, Orsay)

Paroubek, Patrick (LIMSI–CNRS, Orsay)

Zweigenbaum, Pierre (LIMSI–CNRS, Orsay)

Table des matières

Index

Programme — vendredi 21 juin 2013

Session I — Présentation et résultats

8.30–9.00 DEFT2013 se met à table : présentation du défi et résultats. *Cyril Grouin, Pierre Zweigenbaum et Patrick Paroubek*

Session II — Méthodes des participants

9.00–9.30 DEFT2013, une cuisine de caractères. *Gaël Lejeune, Charlotte Lecluze et Romain Brixtel*

9.30–10.00 Efficacité combinée du flou et de l'exact des recettes de cuisine. *Thierry Hamon, Amandine Périnet et Natalia Grabar*

10.00–10.30 *Pause*

10.30–11.00 Systèmes du LIA à DEFT 2013. *Xavier Bost, Ilaria Brunetti, Luis Adrián Cabrera-Diego, Jean-Valère Cossu, Andréa Linhares, Mohamed Morchid, Juan-Manuel Torres-Moreno, Marc El-Bèze et Richard Dufour*

11.00–11.30 Approches hybrides pour l'analyse de recettes de cuisine DEFT. *Luca Dini, André Bittar et Mathieu Ruhlmann*

11.30–12.00 Trois recettes d'apprentissage automatique pour un système d'extraction d'information et de classification de recettes de cuisine. *Eric Charton, Ludovic Jean-Louis, Marie-Jean Meurs et Michel Gagnon*

12.00–12.30 Participation de Orange Labs à DEFT 2013. *Olivier Collin, Aleksandra Guerraz, Yannick Hiou et Nicolas Voisine*

Session III — Discussion et conclusion

12.30–13.15 Discussion sur l'édition 2013 et pistes pour l'édition 2014.

13.15 *Clôture de l'atelier*

Présentation et résultats

DEFT2013 se met à table : présentation du défi et résultats

Cyril Grouin[1,2] Pierre Zweigenbaum[1] Patrick Paroubek[1]
(1) LIMSI–CNRS, Orsay (2) INSERM U872 Eq 20 & UPMC, Paris
prenom.nom@limsi.fr

RÉSUMÉ
L'édition 2013 du défi fouille de texte (DEFT) a porté sur l'analyse des recettes de cuisine sous quatre angles différents : le niveau de difficulté d'une recette, le type de plat préparé, le titre, et les ingrédients nécessaires à la réalisation de la recette. Dans cet article, nous présentons le déroulement du défi au travers des corpus produits et des tests humains réalisés, ainsi que les résultats obtenus par les équipes ayant participé. Les mesures d'évaluation utilisées sont la distance relative moyenne à la solution pour la difficulté du plat et le couple précision/rappel pour le type de plat, la MRR (moyenne de l'inverse du rang) pour l'association titre/recette et la MAP (moyenne des précisions moyennes) pour la liste des ingrédients d'une recette.

ABSTRACT
DEFT2013 spills the beans: challenge presentation and results

The DEFT 2013 challenge focused on the analysis of cooking recipes through four points of view: the level of difficulty of the recipe, the type of dish to be prepared, the title, and the ingredients needed to prepare the recipe. In this paper, we present the challenge: the corpora we produced, the human tests, as well as the results each participant achieved. The evaluation measures we used are the mean distance to the solution for the difficulty task, precision and recall for the type of dish, mean reciprocal rank (MRR) for the title/recipe association and mean average precision (MAP) for the list of ingredients.

MOTS-CLÉS : campagne d'évaluation, classification, mesures, recettes.

KEYWORDS: evaluation campaign, classification, metrics, recipes.

1 Présentation

1.1 Introduction

Le défi DEFT est un atelier annuel d'évaluation francophone en fouille de textes. La neuvième édition de ce défi a porté sur l'analyse des recettes de cuisine rédigées en langue française. Dans le cadre de cette nouvelle édition, nous avons orienté le défi vers un nouveau domaine d'application. La thématique retenue, les recettes de cuisine, a déjà fait par le passé l'objet d'une campagne d'évaluation *(Computer Cooking Contest*[1]*)*. Nous nous intéressons dans DEFT2013 à deux types de fonction d'analyse du langage, la classification de documents *(tâches 1 à 3)* et l'extraction d'information *(tâche 4)*, ceci dans un domaine de spécialité.

1. http://computercookingcontest.net

Pour cette nouvelle édition du défi, nous proposons quatre tâches d'analyse concernant les recettes de cuisine : identifier le niveau de difficulté de réalisation d'une recette, identifier le type de plat préparé, apparier une recette à son titre, identifier les ingrédients d'une recette.

1.2 Organisation

Une étape de réflexion relative au choix des thématiques et des tâches à proposer a été engagée en fin d'année 2012. Durant cette période, des tests de tâches ont été effectués auprès d'étudiants. Ces tests nous ont permis de confirmer les choix effectués, ou dans certains cas, de réorienter certaines tâches envisagées. Nous avons orienté cette nouvelle édition vers les recettes de cuisine du site Marmiton, [2] dont les catégories d'informations proposées correspondaient aux thématiques que nous souhaitions traiter dans ce défi et parce que les recettes sont librement accessibles.

Deux appels à participation ont été lancés sur la liste de diffusion de la communauté du traitement automatique des langues LN les 16 février et 5 mars. Les données d'apprentissage ont été mises à la disposition des participants à partir du 28 février, les données de test l'ont été entre le 25 avril et le 5 mai, dans une fenêtre de trois jours au libre choix de chaque participant. La référence et les résultats individuels *(avec classement, moyenne et médiane)* ont été communiqués aux participants le 6 mai.

Dix équipes se sont inscrites à cette édition du défi, parmi lesquelles deux sont issues du monde industriel. Les six équipes qui ont participé aux tests sont les suivantes :
– **Celi France :** Luca DINI, André BITTAR, Mathieu RUHLMANN ;
– **GREYC :** Gaël LEJEUNE, Charlotte LECLUZE, Romain BRIXTEL ;
– **LIA :** Xavier BOST, Ilaria BRUNETTI, Luis Adrián CABRERA-DIEGO, Jean-Valère COSSU, Andréa LINHARES, Mohamed MORCHID, Juan-Manuel TORRES-MORENO, Marc EL BÈZE, Richard DUFOUR ;
– **LIM&Bio :** Thierry HAMON, Natalia GRABAR, Amandine PÉRINET ;
– **Orange Labs :** Olivier COLLIN, Aleksandra GUERRAZ, Yannick HIOU, Nicolas VOISINE ;
– **Wikimeta Lab :** Eric CHARTON, Ludovic JEAN-LOUIS, Marie-Jean MEURS, Michel GAGNON.

2 Présentation

2.1 Tâches proposées

Nous avons proposé quatre tâches autour de la thématique des recettes de cuisine. La motivation des deux premières tâches concerne le jugement appliqué à une recette. Dans de nombreux cas, le type de plat, et plus encore, le niveau de difficulté de réalisation d'une recette peut se révéler variable d'un individu à un autre, avec pour conséquence une indexation biaisée des recettes présentes sur le site. L'indexation des recettes selon les ingrédients qu'elle contient est également une tâche complexe, qui fait appel à la distinction entre ingrédients obligatoires et ingrédients facultatifs. Enfin, l'appariement d'un titre avec la recette qui lui correspond permet de mettre en évidence les désaccords qui peuvent parfois exister dans ces associations. L'objectif global visé par les quatre tâches du défi vise ainsi à fournir des méthodes adaptées à l'indexation de sites web participatifs, dont le contenu, déposé par chaque utilisateur, est généralement classé et

2. http://www.marmiton.org/

évalué par l'utilisateur qui aura déposé le contenu en ligne, avec des écarts d'appréciation de classement assez importants.

Tâche 1 — Niveau de difficulté de réalisation d'une recette. La première tâche a pour but d'évaluer la capacité d'un algorithme à inférer la difficulté d'une recette de cuisine en se basant sur toutes les informations qu'il est possible d'extraire à partir du texte de la recette et de son titre. Le niveau de difficulté de réalisation d'une recette est évalué sur une échelle à quatre valeurs : *très facile, facile, moyennement difficile, difficile*. La mesure d'évaluation principale retenue pour cette tâche est la distance moyenne à la solution, calculée en micro-moyenne sur une échelle à quatre valeurs. Nous avons aussi vérifié la corrélation des résultats ainsi obtenus avec ceux provenant d'une mesure de précision/rappel en micro-mesure.

Tâche 2 — Type de plat. La deuxième tâche propose de classer les recettes en fonction du type de plat préparé, selon une partition en trois classes : *entrée, plat principal, dessert*. Les mesures d'évaluation retenues sont la précision et le rappel calculées en micro-moyennes.

Tâche 3 — Appariement titre/recette. La troisième tâche demande au système de retrouver pour chaque texte de recette à traiter, son titre original dans une liste de titres de recettes.

Tâche 4 — Ingrédients d'une recette. La dernière tâche se démarque des précédentes car elle ne concerne pas la classification des recettes, mais l'extraction d'information. Il s'agit en effet dans cette tâche d'identifier la liste des ingrédients de la recette. Les ingrédients identifiés doivent être exprimés selon des libellés normalisés présents dans une liste globale fournie aux participants. Cette liste contient au moins tous les ingrédients à trouver dans la base des textes de recettes, mais peut aussi contenir des ingrédients qui ne sont présents dans aucune des recettes des corpus d'entraînement et de test.

2.2 Corpus

2.2.1 Présentation

Le corpus de recettes utilisées pour le défi provient du site de recettes participatif Marmiton. Sur ce site, chaque utilisateur dépose ses propres recettes. Ce type de dépôt implique que l'utilisateur renseigne lui-même certaines des caractéristiques liées à la recette, en particulier les informations de coût et de niveau de difficulté. Les participants ont été autorisés à utiliser des corpus de recettes complémentaire, à l'exclusion des recettes provenant de Marmiton.

2.2.2 Modalités de constitution

Collecte des données. Nous avons téléchargé les recettes du site pendant les deux premières semaines de février 2013, au moyen d'une chaîne effectuant les étapes suivantes :
- Collecte des 26 pages d'index des ingrédients utilisés dans les recettes du site ;
- Pour chaque ingrédient indexé dans les pages d'index précédentes, collecte des pages listant les recettes utilisant cet ingrédient ;
- Conversion des pages web téléchargées au format XML.

Au terme de cette phase de récupération, nous avons collecté 46 176 recettes que nous avons transformées en XML. La moitié de ces recettes a été aléatoirement conservée pour l'édition 2013 du défi, l'autre moitié pouvant servir de corpus pour l'année suivante.

Constitution de la référence. La référence des différentes tâches a été réalisée de manière automatique, sur la base des informations présentes dans les recettes.

– la référence des tâches 1 à 3 est directement inférée des informations présentes dans les recettes : *niveau de difficulté, type de plat, titre de la recette* ;
– la référence de la tâche 4 a été constituée lors de la collecte des données par la correspondance entre les pages d'index des ingrédients et les différentes recettes.

Concernant les trois premières tâches, la référence est issue des informations renseignées par l'utilisateur qui a posté la recette sur Marmiton. Ainsi, le niveau de difficulté de réalisation d'une recette est largement subjectif et dépend également du niveau d'expertise de chacun en cuisine ; d'autre part, l'auteur d'une recette aura tendance à minorer le niveau de difficulté de sa recette, de manière à inciter le plus grand nombre de visiteurs à tester la recette avec l'espoir d'obtenir des commentaires en retour. On observe ce comportement dans la répartition des recettes selon les quatre niveaux de difficultés (tableau 1 gauche). À l'inverse, la répartition en types de plat semble plus équilibrée et moins sujette à la subjectivité (tableau 1 droite). Nous sommes conscients de la présence d'erreurs dans les références mais nous avons considéré que ces erreurs étaient marginales et n'auraient pas un impact trop fort sur l'évaluation finale. En conséquence, aucune phase de nettoyage humain *(tâche forcément coûteuse)* n'a été réalisée sur les références.

Niveau	Nombre	Pourcentage	Type	Nombre	Pourcentage
Très facile	11 602	50,2 %	Entrée	5 407	23,4 %
Facile	9 584	41,5 %	Plat principal	10 742	46,5 %
Moyennement difficile	1 776	7,7 %	Dessert	6 945	30,1 %
Difficile	132	0,6 %			

TABLE 1 – Répartition des recettes par niveau de difficulté (gauche) et type de plat (droite)

En ce qui concerne la dernière tâche, chaque recette contient la liste des ingrédients fournie par l'utilisateur. Cependant, les ingrédients de cette liste ont une forme qui varie avec les recettes, et il serait très difficile d'arriver à prédire exactement le libellé utilisé par l'auteur. C'est pourquoi la référence a été constituée à partir de l'index des ingrédients des recettes, index qui constitue une liste de libellés normalisés d'ingrédients et qui associe à chacun l'ensemble des recettes qui l'utilisent. Si cette méthode a l'avantage de fournir une liste normalisée d'ingrédients pour chaque recette, elle présente néanmoins plusieurs défauts. D'une part, certains ingrédients d'une recette ont pu être oubliés lors de l'indexation, et seront considérés comme faux positifs s'ils sont (correctement) trouvés par un système. D'autre part, certains index peuvent être associés par erreur à une recette (c'est le cas du « maïs » mis comme index d'une recette dont un ingrédient contenait l'expression « mais compter un peu plus long pour la cuisson »), et ils compteront comme faux négatifs si un système ne les produit pas. À noter, la forme normalisée des ingrédients ne contient pas de caractères accentués (les diacritiques sont supprimés) ni d'espaces (ils sont remplacés par des tirets).

Répartition des données. Le corpus de l'édition 2013 totalise 23 094 recettes que nous avons ensuite réparties entre corpus d'apprentissage *(13 864 recettes)* et corpus de test *(9 230 recettes)* selon une répartition respectivement fixée à 60 %/40 % comme utilisée dans les précédentes campagnes de DEFT. Cette répartition a été effectuée de telle sorte que la proportion de recettes en termes de niveau de difficulté et de type de plat soit équivalente dans les deux corpus.

Le corpus d'apprentissage est commun à l'ensemble des quatre tâches proposées cette année, dans la mesure où les données de référence sont incluses dans les méta-informations. En ce qui concerne le corpus de test, nous l'avons segmenté en quatre parties égales *(soit environ 2 300 documents par partie)*, chaque partie étant réservée pour une tâche, en garantissant également que chaque partie conserve la même proportion de recettes en niveaux de difficulté et type de plat *(même si cet impératif ne se révèle nécessaire que pour les tâches 1 et 2)*. Il en ressort que les documents des corpus de test de chacune des quatre tâches sont distincts pour chaque tâche.

Formatage des données. Nous donnons en figure 1 un exemple de recette issue du corpus d'apprentissage. Chaque document se compose de deux parties principales :
– des méta-données contenant : le titre *(référence de la tâche 3)*, le type de plat *(référence de la tâche 2)*, le niveau de difficulté *(référence de la tâche 1)*, le coût *(information complémentaire fournie dans l'apprentissage et le test)*, et la liste des ingrédients renseignés par l'utilisateur ;
– la préparation de la recette contenant les différentes étapes.

En dehors du document, la liste des ingrédients normalisés *(référence de la tâche 4)* a été fournie. La liste des ingrédients normalisés qui indexent cette recette sont les suivants : *ail, basilic, bouillon, citron, echalote, huile-d-olive, parmesan, pignon, poivre, tomate, truite*. Notons que les ingrédients normalisés, parce qu'ils servent à indexer le contenu du site Marmiton, étaient déjà tous désaccentués, avec espaces et apostrophes remplacés par des tirets.

2.3 Tests humains

Nous avons mis à contribution les étudiants de l'EBSI [3] et du M2 Pro d'Ingénierie Linguistique de l'INaLCO [4] en novembre 2012. Les étudiants ont travaillé sur les quatre tâches que nous envisagions à l'époque : (i) évaluer le niveau de difficulté *(très facile, facile, moyennement difficile, difficile)* d'une recette ; (ii) prédire le type de plat *(entrée, plat, dessert)* ; (iii) relier un titre à la recette correspondante ; et (iv) identifier l'ingrédient ajouté à la recette d'origine.

Niveau de difficulté. Les étudiants de l'INaLCO ont pris une demi-heure pour étudier le corpus et prédire le niveau de difficulté. Les résultats de ces tests sont rassemblés dans le tableau 2. La bonne réponse *(niveau de difficulté renseigné dans Marmiton)* est identifiée, au minimum par aucun annotateur et au maximum par 7 annotateurs, avec un taux moyen de réussite sur l'ensemble du corpus de 37,0 %. Aucune recette n'est donc évaluée de la même manière par l'ensemble des annotateurs. La majorité des erreurs effectuées (57,1 %) ne correspond toutefois

3. Ecole de Bibliothéconomie et des Sciences de l'Information, Université de Montréal. Cours assuré par Dominic Forest auprès d'un groupe de douze étudiants.

4. Institut National des Langues et Civilisations Orientales, Paris. Cours assuré par Cyril Grouin pour les parcours « *Ingénierie Multilingue* » et « *Traductiques et gestion de l'information* » du M2 Pro d'Ingénierie Linguistique auprès d'un groupe de dix étudiants : Abdennour Goumiri, Ardas Khalsa, Arthur Boyer, Asma Benahmed, Hamza Affane, Julie Arnal, Laure Chancerelle, Maria Goryainova, Selen Şahin et Thomas Moraine. Tous ces étudiants se sont acquittés de cette tâche d'évaluation humaine avec abnégation, à quelques heures seulement du dîner... Nous les en remercions vivement.

```
<?xml version="1.0" encoding="utf-8"?>
<recette id="54562">
  <titre>Cuisses de poulet au miel cuites au four</titre>
  <type>Plat principal</type>
  <niveau>Très facile</niveau>
  <cout>Bon marché</cout>
  <ingredients>
    <p>2 cuisses de poulet</p>
    <p>1 oignon</p>
    <p>10 cl d'huile d'olive (environ 1/2 verre à eau)</p>
    <p>3 cuillères à soupe de miel</p>
    <p>sel, poivre</p>
  </ingredients>
  <preparation>
  <![CDATA[
    Préchauffez le four à 200'C (thermostat 6–7).
    Peler les oignons et les émincer.
    Les disposer dans un plat à gratin (environ 40 x 20 cm).
    Disposer les cuisses de poulet sur les oignons.
    Dans un bol, mélangez l'huile et le miel.
    Badigeonner le poulet de ce mélange.
    Saler et poiver les cuisses.
    Mettre le plat dans le four pendant environ 50 minutes.
    Vos oignons sont caramélisés et de même pour vos cuisses de poulet.
    Bonne dégustation.
    Cette recette accompagnée par une bonne purée, c'est simple et c'est un vrai
        régal!Vous pouvez ajouter un deuxième oignon selon vos envies.
  ]]>
  </preparation>
</recette>
```

FIGURE 1 – Recette issue du corpus d'apprentissage avec la référence des différentes pistes en méta-données (sauf pour la tâche 4, dont les ingrédients normalisés pour cette recette étaient *cuisse-de-poulet, huile-d-olive, miel, oignon, poivre, sel*)

qu'à des écarts d'un seul niveau. Enfin, les annotateurs humains parviennent à identifier si la recette est plutôt d'un niveau facile ou d'un niveau difficile. La même expérience réalisée sur un second corpus de dix recettes auprès d'un groupe d'étudiants de l'EBSI permet l'obtention de résultats similaires avec un taux de réussite moyen de 34,0 %.

D'autre part, les annotateurs sont majoritairement d'accord entre eux. Si l'évaluation est réalisée non plus d'après le niveau de référence mais selon le niveau majoritairement attribué par les annotateurs, le taux moyen de réussite sur l'ensemble du corpus monte à 52,0 %. Cette observation témoigne du fait que le niveau de difficulté d'origine est mal choisi pour un grand nombre de recettes. Les concepteurs de sites participatifs qui évaluent la qualité d'un produit *(recettes, livres, films, etc.)* pourraient donc être intéressés par des outils d'évaluation automatique.

Type de plat. La tâche de prédiction du type de plat *(entrée, plat, dessert)* a été réalisée par les douze étudiants de l'EBSI. Sur cette tâche, ils ont fourni entre 6 et 9 bonnes réponses sur le corpus de dix recettes, avec un taux moyen de réussite de 77,5 %. À l'évidence, la tâche semble moins simple qu'il n'y paraît, a fortiori sur la distinction entre une entrée et un plat principal.

Recette	Référence	Annotateur									
		01	02	03	04	05	06	07	08	09	10
Couscous végétarien	1	4	3	**1**	3	3	3	2	3	4	3
Pain d'épice	1	2	2	**1**	**1**	**1**	**1**	2	3	2	**1**
Tartiflette courgettes	1	**1**	**1**	**1**	2	**1**	**1**	2	2	**1**	**1**
Garbure landaise	2	3	4	1	4	4	**2**	**2**	3	4	3
Rougets farcis	2	3	3	4	**2**	**2**	4	**2**	3	4	4
Lasagnes bolognaise	3	2	**3**	2	2	2	**3**	1	**3**	2	**3**
Sushi californien	3	**3**	4	**3**	**3**	**3**	4	**3**	2	4	**3**
Boulettes orientales	4	1	2	2	3	1	2	3	3	3	1
Lasagnes de légumes	4	3	2	**4**	3	3	**4**	2	**4**	3	3
Ravioles tartares	4	**4**	**4**	**4**	0	3	3	3	**4**	**4**	**4**

TABLE 2 – Prédictions des annotateurs humains en matière de niveau de difficulté d'une recette (0 : absence de réponse, 1 : très facile, 2 : facile, 3 : moyennement difficile, 4 : difficile)

Appariement titre/recette. La tâche consistant à relier chaque recette à son titre *(un corpus de dix recettes et de dix titres séparés)* a été réalisée par les étudiants de l'EBSI qui ont pratiquement tous réussi à effectuer les appariements corrects ; seul un(e) étudiant(e) a interverti deux titres. Le taux moyen de réussite est donc élevé sur cette tâche et se monte à 98,3 %. Le passage à l'échelle avec plusieurs milliers de titres et de recettes à apparier devrait se révéler plus complexe.

Ingrédient étranger. Sur la tâche d'identification de l'ingrédient étranger à une recette, [5] tous les étudiants de l'INaLCO ont correctement identifié l'ingrédient ajouté. Pour le défi, nous avons remplacé cette tâche par l'identification des ingrédients normalisés qui constituent la recette.

3 Évaluation et discussion

3.1 Modalités d'évaluation

Tâches 1 & 2 Les étiquettes de classe de la tâche 1 correspondent à des graduations sur une échelle de valeurs. Nous avons donc retenu comme mesure principale la distance relative moyenne à la solution, calculée en micro-moyenne comme mesure principale et en macro-moyenne comme mesure secondaire. L'échelle de valeurs comprend 4 valeurs réparties de manière symétrique sur une droite euclidienne (un espace à une dimension), de part et d'autre de l'origine. La distance entre les deux premières valeurs de classe de difficultés opposées est choisie comme double de celle séparant deux valeurs de difficulté de même signe (par ex. d(*très facile*, *facile*)= 1 mais d(*facile*, *moyennement difficile*)= 2), ceci afin de pénaliser plus un système qui ferait une erreur de genre de difficulté (facile/moyennement difficile) qu'un système faisant une erreur de qualification de difficulté (*très facile*/*facile*). Comme mesure secondaire, nous considérons le même calcul en macro-moyenne, afin de séparer des systèmes potentiellement ex-aequo en observant de manière plus fine leur performance sur les classes de recettes ayant des effectifs de

5. Pour chacune des cinq recettes constituant le corpus, nous avons ajouté un ingrédient provenant d'une recette du même type de plat. Sur la recette *« Crevettes au riz et à l'avocat en coque de tomate »* (entrée), nous avons ajouté des tomates cerises, tandis que sur la recette *« Cheesecake à la carotte »* (dessert) nous avons ajouté du sirop d'érable.

petite taille. À chacune des 4 valeurs possibles pour la donnée de référence r_i, correspond une valeur de distance maximale possible entre la réponse du système et cette donnée $dmax(h_i, r_i)$. Par exemple si la bonne réponse est *très facile*, la distance maximale possible est 4 si la réponse du système est *difficile*, mais si la bonne réponse est *facile*, la distance maximale possible sera 3, pour une réponse du système *difficile*. L'exactitude en distance relative à la solution moyenne (EDRM) se calcule en micro-moyenne comme indiqué dans l'équation 1.

$$\text{EDRM} = \frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{d(h_i, r_i)}{dmax(h_i, r_i)}\right) \tag{1}$$

Cependant, la première tâche du défi peut aussi être considérée comme une tâche de classification automatique, comme l'est la tâche 2. À ce titre, les mesures habituelles de rappel, précision et F-mesure sont donc adaptées. Nous avons donc vérifié la corrélation des résultats obtenus avec l'EDRM sur la tâche 1 avec les mesures de précision et rappel, qui sont les mesures principales de la tâche 2. Ces mesures sont calculées à la fois en termes de micro-mesure (formules 2 et 3) et de macro-mesures (formules 4 et 5), le classement officiel étant établi sur la base des micro-mesures de manière à accorder un poids équivalent à chaque élément mesuré, même si cette mesure revient à privilégier les classes composées d'un grand nombre d'individus *(très facile, facile)* au détriment des classes faiblement représentées *(moyennement difficile, difficile)*.

$$\text{Micro-rappel} = \frac{\sum_{i=1}^{n} \text{vrais positifs}(i)}{\sum_{i=1}^{n} \text{vrais positifs}(i) + \sum_{i=1}^{n} \text{faux négatifs}(i)} \tag{2}$$

$$\text{Micro-précision} = \frac{\sum_{i=1}^{n} \text{vrais positifs}(i)}{\sum_{i=1}^{n} \text{vrais positifs}(i) + \sum_{i=1}^{n} \text{faux positifs}(i)} \tag{3}$$

$$\text{Macro-rappel} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\text{vrais positifs}(i)}{\text{vrais positifs}(i) + \text{faux négatifs}(i)}\right) \tag{4}$$

$$\text{Macro-précision} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\text{vrais positifs}(i)}{\text{vrais positifs}(i) + \text{faux positifs}(i)}\right) \tag{5}$$

Tâches 3 et 4. Les tâches 3 et 4 demandaient de renvoyer une liste ordonnée d'hypothèses, l'objectif étant de placer le plus haut possible dans cette liste la ou les bonnes hypothèses, et peuvent de ce fait être mises en parallèle avec le format et le mode d'évaluation des tâches de recherche d'information. Dans la tâche 3, une seule bonne réponse (le titre original de la recette) était possible pour chaque recette R_i. Une mesure basée sur le rang r_i de cette bonne réponse a donc été employée : l'inverse de ce rang, et donc pour l'ensemble des N recettes la moyenne de l'inverse du rang (MRR, voir formule 6). Dans la tâche 4, plusieurs bonnes réponses (les n_i ingrédients $\{I_i^1 \dots I_i^j \dots I_i^{n_i}\}$ de la recette R_i) étaient attendues pour chaque recette. La moyenne des précisions non interpolées $P(I_i^j)$ calculées à chaque position, dans la liste d'hypothèses, d'une des n_i réponses correctes I_i^j pour la recette R_i, est alors une mesure pertinente, et pour l'ensemble des recettes la moyenne de cette précision moyenne (MAP, voir formule 6).

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{r_i} \qquad MAP = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i}P(I_i^j) \tag{6}$$

On rappelle que si une bonne réponse n'est pas présente dans la liste fournie par un système, sa précision moyenne vaut zéro, comme si elle avait été classée à l'infini par le système. MRR et MAP ont été calculés à l'aide du programme `trec_eval` [6].

Pour la tâche 3, rappel, précision et F-mesure ont également été calculés en prenant comme classes les titres des recettes et en examinant le titre proposé au rang 1.

3.2 Résultats des participants

Tâche 1 — Niveau de difficulté. Nous indiquons dans le tableau 3 les résultats globaux obtenus par les participants sur la première tâche. Le tableau 4 donne les résultats par classe. Les meilleurs résultats et le vainqueur sont indiqués en gras. *rf* et *nb* sont présentés en section 3.3.

Équipe, soumission	Macro				**Micro**		Rang
	R	P	F	EDRM	P=R=F	**EDRM**	
02 — LIM&Bio, #1	0,399	0,332	0,363	0,609	0,580	0,826	
02 — LIM&Bio, #2	0,415	0,339	0,373	0,624	0,586	0,828	4
02 — LIM&Bio, #3	0,406	0,329	0,364	0,621	0,574	0,824	
04 — GREYC, #1	0,273	0,250	0,261	0,501	0,489	0,794	6
04 — GREYC, #2	0,265	0,248	0,256	0,501	0,465	0,787	
04 — GREYC, #3	0,289	0,281	0,285	0,543	0,360	0,624	
22 — LIA, #1	0,353	0,633	0,453	0,600	0,592	0,828	3
22 — LIA, #2	0,363	0,603	0,453	0,643	0,581	0,813	
22 — LIA, #3	0,364	0,613	0,457	0,638	0,588	0,820	
25 — Celi France, #1	0,304	0,342	0,322	0,558	0,511	0,799	5
26 — Orange Labs, #1	0,252	0,275	0,263	0,503	0,086	0,234	
26 — Orange Labs, #2	0,395	0,524	0,451	0,659	0,612	0,840	2
28 — Wikimeta Lab, #1	**0,419**	0,460	0,438	**0,690**	0,609	0,835	
28 — Wikimeta Lab, #2	0,375	**0,682**	**0,484**	0,612	**0,625**	**0,843**	1
comparaison : rf	0,301	0,337	0,318	0,511	0,543	0,810	
comparaison : nb	0,489	0,373	0,423	0,723	0,525	0,754	

TABLE 3 – Résultats globaux sur la tâche 1, *Niveau de difficulté*, évalués en termes de rappel, précision, F-mesure et EDRM en macro-moyenne, P=R=F et EDRM en micro-moyenne

Tâche 2 — Type de plat. Le tableau 5 donne les résultats globaux tandis que le tableau 6 fournit les résultats par classe. Les meilleurs résultats et le vainqueur sont indiqués en gras.

Tâche 3 — Appariement titre/recette. Le tableau 7 donne les résultats globaux. Les meilleurs résultats et le vainqueur sont indiqués en gras. Les équipes 02 et 26 ont renvoyé plusieurs fois le même titre au rang 1 pour des recettes différentes : ces titres (= classes) sont donc proposés plusieurs fois, dont toutes sauf au mieux une sont incorrectes, ce qui fait baisser leur précision.

6. `http://trec.nist.gov/trec_eval/` : pour la moyenne de l'inverse du rang,
`trec_eval -c -q -mrecip_rank reference.qrels hypotheses.txt`,
et pour la précision moyenne non interpolée,
`trec_eval -c -q -mmap reference.qrels hypotheses.txt`.

Équipe	Très facile			Facile			Moyennement diff.			Difficile		
	R	P	F	R	P	F	R	P	F	R	P	F
02 #1	0,769	0,625	0,689	0,478	0,528	0,502	0,000	0,000	0,000	0,350	0,175	0,233
02 #2	0,769	0,630	0,692	0,491	0,536	0,512	0,000	0,000	0,000	0,400	0,190	**0,258**
02 #3	0,797	0,606	0,689	0,429	0,535	0,476	0,000	0,000	0,000	0,400	0,174	0,242
04 #1	0,444	0,555	0,493	0,646	0,446	0,527	0,000	0,000	0,000	0,000	0,000	0,000
04 #2	0,292	0,557	0,384	0,768	0,433	0,554	0,000	0,000	0,000	0,000	0,000	0,000
04 #3	0,232	0,575	0,331	0,506	0,440	0,471	0,418	0,107	0,170	0,000	0,000	0,000
22 #1	0,762	0,624	0,686	0,499	0,555	0,525	0,101	0,352	0,156	0,050	1,000	0,095
22 #2	0,711	0,645	0,676	0,521	0,554	0,537	0,169	0,215	0,189	0,050	1,000	0,095
22 #3	0,719	0,644	0,679	0,529	0,553	0,541	0,159	0,254	0,195	0,050	1,000	0,095
25 #1	0,498	0,586	0,539	0,617	0,465	0,530	0,101	0,317	0,153	0,000	0,000	0,000
26 #1	0,002	0,667	0,004	0,007	0,350	0,014	1,000	0,083	0,153	0,000	0,000	0,000
26 #2	0,715	0,671	0,692	0,587	0,554	**0,570**	0,180	0,472	**0,261**	0,100	0,400	0,160
28 #1	0,759	0,662	0,707	0,525	0,570	0,547	0,190	0,343	0,245	0,200	0,267	0,229
28 #2	0,786	0,660	**0,717**	0,546	0,579	0,562	0,116	0,489	0,188	0,050	1,000	0,095
rf	0,648	0,591	0,618	0,533	0,491	0,511	0,021	0,267	0,039	0,000	0,000	0,000
nb	0,640	0,651	0,646	0,427	0,560	0,484	0,339	0,202	0,253	0,550	0,077	0,136

TABLE 4 – Résultats détaillés par classe sur la tâche 1, *Niveau de difficulté*

Équipe, soumission	Macro			Micro	Rang
	R	P	F		
02 — LIM&Bio #1	0,806	0,832	0,819	0,834	
02 — LIM&Bio #2	0,841	0,842	0,841	0,849	3
02 — LIM&Bio #3	0,806	0,832	0,819	0,834	
04 — GREYC #1	0,760	0,741	0,750	0,746	5
04 — GREYC #2	0,584	0,609	0,596	0,573	
04 — GREYC #3	0,381	0,418	0,398	0,331	
22 — LIA #1	0,873	0,868	0,871	0,876	
22 — LIA #2	0,879	0,880	0,879	0,886	
22 — LIA #3	**0,881**	**0,884**	**0,882**	**0,889**	1
26 — Orange Labs #1	0,767	0,857	0,810	0,821	
26 — Orange Labs #2	0,784	0,840	0,811	0,827	4
26 — Orange Labs #3	0,779	0,789	0,784	0,784	
28 — Wikimeta Lab, #1	0,843	0,850	0,847	0,856	2
comparaison : nb	0,862	0,852	0,857	0,859	

TABLE 5 – Résultats globaux sur la tâche 2, *Type de plat*, évalués en termes de rappel, précision et F-mesure en macro-moyenne, micro-moyenne

Pour ce qui est du rappel, il est à 1 pour un titre (= une classe) si ce titre est trouvé au moins une fois correctement au rang 1 : donc c'est la proportion des titres qui sont trouvés correctement pour au moins (= exactement) une recette. Cela explique pourquoi il est égal à la précision au rang 1 (P@1), proportion des recettes dont le titre correct a été trouvé au rang 1.

Équipe	Entrée			Plat principal			Dessert		
	R	P	F	R	P	F	R	P	F
02 #1	0,544	0,730	0,624	0,894	0,794	0,841	0,980	0,970	0,975
02 #2	0,687	0,702	0,694	0,851	0,844	0,848	0,983	0,980	*0,982*
02 #3	0,544	0,729	0,623	0,898	0,796	0,844	0,976	0,971	0,974
04 #1	0,730	0,568	0,639	0,677	0,828	0,745	0,873	0,825	0,849
04 #2	0,662	0,378	0,481	0,540	0,699	0,609	0,551	0,749	0,635
04 #3	0,792	0,265	0,397	0,208	0,528	0,298	0,142	0,461	0,217
22 #1	0,779	0,744	0,761	0,868	0,890	0,879	0,971	0,971	0,971
22 #2	0,756	0,777	0,766	0,890	0,882	0,886	0,989	0,982	**0,986**
22 #3	0,762	0,784	**0,773**	0,893	0,883	**0,888**	0,989	0,983	**0,986**
26 #1	0,345	0,882	0,496	0,965	0,755	0,847	0,991	0,934	0,962
26 #2	0,427	0,784	0,553	0,937	0,764	0,842	0,986	0,972	*0,979*
26 #3	0,514	0,623	0,563	0,836	0,774	0,804	0,988	0,970	*0,979*
28 #1	0,669	0,742	0,703	0,872	0,840	0,856	0,989	0,969	*0,979*
nb	0,776	0,696	0,734	0,825	0,883	0,853	0,986	0,976	0,981

TABLE 6 – Résultats détaillés par classe sur la tâche 2, *Type de plat*

Équipe, soumission	Macro			P@1	MRR	Rang
	R	P	F			
02 — LIM&Bio #1	0,036	0,019	0,025	0,036	0,036	
02 — LIM&Bio #2	0,241	0,200	0,218	0,241	0,241	2
02 — LIM&Bio #3	0,133	0,119	0,125	0,133	0,133	
04 — GREYC #1	0,127	0,127	0,127	0,127	0,127	3
26 — Orange Labs #1	0,309	0,249	0,276	0,308	0,430	
26 — Orange Labs #2	0,306	0,251	0,276	0,306	0,423	
26 — Orange Labs #3	**0,314**	**0,259**	**0,284**	**0,314**	**0,434**	1

TABLE 7 – Résultats globaux sur la tâche 3, *Appariement titre-recette*, évalués en termes de rappel, précision et F-mesure en macro-moyenne, précision au rang 1, et moyenne de l'inverse du rang (MRR)

Tâche 4 — Ingrédients d'une recette. Les résultats de la dernière tâche sont donnés dans le tableau 8. Notons que les systèmes 02 et 25 n'avaient pas normalisé les noms des ingrédients qu'ils produisaient : nous avons inclus cette normalisation (désaccentuation, remplacement des espaces par des tirets), ce qui n'a pas changé le classement. La meilleure MAP de l'équipe 25 serait sinon de 0,6559 (−0,01), et celle de l'équipe 02 serait de 0,4083 (−0,05). Les équipes 25 et 28 ont également laissé passer des ingrédients hors liste normalisée (respectivement 94 et 7 dans leur meilleur système), seules les équipes 04 et 22 se sont assurées que leur système produisait uniquement des ingrédients normalisés.

Équipe	02 LIM&Bio	04 GREYC	22 LIA	25 Celi France	28 Wikimeta Lab
#1	0,4115	0,4881	*0,6287*	**0,6662**	0,5675
#2	0,4170	0,5074	0,6218	—	*0,6428*
#3	0,4649	0,5556	0,6191	—	—
Rang	5	4	3	1	2

TABLE 8 – Résultats globaux sur la tâche 4, *Ingrédients d'une recette*, évalués en termes de moyenne de la précision non interpolée (MAP)

3.3 Discussion

Tâche 1 — Niveau de difficulté. Les classes de cette tâche étaient très déséquilibrées : la troisième classe (Moyennement difficile) avait un nombre d'instances un ordre de grandeur au-dessous des deux premières, et la quatrième (Difficile) était encore un ordre de grandeur plus bas. De ce fait, il était beaucoup plus difficile de détecter correctement les deux dernières classes, ce que reflètent les résultats des systèmes par classe : F-mesure généralement vers 0,6–0,7 pour Très facile, 0,5 pour Facile, 0,1–0,2 pour Moyennement difficile, et 0–0,2 pour Difficile. Une bonne performance sur les deux premières classes assurait un résultat honorable en micro-mesure, et rares sont les systèmes qui ont pu détecter un nombre non négligeable d'instances des deux classes difficiles (26-#2 et #28-1).

À titre de base de comparaison, nous avons mis au point une méthode simple fondée sur une représentation des recettes par les attributs suivants. Les trois champs textuels (titre, ingrédients, préparation) ont été segmentés en mots (en conservant les nombres et ponctuations), les mots des titres ont été représentés par des attributs binaires (présence / absence du mot) et les mots des deux autres champs par des attributs numériques (tf.idf, normalisé par la longueur du champ). Le nombre d'ingrédients (nombre de <p> dans la liste des ingrédients) a également été ajouté, ainsi que la longueur en mots et caractères de la liste d'ingrédients et de la préparation. Un classifieur bayésien naïf (*nb*, NaiveBayesMultinomial dans Weka) et un classifieur à forêt d'arbres (*rf*, RandomForest dans Weka) ont été entraînés sur ces vecteurs d'attributs : ces deux classifieurs ont un temps d'entraînement court à relativement court et sont assez robustes. La forêt d'arbres a obtenu une meilleure proportion d'instances bien classées (0,544) en validation croisée sur dix parties sur le corpus d'entraînement, mais sans savoir prédire la classe Difficile ; le classifieur bayésien, avec une correction plus faible (0,527), y trouvait en revanche près de la moitié des recettes difficiles. Les résultats sur le jeu de test (bas des tableaux 3 et 4, *rf* et *nb*) reflètent ces différences, qui induisent des différences importantes dans la macro F-mesure.

Tâche 2 — Type de plat. Quatre équipes sur cinq (Bost *et al.*, 2013; Hamon *et al.*, 2013; Collin *et al.*, 2013; Charton *et al.*, 2013) ont obtenu des résultats proches de la perfection pour la catégorisation du dessert (F=0,986, 0,982, 0,979, 0,979). Cela semble indiquer que cette catégorie était plus facile à trouver, ou mieux définie : les desserts contiennent en général du sucre, etc. Pour vérifier cela, nous avons entraîné pour cette tâche le classifieur bayésien naïf avec les mêmes attributs que ci-dessus (la forêt d'arbres donnait des résultats beaucoup moins bons sur le jeu d'entraînement). Ce classifieur obtient les résultats résumés au bas des tableaux 5 et 6. Dans la catégorie Dessert, il se placerait lui aussi dans le groupe des très bons résultats. Dans les deux autres catégories, il est plusieurs points derrière le meilleur système, ce qui montre l'intérêt

des traitements supplémentaires effectués par le LIA au-delà d'une simple segmentation en mots.

La distinction entre entrée et plat principal était plus difficile à établir : de fait, certains plats peuvent se servir indifféremment en entrée ou en plat principal, et les indications de quantité ne suffisent pas non plus à distinguer leur destination. Ceci étant posé, la bonne performance de cette méthode de base montre que cette tâche était nettement plus facile que la tâche 1, parce qu'elle avait une classe de moins, que ses classes étaient plus équilibrées, et que leur définition était moins subjective.

Tâche 3 — Appariement titre/recette. Les scores (MRR) pour la tâche 3 vont de 0,13 à 0,43. L'équipe 02 (Hamon *et al.*, 2013) a fourni zéro ou un titre par recette, l'équipe 04 (Lejeune *et al.*, 2013) exactement un titre par recette, et l'équipe 26 (Collin *et al.*, 2013) a produit 50 titres par recette. Cette dernière se détache des deux autres participants. Un score de 0,43 correspond à un placement du bon titre en moyenne entre les positions 2 et 3, ou encore, pour un système qui ne renverrait qu'un titre par recette, à trouver le bon titre une fois sur 2 à 3. En pratique, les systèmes des participants ont placé le bon titre en première position (P@1, précision au rang 1) dans respectivement 31,4 % des cas (éq. 26), 24,1 % des cas (02), et 12,7 % des cas (04). On vérifie ainsi que seul un système qui classe plusieurs titres pour chaque recette, comme le 26, peut obtenir un MRR supérieur à sa précision au rang 1.

Tâche 4 — Ingrédients d'une recette. Les scores (MAP) pour la tâche 4 s'étagent de 0,36 à 0,66. Le groupe des trois équipes de tête a bien réussi la tâche, avec une meilleure MAP entre 0,63 et 0,66. Ce score correspond par exemple au fait de classer en premier, dans chaque recette, les deux tiers des ingrédients, ce qui est très bon étant donné les défauts de la référence signalés plus haut.

Le tableau 9 montre le nombre de recettes pour lesquelles tous les ingrédients ont été trouvés et classés en premier (précision moyenne = 1) : les équipes 28 (Charton *et al.*, 2013) et 22 (Bost *et al.*, 2013) font beaucoup plus souvent le grand chelem que l'équipe 25 (Dini *et al.*, 2013), qui obtient pourtant la meilleure MAP (et met donc sans doute plus régulièrement les bons ingrédients vers le haut, même si ce n'est pas tout en haut). On reconnaît donc ici une stratégie

Équipe	02 LIM&Bio	04 GREYC	22 LIA	25 Celi France	28 Wikimeta Lab
#1	9	19	**173**	84	53
#2	11	68	147	—	**176**
#3	24	44	141	—	—

TABLE 9 – Tâche 4 : nombre de résultats parfaits (tous les ingrédients corrects en tête de liste)

différente entre l'équipe 25 et les équipes 22 et 28. L'équipe 25 est la seule à avoir appliqué le précepte de la MAP : classer tous les ingrédients est toujours mieux que de n'en classer qu'une partie. Elle a ainsi systématiquement classé une longue liste de quelque 850 ingrédients pour chaque recette. Si par exemple elle n'avait classé que 20 ingrédients par recette, sa MAP aurait été de 0,6513 (−0,015) — on constate que cela n'aurait néanmoins pas modifié le classement.

4 Conclusion

L'édition DEFT2013 a porté sur l'analyse des recettes de cuisine au travers de quatre tâches. La référence a été constituée en reprenant directement les informations présentes sur Marmiton, le site utilisé pour constituer les corpus. Du fait de cette référence « naturelle », les tâches présentaient des difficultés importantes : définitions subjectives et classes très déséquilibrées pour la difficulté de la recette, similarité entre entrée et plat principal dans les types de plats, créativité des titres de recettes, référence imparfaite et difficultés de normalisation dans la détection des ingrédients.

Au regard de ces difficultés, il apparaît que les participants ont réussi fort honorablement à prédire les valeurs renseignées par les humains qui déposent une recette sur le site. On peut même supposer que les méthodes développées pour estimer la difficulté d'une recette ou détecter ses ingrédients pourraient être utiles pour indexer automatiquement les recettes du site et lui apporter une plus grande cohérence.

Remerciements. Nous remercions Flora Badin (INIST–CNRS) et Dominic Forest (EBSI–Université de Montréal) pour les réflexions qu'ils ont menées sur les tâches à proposer aux participants du défi cette année. Nous remercions également les étudiants français (INaLCO) et canadiens (EBSI) pour les tests humains qu'ils ont réalisés et l'aide qu'ils nous ont ainsi apportée. Merci à tous les participants pour les efforts déployés et la richesse des méthodes utilisées. Nous les félicitons également pour la manière dont ils ont filé la métaphore culinaire dans leurs articles ! Enfin, nous remercions les organisateurs de TALN/Recital 2013 pour leur aide logistique dans la préparation de l'atelier de clôture de DEFT.

Références

Bost, X., Brunetti, I., Cabrera-Diego, L. A., Cossu, J.-V., Linhares, A., Morchid, M., Torres-Moreno, J.-M., El Bèze, M. et Dufour, R. (2013). Système du LIA à DEFT'13. *In Actes de DEFT*.

Charton, E., Jean-Louis, L., Meurs, M.-J. et Gagnon, M. (2013). Trois recettes d'apprentissage automatique pour un système d'extraction d'information et de classification de recettes de cuisine. *In Actes de DEFT*.

Collin, O., Guerraz, A., Hiou, Y. et Voisine, N. (2013). Participation de Orange Labs à DEFT 2013. *In Actes de DEFT*.

Dini, L., Bittar, A. et Ruhlmann, M. (2013). Approches hybrides pour l'analyse de recettes de cuisine. *In Actes de DEFT*.

Hamon, T., Périnet, A. et Grabar, N. (2013). Efficacité combinée du flou et de l'exact des recettes de cuisine. *In Actes de DEFT*.

Lejeune, G., Lecluze, C. et Brixtel, R. (2013). DEFT2013, une cuisine de caractères. *In Actes de DEFT*.

Méthodes des participants

Efficacité combinée du flou et de l'exact des recettes de cuisine

Thierry Hamon[1] Amandine Périnet[1,2] Natalia Grabar[3]
(1) LIM&BIO (EA3969), Université Paris 13, Sorbonne Paris Cité, 74, rue Marcel Cachin, 93017 Bobigny
thierry.hamon@univ-paris13.fr
(2) Lingua et Machina c/o INRIA Rocquencourt BP 105 78153 Le Chesnay Cedex
amandine.perinet@lingua-et-machina.com
(3) CNRS UMR 8163 STL, Université Lille 1&3, 59653 Villeneuve d'Ascq
natalia.grabar@univ-lille3.fr

RÉSUMÉ

La cuisine et les recettes de cuisine occupent une place importante dans notre vie. L'alimentation a par ailleurs une influence directe sur notre bien-être, santé et bonne humeur. La compétition DEFT 2013 étant dédiée à l'analyse des recettes de cuisine, nous y avons participé pour étudier ce domaine de spécialité. Quatre tâches orientées catégorisation et extraction d'informations sont proposées. En fonction des tâches, nous utilisons des approches à base de règles et d'apprentissage. Nous avons aussi construit des ressources sémantiques pour traiter des informations exactes (par exemple, quantités exprimées en grammes ou litres, durées exprimées en minutes ou jours) et floues (par exemple, quantités exprimées en *chouilla* et *louche*, durées séquencées avec *après, ensuite, alors que*). En exploitant notre approches et nos ressources, et en naviguant entre l'exact et le flou, nous avons obtenu les résultats officiels suivants : 4e/6 sur la tâche 1, 3e/5 sur la tâche 2, 2e/3 sur la tâche 3, et 5e/5 sur la tâche 4. Plusieurs améliorations sont envisagées.

ABSTRACT

Combined efficiency of fuzziness and exactness in recipes.

An important place in our lifes is dedicated to *cuisine* and recipes. Besides, food has a direct impact on our well-being, health and mood. Because the DEFT 2013 challenge was dedicated to the analysis of recipes, we participated in this challenge to study this specialized area. Four tasks oriented on categorization and information extraction have been proposed. According to tasks, we use rule-based and machine learning approaches. We have also built semantic resources to process exact (*i.e.*, quantities expressed in grams or liters, durations expressed in minutes or days) and fuzzy (*i.e.*, quantities expressed in *chouilla* and *louche*, durations sequenced with *après, ensuite, alors que*) information. Using the approaches and resources, and sailing between the exactness and fuziness, we have obtained the following official results : 4e/6 on task 1, 3e/5 on task 2, 2e/3 on task 3, and 5e/5 on task 4. Several improvements are planned.

MOTS-CLÉS : Traitement Automatique des Langues, campagne d'évaluation, recettes de cuisine, annotation sémantique, création de ressources sémantiques, systèmes à base de règles, système d'apprentissage automatique.

KEYWORDS: Natural Language Processing, evaluation campaign, recipes, semantic annotation, semantic resource creation, rules-based system, machine learning system.

1 Introduction

La cuisine et les recettes de cuisine occupent une place importante dans notre vie. Elles font certainement aussi partie des facteurs principaux qui influencent notre bien-être et la bonne humeur au quotidien. En effet, selon un dicton russe, *Любовь и голод правят миром* (*L'amour et la faim gouvernent le monde*). D'autres dictons marquent aussi l'importance de la cuisine et de la nourriture dans notre vie (de Rudder, 2006; Lim, 2007) : *les carottes sont cuites*, *la fin des haricots*, *s'occuper de ses oignons* ou *avoir mangé son pain blanc*. Même nos hommes les plus illustres s'intéressent à la cuisine (Dumas, 1873).

Il n'est donc pas étonnant que les scientifiques aussi s'intéressent à l'alimentation et la nutrition. Des phénomènes connus au grand public comme ceux de *malbouffe* ou de *French paradox* sont étudiés. Si ces études expliquent et justifient les bienfaits du vin rouge et de ses substances, elles montrent aussi que la malbouffe (par exemple, la consommation de hamburgers, de hot-dogs, de frites, de chips ou de sodas) peut favoriser l'obésité, le diabète, les maladies cardiovasculaires, des dépressions et même certains cancers. Il ne faut donc pas sous-estimer la place que l'alimentation, et la bonne alimentation surtout, occupent dans notre vie. Pour mieux étudier les questions relatives à l'alimentation de la population, il existe même une sous-section 44.04 de CNU dédiée à la nutrition. L'effort principal des chercheurs consiste à attirer notre attention sur la qualité de notre alimentation et son influence sur notre santé. Depuis quelques années, l'étude NutriNet-Santé [1] semble avoir concentré les efforts et les initiatives autour de ces questions en France. Dans de telles études, une attention particulière peut être portée à certaines pathologies, comme le diabète, à la condition sociale ou à la provenance géographique de la population étudiée (Jones-Smith *et al.*, 2013; Andreeva *et al.*, 2013).

Dans les domaines de l'informatique et du Traitement Automatique de Langues (TAL), les textes de recettes de cuisine ont aussi attiré l'attention des chercheurs. Le premier travail est certainement celui décrivant le système Epicure et son application aux recettes de cuisine (Dale, 1989). Ce système a pour objectif de générer une description linguistique des recettes de cuisine. Des représentations syntaxiques profonde et de surface sont proposées en utilisant la grammaire d'unification. Le fonctionnement du système s'appuie sur des connaissances du domaine, comme par exemple les objets (individuels, massifs, quantifiés ou non) encodés dans une ontologie. Les objets évoluent en fonction des états qu'ils subissent. Ce système de TAL prend aussi en compte la pronominalisation et les anaphores. L'objectif de ce système semble d'avoir implémenté une approche sémantique fine. Le passage à l'échelle et le traitement de grosses données n'étaient pas au centre d'intérêt à l'époque.

Un travail plus récent est fait dans le cadre du Computer Cooking Contest [2], qui a lieu tous les ans depuis 2008. Cette compétition propose de traiter les recettes de cuisine. En 2012, les objectifs étaient clairement orientés sur le Raisonnement à partir de cas. Par exemple, il s'agit de construire automatiquement un système à base de cas (1) d'une part pour en montrer la faisabilité à partir du texte libre et (2) d'autre part pour proposer de nouvelles recettes en se basant sur les recettes existantes et en fonction des spécifications données (Dufour-Lussier *et al.*, 2013). Dans ce travail cité, l'attention particulière est réservée aux actions (souvent des verbes) et à leurs arguments. En plus, plusieurs traitements de TAL sont appliqués : étiquetage morpho-syntaxique, résolution d'anaphores, analyse syntaxique. 15 recettes de cuisine sont traitées. Cette compétition

1. *https ://www.etude-nutrinet-sante.fr*
2. *http ://computercookingcontest.net*

a donné aussi lieu aux travaux interdisciplinaires sur la reconnaissance d'aliments dans notre frigo (Kamoda *et al.*, 2012), ou encore sur l'apprentissage et modélisation des gestes culinaires grâce à des gants spéciaux (Ota *et al.*, 2012).

La compétition de TAL DEFT 2013 propose aussi de travailler avec les recettes de cuisine. Devant cette perspective de découverte et de mise au point scientifique, culturelle et culinaire, nous avons décidé de participer au challenge. Les quatre tâches sympathiques de la compétition sont orientées sur la catégorisation des recettes (tâches 1 et 2) et l'extraction d'information (tâches 3 et 4). Ces tâches sont les suivantes :

1. 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.
2. Identifier à partir du titre et du texte de la recette le type de plat préparé : entrée, plat principal, dessert.
3. Apparier le texte d'une recette à son titre.
4. Extraire du titre et du texte d'une recette la liste de ses ingrédients.

Il va de soi que, pour aborder efficacement et positivement le traitement des recettes de cuisine, il faut être muni de bons ingrédients linguistiques (section 2) et de bons ustensiles de TAL (section 3). Nous présentons aussi les recettes proposées (section 4) et les résultats que nous avons pu obtenir en sortie (section 5). Comme le perfectionnement (au moins scientifique, culturel et culinaire) n'a pas de limites, nous mentionnons quelques perspectives que nous aimerions mettre en oeuvre dans l'avenir proche (section 6).

2 Ingrédients

Parmi les ingrédients utilisés, nous avons plusieurs ressources (résumées dans la table 1) :

– Une liste d'ingrédients de cuisine et d'aliments collectés dans des sources disponibles en ligne [3] [4] [5] [6], dans la partie française de l'UMLS (NLM, 2011) et à partir des ingrédients connus dans l'ensemble d'entraînement de la campagne. Une des difficultés a été de distinguer entre les ingrédients et les aliments. La solution que nous avons adoptée est de considérer que les ingrédients sont "bruts" alors que les aliments sont déjà préparés voire cuisinés. Logiquement, ce sont les ingrédients qui sont utilisés dans les recettes... Ce qui n'est bien sûr pas totalement vrai car de bons aliments comme les pâtes, les raviolis, les glaces, les sauces diverses et variées, sans parler des fromages, font aussi de très bons ingrédients irremplaçables dans notre cuisine. Devant ce flou inattendu, nous avons finalement décidé (1) d'introduire l'ambiguïté et de catégoriser les aliments concernés comme ingrédients aussi, ou bien (2) de simplifier la réalité et de les considérer comme de vrais ingrédients. Cette liste comporte 6 070 entrées catégorisées en viandes, légumes, fruits, produits de boulangerie, poissons, sucreries, etc.
– Une liste d'ustensiles collectés à partir de ressources disponibles en ligne [7] [8]. Cette liste comporte 222 entrées.

3. *http ://www.bioweight.com/glucides.html*
4. *http ://www.bioweight.com/proteines.html*
5. *http ://www.centre-clauderer.com/acides-bases/femme-2.htm*
6. *http ://les.calories.free.fr/*
7. *http ://popoblog.unblog.fr/liste-ustensiles-de-cuisine-mise-a-jour-le-130808/*
8. *fr.wikipedia.org/wiki/Ustensile_de_cuisine*

– Une liste de mots vides du français composée de 616 entrées et une liste d'exceptions composée de 37 entrées.
– Une ressource pour la détection de quantités des ingrédients, comme par exemple : *250 g de sucre, 3 oeufs, une bonne cuillère d'huile, 100 + 50 gr de beurre, 1/2 l de lait, deux graines de cardamone, un chouilla de sel, 2 louches de bouillon, beaucoup de menthe*. Nous avons distingué les quantités standards, déjà exprimées en grammes ou litres, et les quantités non standards (Delamasure, 2007), non exprimées en grammes ou en litres, mais qui font le charme de nos recettes de cuisine et en garantissent la réussite. Afin d'en assurer un traitement égalitaire et une utilisation possible dans un système de TAL, nous avons établi des heuristiques et avons converti toutes ces quantités en litres ou en kilogrammes. Nous nous sommes aidés pour ceci d'autres ressources et convertisseurs disponibles en ligne [9] [10]. Lorsque plusieurs équivalences étaient connues ou lorsque la question de conversion ne s'était pas encore posée, nous avons avancé nos propres propositions. Par exemple, les expressions comme *pincée, soupçon, chouilla, lichette, rasade, giclée, goutte, microgramme, 1 peu* signifient que la recette comporte 1 gramme de produit ; les expressions comme *louche, tube, fond, ravier, assiette creuse, poignée* signifient que la recette comporte 100 grammes de produit, etc. Lors de la normalisation, plusieurs opérations arithmétiques sont appliquées (somme, division, multiplication).
– Une ressource pour détecter des durées. Là aussi, nous distinguons la durée quantifiée et exprimée en minutes, heures, jours, nuits, etc., et la durée non quantifiée et séquencée par des expressions comme *puis, pendant que, alors que, ensuite*.
– Une liste d'actions (verbes et noms) de la ressource Verbaction (Hathout *et al.*, 2001).
– Une ressource distributionnelle (Harris, 1954) construite à partir des titres et du contenu des recettes (ensembles d'entraînement et de test). L'approche s'appuie sur deux méthodes d'acquisition de relations d'hyperonymie (Morin et Jacquemin, 2004; Bodenreider *et al.*, 2001) et une méthode l'acquisition de variantes terminologiques (Jacquemin, 1996). Pour délimiter les contextes de mots, nous utilisons des fenêtres graphiques de 10 mots à droite et 10 mots à gauche, en limitant les mots en relation aux mots appartenant à la même catégorie syntaxique (noms, adjectifs et termes identifiés par l'extracteur YaTeA (Aubin et Hamon, 2006)). En ce qui concerne la mesure de similarité, nous utilisons l'indice de Jaccard (Grefenstette et Tapanainen, 1994), qui normalise le nombre de contextes partagés par deux mots par le nombre total de contextes de ces mots. Nous avons également filtré les relations acquises avec la moyenne de trois seuils (nombre de contextes partagés, fréquence des contextes partagés, fréquence des mots pivots).
– Une ressource distributionnelle *FreDist* (Anguiano et Denis, 2011).
– Une ressource de synonymes de la langue générale fournis par le Petit Robert (Robert, 1990).

L'ingrédient principal reste bien sûr l'ensemble d'entraînement fourni par les organisateurs. Cet ensemble comporte 13 864 recettes. Chaque recette contient les informations suivantes : le titre, les ingrédients, les étapes ou le déroulement de la préparation, le coût, le niveau de difficulté et le type de plat. Mais seule le coût est disponible dans l'ensemble de test. Souvent, la boisson conseillée est aussi indiquée. L'ensemble d'entraînement, aidé par les ressources, a permis de créer le système de TAL et d'en faire les tests. Un autre ingrédient important sont les recettes du corpus de test, au nombre de 9 200 (2 300 par tâche). Ce corpus a été fourni pour une durée de trois jours : comme le feu de nos cuisinières ne chauffait pas assez fort et vite et pour ne pas risquer l'indigestion, le système de TAL était le seul moyen de test possible pour traiter les quatre tâches.

9. *http ://www.supertoinette.com/mesures-equivalences-culinaires.html*
10. *http ://webcafe.highbb.com/t1827-mesures-metriques-et-nord-americaine#13245*

Ressources	Exactes	Floues
Ressources lexicales	+ (ingrédients, ustensiles, synonymes)	+ (distributionnelles, aliments)
Quantités	+ exprimées en grammes et litres	+ exprimées autrement
Durées	+ quantifiées	+ non quantifiées
Actions	+	
Mots vides	+	
Recettes de cuisine	+ ensemble d'entraînement	+ ensemble de test

TABLE 1 – Ressources construites et utilisées afin de pouvoir traiter les informations exactes et floues contenues dans les recettes de cuisine.

3 Ustensiles de TAL

Nos ustensiles de TAL sont composés d'un système à base de règles et d'un système d'apprentissage automatique.

3.1 Système à base de règles

L'objectif du système à base de règles est d'assurer la reconnaissance de mots clés (ingrédients, ustensiles de cuisine...) et d'informations associées (quantités d'ingrédients, durées de préparation...). Ce système a quatre étapes principales. Trois étapes (reconnaissance de termes et d'informations associées, pondération de termes, et leur filtrage et sélection) sont adaptées du système utilisé lors de la compétition DEFT 2012 (Hamon, 2012). La quatrième étape, dédiée à l'appariement de titres, est adaptée du travail précédent (Hamon et Gagnayre, 2012).

3.1.1 Reconnaissance de termes (TermTagger) et d'informations associées

Les ressources décrites plus haut (listes d'ingrédients, d'ustensiles et d'actions) sont projetées sur les textes traités. Cette projection prend en compte les lemmes de mots du corpus. Nous utilisons le module Perl `Alvis::TermTagger`[11]. Parallèlement à la reconnaissance des termes, les entités nommées associées sont aussi recherchées. Par exemple, nous exploitons les ressources décrites plus haut pour reconnaître les quantités des ingrédients ou bien les durées nécessaires à la préparation d'un recette.

3.1.2 Pondération de termes

Pour identifier les termes les plus importants parmi ceux qui sont reconnus, nous les trions par ordre de pertinence avec plusieurs méthodes de pondération :

11. `http://search.cpan.org/~thhamon/Alvis-TermTagger/`

– La fréquence du terme dans le document (**tf**) ;
– Le nombre de documents du corpus où le terme apparaît (**df**) ;
– La position de la première occurrence du terme (**position**). Nous considérons ici que les termes situés au début du document ont un poids plus élevé que ceux situés à la fin ;
– Le cosinus de la position de la première occurrence du terme (**positionCos**). Nous faisons l'hypothèse que les termes qui apparaissent au début ou à la fin du texte sont les plus pertinents. Nous plaçons ainsi la première occurrence de chaque terme sur le cercle trigonométrique en considérant que le début du document est l'angle 0 et la fin l'angle 2π. Nous avons calculé le cosinus de l'angle formé par la position ;
– La fréquence de la forme canonique (lemmatisée) du terme (**canon**) ;
– Le fait que les termes sont quantifiés (**quant**). Ceci est essentiellement le cas des ingrédients.

3.1.3 Filtrage et sélection des termes

Les termes peuvent ensuite être filtrés ou regroupés suivant différents critères :

– Suppression des termes isolés étiquetés comme adjectifs (**filtrAdj**). Il s'agit alors de modifieurs de termes complexes, qui n'ont pas lieu d'apparaître tout seuls. Les adjectifs correspondant à des termes des ressources sont conservés ;
– Prise en compte de l'inclusion lexicale (**filtrInclLex**). Les termes en position tête d'un terme et ayant de rang plus élevé dans la liste triée sont supprimés. Par exemple, nous traitons de cette manière les termes comme {*crème anglaise*, *crème*}, {*pomme de terre*, *pomme*}, {*fève de tonka*, *fève*}. Les calculs des inclusions sont effectués au niveau syntaxique et au niveau des chaînes de caractères, lorsque la syntaxe ne détecte pas de relations ni d'inclusions ;
– Regroupement des termes en fonction de leur forme canonique (**filtrCan**). Il s'agit du regroupement des lemmes des composants et du filtrage par la forme fléchie la plus fréquente.

3.1.4 Appariement de titres

L'appariement de titres avec les recettes est effectué suivant les étapes suivantes :
– *Pré-traitement des titres et leur conversion en mots-clés.* À cette étape, les titres sont segmentés en mots, étiquetés morpho-syntaxiquement et lemmatisés (Schmid, 1994). En fonction des expériences, tous les mots ou seulement les mots "significatifs" (noms, adjectifs et verbes) sont retenus. Dans ce dernier cas, les mots grammaticaux et les mots de la liste d'exceptions ne sont pas considérés. Par exemple, le titre *Tarte au caramel et aux bananes* est converti en mots-clés suivants (*tarte, caramel, banane*).
– *Enrichissement de mots-clés avec les ressources linguistiques.* Les mots-clés sont étendus avec les ressources linguistiques : synonymes ou les mots associés selon les ressources. Chaque mot-clé, et les mots faisant partie de son extension, forment un cluster. Par exemple, le mot-clé *tarte* est enrichi avec *gâteau, pâtisserie, tartelette, gaufre, cake, galette, crêpe, beignet*, le mot-clé *banane* avec *pomme, fruit, abricot, poire*, et le mot-clé *caramel* avec *chocolat, beurre, vanille, croûton, pâte, sirop*. Les mots-clés sont considérés comme les labels sémantiques de leurs clusters : *tarte, banane, caramel*, respectivement. De cette manière, chaque titre est décrit par *n* clusters, en fonction du nombre de ses mots-clés.
– *Pré-traitement des recettes et leur sélection.* Les recettes sont pré-traitées de la même manière que les titres, et segmentées en phrases. Ensuite, les mots et termes des clusters sont appariés

avec les mots et les termes des recettes. C'est l'étape la plus importante : en plus de l'expansion et de l'appariement, une sélection des appariements entre les titres et les recettes est effectuée. Notons qu'à cette étape, il est possible d'effectuer l'appariement complet ou partiel (avec un pourcentage donné) entre les titres et les recettes. Par exemple, nous pouvons apparier le titre *Tarte au caramel et aux bananes* à la recette 90954 grâce à l'appariement direct et partiel (le mot-clé *tarte* n'apparaît pas dans le texte de la recette).

3.2 Système d'apprentissage automatique

Le système d'apprentissage est basé sur les fonctionnalités proposées par la plate-forme Weka (Witten et Frank, 2005). Plus particulièrement, nous utilisons et effectuons :
- le sur-échantillonnage avec l'algorithme *SMOTE* (Chawla *et al.*, 2002), particulièrement utile sur la tâche 1 (difficulté des recettes), où il existe un déséquilibre net dans les données ;
- la sélection d'attributs (avec *CFS combiné avec BestFirst* (Hall, 1998)), particulièrement utile pour rendre possibles les traitements étant donné le nombre total d'attributs ($n > 20\,000$) ;
- les algorithmes d'apprentissage testés sont : SVM, arbres de décision, régression logistique ;
- les attributs exploités sont assez variés, par exemple :
  - les lemmes ($n = 8\,145$) ;
  - les formes fléchies qui ne sont pas les lemmes ($n = 7\,830$) ;
  - les étiquettes morpho-syntaxiques ($n = 34$) ;
  - les étiquettes sémantiques provenant des ressources ($n = 61$), dont les types d'ingrédients, les ustensiles, les verbes d'action ;
  - le coût estimé d'une recette ;
  - la taille de la recette en nombre de caractères et de mots ;
  - le nombre d'étapes explicitement indiquées dans la recette ;
  - le nombre d'étapes implicites dans la recette ;
  - le nombre total d'étapes dans une recette ;
  - le nombre de lignes dans les ingrédients lorsque disponibles ;
  - la liste des actions ;
  - le nombre de documents du corpus où le terme apparaît (**df**) ;
  - le nombre d'occurrences des notions suivantes : sucre, sel, poivre, piment, huile, moutarde. Par exemple, lorsque *sucre, cassonade, sucre roux* sont reconnus, ils incrémentent la notion de sucre ;
  - les indications de temps exactes ou floues ;
  - les quantités exactes ou floues.

Les classes recherchées varient en fonction des tâches.

4 Recettes et tests réalisés

Pour aborder les tâches de la compétition, et comme *chaque tache a sa recette*, nous avons exploité le corpus d'entraînement pour mettre au point des stratégies différentes selon les tâches.

4.1 Tâche 1 : niveau de difficulté

Les textes des recettes sont d'abord traités avec le système à base de règles (section 3.1). Ensuite le système d'apprentissage (section 3.2) est appliqué. Les attributs exploités sont les suivants :
– les lemmes ;
– les étiquettes morpho-syntaxiques ;
– les étiquettes sémantiques provenant des ressources ;
– le coût estimé d'une recette ;
– la taille de la recette en nombre de caractères et de mots ;
– le nombre d'étapes explicitement indiquées dans la recette ;
– le nombre d'étapes implicites dans la recette ;
– le nombre total d'étapes dans une recette ;
– le nombre de lignes dans les ingrédients lorsque disponibles ;
– la liste des actions ;
– le nombre de documents du corpus où le terme apparaît (**df**) ;
– les indications de temps exactes ou floues ;
– les quantités exactes ou floues.
Il existe un déséquilibre dans les données du corpus d'entraînement, où les plat difficiles sont clairement sous-représentés :
– plats très faciles : 6 962,
– plats faciles : 5 752,
– plats moyennement difficiles : 1 068,
– plats difficiles : 80.
Nous effectuons donc un sur-échantillonage avec *SMOTE* : +1000% pour la classe 4, +200% pour la classe 3, combinaison des deux, tests d'autres valeurs par défaut. Nous pouvons ainsi constaté qu'il existe une amélioration nette lorsque le sur-échantillonage est effectué sur la classe 4 (la moins bien représentée).

Nous effectuons aussi une sélection d'attributs et pouvons par exemple constater que :
– très peu d'attributs apparaissent utiles ($n = 53$) : catégories morpho-syntaxiques, des lemmes, mais très peu de formes fléchies ce qui explique qu'elles ne sont pas utilisées,
– les catégories sémantiques ne sont pas saillantes,
– la taille et les sections des recettes sont utiles,
– les actions et le coût se révèlent importants aussi,
– les durées et le temps, exact ou flou, sont utiles,
– de même que la fréquence dans le corpus total de certains ingrédients.

4.2 Tâche 2 : type de plat

Les textes des recettes sont d'abord traités avec le système à base de règles (section 3.1). Ensuite le système d'apprentissage (section 3.2) est appliqué. Les attributs exploités sont les suivants :
– les lemmes ($n = 8\ 145$) ;
– les étiquettes morpho-syntaxiques ($n = 34$) ;
– les étiquettes sémantiques provenant des ressources ;
– le coût estimé d'une recette ;
– la taille de la recette en nombre de caractères et de mots ;
– le nombre d'étapes explicitement indiquées dans la recette ;

– le nombre d'étapes implicites dans la recette ;
– le nombre total d'étapes dans une recette ;
– le nombre de lignes dans les ingrédients lorsque disponibles ;
– la liste des actions ;
– la fréquence des ingrédients dans le corpus total (**df**) ;
– le nombre d'occurrences des notions suivantes : sucre, sel, poivre, piment, huile, moutarde.
– les indications de temps exactes ou floues ;
– les quantités exactes ou floues.

Nous avons testé le sur-échantillonnage, mais comme les données sont équilibrées pour cette tâche (3 246 entrées, 6 449 plats, 4 169 desserts), le sur-échantillonnage ne fournit pas de gain.

Nous effectuons aussi la sélection d'attributs et pouvons constater que :
– une réduction importante du nombre d'attributs est effectuée ($n = 72$),
– beaucoup de lemmes sont gardés, mais seulement 4 formes fléchies, qui ne seront donc pas utilisées pour cette tâche non plus,
– les catégories morpho-syntaxiques ne semblent pas être saillantes,
– les catégories sémantiques ne sont pas importantes,
– les quantités (exactes ou floues) de certains ingrédients (sel, poivre, moutarde) sont utiles,
– les indications sur les durées ne sont pas saillantes,
– certaines actions (comme *entrer*), dont la nominalisation est le type de plat *entrée*,
– le nombre de documents où apparaît l'ingrédient.

4.3 Tâche 3 : appariement du texte d'une recette à son titre

L'appariement des titres avec les textes des recettes est effectué avec l'approche décrite dans la section 3.1.4. Nous testons l'influence des ressources sémantiques (synonymes, distributionnelles) et constatons par exemple que les synonymes détériorent les résultats. Nous testons aussi l'utilisation de tous les mots ou des mots "significatifs" seulement et constatons que dans ce dernier cas, les résultats sont améliorés. Comme l'appariement à 100 % ne couvre pas la moitié de recettes, nous appliquons l'appariement partiel, ce qui permet d'augmenter la couverture. Afin de limiter l'appariement de plusieurs recettes à un même titre, lorsqu'un titre est affecté à une recette, celui-ci est retiré de l'ensemble des titres à assigner. Pour une recette donnée, les titres candidats sont ordonnés en fonction du rapport entre le nombre de mots appariés et le nombre maximum de mots appariés parmi tous les titres candidats. Parmi tous les titres candidats, le choix du titre est effectué de la manière suivante :
– le premier titre candidat, qui n'a pas déjà été assigné, est sélectionné,
– lorsqu'il existe aucun titre candidat non assigné, c'est le premier titre candidat qui est sélectionné.

Pour les recettes qui n'ont pas de titres avec des correspondances lexicales appariées ou bien lorsque cet appariement est inférieur à un seuil donné (par exemple, 75 % ou 50 %), aucun titre ne leur est assigné.

4.4 Tâche 4 : extraction de la liste des ingrédients

Pour l'extraction des ingrédients du texte des recettes, nous exploitons les étapes 3.1.1 à 3.1.3 du système basé sur les règles. Cela veut dire que les ressources sémantiques sont projetées sur les

Tâche/Run	Macro			Micro			MRR/MAP
	R	P	F	R	P	F	
T1/R1	0.399	0.332	0.363	0.580	0.580	0.580	
T1/R2	0.415	0.339	0.373	0.586	0.586	**0.586**	
T1/R3	0.406	0.329	0.364	0.574	0.574	0.574	
T2/R1	0.806	0.832	0.819	0.834	0.834	0.834	
T2/R2	0.841	0.842	0.841	0.849	0.849	**0.849**	
T2/R3	0.806	0.832	0.819	0.834	0.834	0.834	
T3/R1	0.036	0.019	0.025	0.036	0.036	0.036	0.0360
T3/R2	0.241	0.200	0.218	0.241	0.241	**0.241**	**0.2413**
T3/R3	0.133	0.119	0.125	0.133	0.133	0.133	0.1326
T4/R1							0.4115/0.4543
T4/R2							0.4170/0.4596
T4/R3							**0.4649**/0.4705

TABLE 2 – Performances officielles sur les quatre tâches sur le corpus de test

textes des recettes afin d'y reconnaître les ingrédients. Ensuite, les ingrédients sont pondérés avec les différentes méthodes. La solution qui s'avère être la plus efficace est la suivante : **position** * (**tf** des ingrédients quantifiés). Lorsque plusieurs ingrédients ont un poids égal, ils sont ordonnés en fonction de la fréquence **canon** dans la recette.

Les difficultés qui restent avec cette tâche sont :
– hésitation entre les formes courtes et étendues des ingrédients,
– hésitation entre les formes fléchies et lemmatisées des ingrédients.
Le calcul des inclusions lexicales syntaxiques et au niveau des chaînes de caractères a été effectué à cet effet. Cela a permis de faire des tests systématiquement, mais il reste difficile de reproduire la logique des données de référence et de la section *Ingrédients* des recettes. Le plus souvent, notre approche permet d'extraire les ingrédients mais leurs formes ou positions pondérées peuvent ne pas être correctes.

5 Résultats obtenus

Nous avons soumis trois runs pour chacune des quatre tâches de la compétition. Notre classement officiel est le suivant :
– tâche 1 : 4e/6
– tâche 2 : 3e/5
– tâche 3 : 2e/3
– tâche 4 : 5e/5
Les résultats détaillés pour ces tâches sont indiqués dans le tableau 2.

Selon les organisateurs de la compétition :
– Sur la première tâche (niveau de difficulté), six équipes ont participé. Les résultats globaux obtenus en termes de micro-mesure varient de 0,625 à 0,489 sur les meilleures soumissions de chaque équipe (micro mesure moyenne = 0,569, médiane = 0,589).
– Sur la deuxième tâche (type de plat), cinq équipes ont participé. Les résultats globaux

obtenus varient entre 0,889 et 0,746 (micro-mesure) sur les meilleures soumissions de chacun (moyenne de 0,833, médiane de 0,849).
– Sur la troisième tâche (appariement titre/recette), trois équipes ont participé. Les micro-mesures varient de 0,314 à 0,127 (moyenne de 0,227, médiane de 0,241). Le MRR (Mean Reciprocal Rank) calculé varie de 0,434 à 0,196 sur ces mêmes soumissions.
– Sur la quatrième tâche (normalisation des ingrédients), cinq équipes ont participé. Les résultats, évalués en termes de MAP (Mean Average Precision) varient, sur les meilleures soumissions de chaque équipe, entre 0,6662 et 0,4649 (moyenne de 0,5916, médiane de 0,6287).

Par rapport à d'autres participants, nous sommes au-dessus des moyennes sauf pour la tâche 4. Sur la tâche 4, les résultats produits comportent un défaut de format. Les résultats obtenus après la correction sont : 0.4543, 0.4596, 0.4705 pour les trois runs respectivement. Ces nouveaux résultats sont indiqués après / dans le tableau.

5.1 Tâche 1 : niveau de difficulté

Les runs évalués ont été obtenus avec l'algorithme SVM, le sur-échantillonnage avec *SMOTE*, et la sélection d'attributs. La différence entre les runs est la suivante :
– run 1 : tous les attributs sont utilisés,
– run 2 : les actions ne sont pas utilisées,
– run 3 : les lemmes ne sont pas utilisés, mais les actions sont utilisées.

Au sein de ces tests, c'est la configuration SVM, sur-échantillonnage, sélection d'attributs, et sans les actions qui montre de meilleurs résultats.

5.2 Tâche 2 : type de plat

Les runs évalués ont été obtenus avec l'algorithme SVM. La différence entre les runs est la suivante :
– run 1 : sélection d'attributs, utilisation des actions,
– run 2 : pas de sélection d'attributs, utilisation des actions,
– run 3 : sélection d'attributs, sans l'utilisation des actions.

C'est la configuration SVM, sans sélection d'attributs, et avec utilisation des actions qui donne les meilleurs résultats.

5.3 Tâche 3 : appariement du texte d'une recette à son titre

Les paramètres des runs de la tâche 3 sont :
– Run 1 : catégories majeures (noms, verbes, adjectifs) ; prise en compte de la liste d'exceptions, appariement partiel à 75 % ;
– Run 2 : tous les mots du titre, prise en compte de la liste d'exceptions, appariement partiel à 50 % ;
– Run 3 : catégories majeures (noms, verbes, adjectifs) ; prise en compte de la liste d'exceptions ; utilisation de la ressource distributionnelle ; appariement complet.

C'est le run 2 qui a montré de meilleurs résultats sur les données de test.

5.4 Tâche 4 : extraction de la liste des ingrédients

Pour cette tâche, nous utilisons le module Perl TermTagger. Les paramètres communs des runs soumis sont :
– utilisation de la liste des ingrédients constituée à partir de ressources en ligne,
– assignation du poids comme spécifié dans la section 4.4,
– les termes les plus grands calculés avec les inclusions syntaxiques.
La différence entre les runs est la suivante :
– run 1 : ajout de la liste des ingrédients du corpus d'entraînement,
– run 2 : paramétrage par défaut,
– run 3 : ajout des termes les plus grands calculés avec les inclusions au niveau des chaînes de caractères, et utilisation de FLEMM (Namer, 2000).
Le meilleur run est obtenu avec l'ajout des termes les plus grands calculés avec les inclusions au niveau des chaînes de caractères, et avec l'utilisation de FLEMM. Comme nous l'avons noté plus haut, souvent notre approche permet d'extraire les ingrédients mais leurs formes ou positions pondérées peuvent ne pas être correctes. Il reste en effet difficile de reproduire la logique des données de référence.

6 Conclusions et perspectives de perfectionnement

Nous avons présenté les expériences et tests effectués dans le cadre de la compétition DEFT 2013 dédiée au traitement des recettes de cuisine. Il s'agit d'un domaine intéressant où les informations exactes et floues co-existent et garantissent certainement la réussite de nos recettes de cuisine. Face aux systèmes de TAL, ces informations doivent être traitées de manière spéciale. Nous avons ainsi effectué une normalisation du flou vers de l'exact, en nous inspirant de la norme pifométrique dédiée (Delamasure, 2007) et des instructions précieuses trouvées en ligne. L'approche développée est basée sur un système à base de règles et un système d'apprentissage, de même que différentes ressources. Nous avons obtenus les résultats officiels suivants : 4e/6 sur la tâche 1, 3e/5 sur la tâche 2, 2e/3 sur la tâche 3, et 5e/5 sur la tâche 4.

En travaillant sur les recettes de cuisine, nous nous sommes rendus compte que ces recettes sont bien plus que de simples instructions pour bien remplir nos petits ventres. Les recettes de cuisines deviennent en effet une vitrine d'expression. Par exemple, nous pouvons y trouver l'appel à la consommation de la viande des alligators du Canada, la fierté pour les recettes familiales proposées, les conseils de vie et de bonne nourriture, sans parler des émotions, de l'humour et des opinions.

Il reste plusieurs perspectives à notre travail. Nous avons plusieurs perspectives scientifiques :
– tester mieux l'influence de différents attributs dans les tâches 1 et 2 ;
– faire d'autres tests sur l'ordonnancement des ingrédients grâce à l'utilisation des CRF dans la tâche 4 ;
– effectuer d'autres tests avec les ressources pour l'expansion de titres et l'ordonnancement de titres dans la tâche 3 ;
– prendre en compte d'autres facteurs du flou comme les erreurs d'orthographe ;
– prendre en compte les anaphores ;
– exploiter les relations sémantiques entre les ingrédients pour réduire le nombre d'attributs.

Parmi les perspectives culinaires, nous aimerions tester autant de recettes que possible, moyennant la possibilité de trouver les ingrédients nécessaires (viandes d'alligator, jésus...).

Références

ANDREEVA, V., MARTIN, C., ISSANCHOU, S., HERCBERG, S., KESSE-GUYOT, E. et MÉJEAN, C. (2013). Sociodemographic profiles regarding bitter food consumption. cross-sectional evidence from a general french population. *Appetite*, 67(3):53–60.

ANGUIANO, E. et DENIS, P. (2011). FreDist : Automatic construction of distributional thesauri for French. *In TALN*, pages 119–124.

AUBIN, S. et HAMON, T. (2006). Improving term extraction with terminological resources. *In* SALAKOSKI, T., GINTER, F., PYYSALO, S. et PAHIKKALA, T., éditeurs : *Advances in Natural Language Processing (5th International Conference on NLP, FinTAL 2006)*, numéro 4139 de LNAI, pages 380–387. Springer.

BODENREIDER, O., BURGUN, A. et RINDFLESCH, T. (2001). Lexically-suggested hyponymic relations among medical terms and their representation in the UMLS. *In* URI INIST CNRS, éditeur : *Terminologie et Intelligence artificielle (TIA)*, pages 11–21, Nancy.

CHAWLA, N. V., BOWYER, K. W., HALL, L. O. et KEGELMEYER, W. P. (2002). Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

DALE, R. (1989). Cooking up referring expressions. *In Annual meeting on Association for Computational Linguistics*, pages 68–75.

DE RUDDER, O. (2006). *Aux petits oignons ! : Cuisine et nourriture dans les expressions de la langue française*. Larousse.

DELAMASURE, M. (2007). Norme française NF UNM 00-003. système d'unités pifométriques. Rapport technique, NA.

DUFOUR-LUSSIER, V., LE BER, F., LIEBER, J. et NAUER, E. (2013). Automatic case acquisition from texts for process-oriented case-based reasoning. *Information Systems*. Comming soon.

DUMAS, A. (1873). *Le grand dictionnaire de cuisine*. Editions Pierre Grobel.

GREFENSTETTE, G. et TAPANAINEN, P. (1994). What is a word, what is a sentence ? Problems of tokenization. *In COMPLEX (Computational lexicography and text research)*, pages 79–87.

HALL, M. A. (1998). *Correlation-based Feature Subset Selection for Machine Learning*. Thèse de doctorat, University of Waikato, Hamilton, New Zealand.

HAMON, T. (2012). Acquisition terminologique pour identifier les mots clés d'articles scientifiques. *In Actes de l'atelier DEFT 2012*, pages 25–31, Grenoble, France.

HAMON, T. et GAGNAYRE, R. (2012). Linking expert and lay knowledge with distributional and synonymy resources : application to the mining of diabetes fora. *In Proceedings of The Fourth Swedish Language Technology Conference (SLTC 2012)*, pages 35–36, Lund, Sweden.

HARRIS, Z. (1954). Distributional structure. *Word*, 10(23):146–162.

HATHOUT, N., NAMER, F. et DAL, G. (2001). An experimental constructional database : the MorTAL project. *In* BOUCHER, P., éditeur : *Morphology book*. Cascadilla Press, Cambridge, MA.

JACQUEMIN, C. (1996). A symbolic and surgical acquisition of terms through variation. *In* WERMTER, S., RILOFF, E. et SCHELER, G., éditeurs : *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, pages 425–438, Springer.

JONES-SMITH, J., KARTER, A., WARTON, E., KELLY, M., KERSTEN, E., MOFFET, H., ADLER, N., SCHILLINGER, D. et LARAIA, B. (2013). Obesity and the food environment : Income and ethnicity differences among people with diabetes : The diabetes study of Northern California (DISTANCE). *Diabetes Care*, 36(1):1200–1208.

KAMODA, R., UEDA, M., FUNATOMI, T., IIYAMA, M. et MINOH, M. (2012). Grocery re-identification using load balance feature on the shelf for monitoring grocery inventory. *In Proceedings of the Cooking with Computers workshop (CwC)*, pages 8–18.

LIM, J. S. (2007). Description structurale des proverbes coréens autour des « noms de cuisine ». *In Proceedings of the 26th conference on Lexis and Grammar*, Bonifacio.

MORIN, E. et JACQUEMIN, C. (2004). Automatic Acquisition and Expansion of Hypernym Links. *Computers and the Humanities*, 38(4):363–396.

NAMER, F. (2000). FLEMM : un analyseur flexionnel du français à base de règles. *Traitement automatique des langues (TAL)*, 41(2):523–547.

NLM (2011). *UMLS Knowledge Sources Manual*. National Library of Medicine, Bethesda, Maryland. www.nlm.nih.gov/research/umls/.

OTA, S., SOGA, M., YAMAMOTO, N. et TAKI, H. (2012). Design and development of a learning support environment for apple peeling using data gloves. *In Proceedings of the Cooking with Computers workshop (CwC)*, pages 7–12.

ROBERT, L. (1990). *Le petit Robert*. Le Robert, Paris.

SCHMID, H. (1994). Probabilistic part-of-speech tagging using decision trees. *In Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

WITTEN, I. et FRANK, E. (2005). *Data mining : Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco.

DEFT2013, une cuisine de caractères

Gaël Lejeune[1] Charlotte Lecluze[1] Romain Brixtel[1]
(1) Équipe HULTECH (GREYC - CNRS UMR 6072 - Université de Caen),
Bd Maréchal juin, 14032 Caen Cedex
prenom.nom@unicaen.fr

RÉSUMÉ

Nous présentons dans cet article les méthodes utilisées par l'équipe HULTECH pour sa participation au DÉfi Fouille de Textes 2013 (DEFT2013). Cette neuvième édition porte sur l'analyse automatique de recettes de cuisine en langue française. Elle comporte quatre tâches : trois de classification de documents et une d'extraction d'information. Notre équipe participe aux quatre tâches. Nous nous appuyons pour chaque tâche sur une technique d'algorithmique du texte : la détection de chaînes de caractères répétées maximales ($rstr_{max}$). Les méthodes développées sont simples et non supervisées.

ABSTRACT

DEFT2013, a distinctive character-based cuisine

We present here the HULTECH (Human Language Technology) team approach for the DEFT2013 (french text mining challenge). The aim of the challenge is to automatically analyze recipes in French. It has four tasks : three of document classification and one of information extraction. Our team participate in four tasks. Our methods relie on a text algorithmics technic : detection of maximal repeated strings ($rstr_{max}$). The developed methods are simple and unsupervised.

MOTS-CLÉS : classification, extraction d'information, appariement, algorithmique du texte.

KEYWORDS: classification, information extraction, matching, text algorithmics.

1 Introduction

Pour cette nouvelle édition du DÉfi Fouille de Textes, quatre tâches d'analyse concernant les recettes de cuisine ont été proposées :

Niveau de difficulté de réalisation d'une recette : La tâche 1 a pour but d'évaluer la capacité d'un algorithme à inférer la difficulté d'une recette de cuisine en se basant sur le texte de la recette et son titre (cf. Figure 1). L'appréciation de la difficulté est donnée sur une échelle à 4 valeurs : très facile, facile, moyennement difficile, très difficile.

Type de plat : À partir du texte et du titre, la tâche 2 propose de classer les recettes en fonction du type de plat préparé, selon trois classes : entrée, plat principal, dessert.

Appariement titre/recette : La tâche 3 consiste à retrouver pour chaque texte de recette, son titre original dans une liste de titres de recettes. Pour chaque texte de recette, le système doit fournir une liste de titres par ordre de pertinence décroissante.

Ingrédients d'une recette : La tâche 4 diffère des précédentes car elle ne concerne pas la classification des recettes, mais l'extraction d'information. Dans cette tâche, il s'agit d'identifier la liste des ingrédients de la recette au moyen d'une liste normalisée globale de libellés d'ingrédients fournie aux participants. Cette liste contient tous les ingrédients présents dans la base des textes de recettes, mais peut aussi contenir des ingrédients qui ne sont pas présents dans les recettes.

2 Description des corpus d'apprentissage et de test

Le corpus utilisé comporte des recettes de cuisine (Figure 1) provenant du site Marmiton. 60% de ce corpus sert pour l'apprentissage, les 40% restant sont utilisés pour le test. La proportion de document de chaque classe entre apprentissage et test est la même. L'équilibre entre corpus d'apprentissage et corpus de test s'arrête là, la taille des fichiers n'ayant pas été vérifiée.

2.1 Corpus d'apprentissage

Le corpus d'apprentissage comprend 13863 documents. Dans le tableau 1, nous présentons la répartition des recettes du corpus dans les différentes classes des tâches 1 et 2.

Type \ Difficulté	Très facile	Facile	Moyennement difficile	Difficile	Total
Entrée	1787 (55,1%)	1259 (38,8%)	189 (5,8%)	10 (0,3%)	3245
Plat principal	3025 (46,9%)	2897 (44,9%)	496 (7,7%)	29 (0,5%)	6449
Dessert	2150 (51,6%)	1595 (38,2%)	383 (9,2%)	41 (1%)	4169
Total	6962	5751	**1068**	**80**	13863 / 13861

Tableau 1 – Répartition des recettes par type de plat et difficulté dans le corpus d'apprentissage. La différence entre les totaux globaux correspond à l'absence de l'étiquette difficulté pour deux plats principaux.

0. http://www.marmition.org

```
<recette id="94727">
<titre>Truites tricolores</titre>
<type>Entrée</type>
<niveau>Moyennement difficile</niveau>
<cout>Moyen</cout>
<ingredients>
<p>2 filets de truite</p>
<p>2 grosses tomates</p>
<p>15g de pignons grillés</p>
<p>1 échalote</p>
<p>zeste de citron</p>
<p>court-bouillon</p>
<p>Pour le pesto:</p>
<p>un gros bouquet de basilic</p>
<p>15-20g de pignons grillés</p>
<p>30-40g de parmesan</p>
<p>1 petite gousse d'ail</p>
<p>20g d'huile d'olive</p>
<p>sel, poivre</p>
</ingredients>
<preparation>
<![CDATA[
```
Après avoir inciser les tomates, les plonger quelques secondes dans de l'eau frémissante, les ressortir et les peler. Les couper en deux, les épépiner, les poser sur une plaque huilée (huile d'olive) allant au four, les mettre bien à plat, y poser dessus l'échalote hachée finement ainsi qu'un peu de zeste de citron, saler, poivrer. Les laisser au four (150°) pour 20 minutes.
En parallèle, plonger les filets de truite dans un court-bouillon tiède [...]
Mixer tous les ingrédients du pesto ou acheter un petit verre de pesto. Mais c'est meilleur fait maison ! Tartiner le pesto sur les truites. Couvrir le peso avec les lamelles de tomates et manger tout de suite sur des assiettes préchauffées.Décorer avec les pignons grillés et un filet de court-bouillon ou un filet d'huile d'olive.on peut le prendre comme plat principal en mettant les portions doubles (2 filets par personne). Un bon verre de Riesling serait un bon accompagnement.
```
 ]]>
 </preparation>
</recette>
```

FIGURE 1 – Un exemple de recette

2.2 Corpus de test

Un corpus de test différent était fourni pour chaque tâche.

Type \ Difficulté	Très facile	Facile	Moyennement difficile	Difficile	Total
Entrée	298 (52,5%)	234 (41,3%)	34 (6%)	1 (0,2%)	567
Plat principal	477 (45,5%)	471 (44,9%)	91 (8,7%)	9 (0,9%)	1048
Dessert	357 (51,5%)	262 (37,8%)	64 (9,2%)	10 (1,5%)	693
Total	1132	967	189	20	2308

Tableau 2 – Répartition des recettes par type de plat et difficulté, corpus de test de la tâche 1.

Type \ Difficulté	Très facile	Facile	Moyennement difficile	Difficile	Total
Entrée	310	215	34	3	562
Plat principal	518	473	87	5	1083
Dessert	334	251	72	4	661
Total	1162	939	193	12	2306

Tableau 3 – Répartition des recettes par type de plat et difficulté, corpus de test de la tâche 2.

Type \ Difficulté	Très facile	Facile	Moyennement difficile	Difficile	Total
Entrée	283	224	39	0	546
Plat principal	507	495	74	0	1076
Dessert	349	264	65	7	685
Total	1139	983	178	7	2307

Tableau 4 – Répartition des recettes par type de plat et difficulté, corpus de test de la tâche 3.

Type \ Difficulté	Très facile	Facile	Moyennement difficile	Difficile	Total
Entrée	300	166	18	2	486
Plat principal	512	489	77	5	1083
Dessert	392	286	52	6	736
Total	1204	941	147	13	2305

Tableau 5 – Répartition des recettes par type de plat et difficulté, corpus de test de la tâche 4.

3 Description de la méthode et résultats

Pour toutes ces tâches, nous exploitons les mêmes concepts : les chaînes de caractères répétées maximales et les affinités. Nous présentons ci-après les caractéristiques de ces deux concepts.

Ainsi, pour chaque tâche, nous procédons à une recherche de chaînes de caractères répétées maximales, des $rstr_{max}$. Ces chaînes sont déduites d'un tableau de suffixes. Elles sont obtenues en calculant des motifs sans trou tels que décrits par (Ukkonen, 2009) [1]. Ces chaînes possèdent les caractéristiques suivantes :

répétées : les chaînes ont un effectif de 2 ou plus ;

maximales : les chaînes ne peuvent être étendues à gauche ou à droite sans perdre une occurrence.

Le tableau 6 contient les répétitions maximales les plus longues extraite de l'exemple de recette de la figure 1.

1. Les outils permettant le calcul de ces chaînes sont disponibles ici : `https://code.google.com/p/py-rstr-max/`

Nombre de caractères	$rstr_{max}$	Effectif dans la recette
32	'g de pignons grillés '	2
22	' les filets de truite '	2
17	' filets de truite'	3
16	'e court-bouillon'	2
16	' pignons grillés'	3
16	' d'huile d'olive'	2
16	' court-bouillon '	2
15	'zeste de citron'	2
15	'ingredients> '	2
15	' court-bouillon'	3

Tableau 6 – Exemples des $rstr_{max}$ les plus longues dans la recette de la figure 1 « Truite tricolores ».

À l'occasion de notre participation à la campagne DEFT2011 sur l'appariement entre des articles scientifiques et leur résumé , nous avions introduit le concept d'affinité (Lejeune *et al.*, 2011). Une affinité est une $rstr_{max}$ commune à au moins deux documents et qui possède certaines caractéristiques de longueur et d'effectif. La présence d'affinités « hapax » dans la collection d'articles et dans la collection de résumés était caractéristique d'un appariement correct.

Ici, nous utilisons la notion d'affinité pour mettre en relation différentes recettes. Deux recettes partageant un grand nombre d'affinités sont donc considérées comme proches.

À partir de deux recettes du corpus d'apprentissage, nous illustrons la notion d'affinité :

Nombre de caractères	affinité	Effectif
93	'</type> <niveau>Moyennement difficile</niveau>' ' <cout>Moyen</cout> <ingredients> '	2
55	'e </ingredients> <preparation> <![CDATA['	2
34	']]> </preparation> </recette>'	2
25	' gousse d'ail '	2
15	' sel'	2
14	'un peu d'huile'	2
14	' les tomates, '	2

Tableau 7 – Exemples d'affinités entre la recette de la « Truite tricolores » et celle du « Carry de poulet ».

3.1 Tâche 1 : identification du niveau de difficulté

Les recettes de même difficulté ont des contenus proches. Cette similarité entre recette est fondée sur les chaînes de caractères communes entre ces recettes. Nous calculons la similarité entre recettes en utilisant le concept d'affinité décrit précédemment. Ainsi, deux recettes ayant

beaucoup d'affinités entre elles possèdent, par définition, beaucoup de $rstr_{max}$. Nous avons établi une étiquette par défaut : l'étiquette « Facile ». Au début du traitement, toute recette que nous cherchons à étiqueter reçoit l'étiquette « facile ». La méthode de classification calcule les probabilités d'allocation d'une autre étiquette. Si jamais aucune autre étiquette n'est trouvée, alors la méthode considère la recette comme facile, sinon la nouvelle étiquette est attribuée à la recette.

Partant de cette hypothèse, nous avons établi la chaîne de traitement suivante :

1. Comparaison de la recette à étiqueter avec les recettes dont l'étiquette est connue
2. Classement des appariements par ordre décroissant du nombre d'affinités
3. Sélection des n meilleurs appariements
4. Calculer l'effectif de chaque étiquette dans cette sélection
5. Si une étiquette a un effectif supérieur à $n/2$: elle est choisie [2]

De façon purement arbitraire, nous avons choisi trois valeurs de n : 10, 20 et 50. Pour $n = 10$, le système avait tendance à surgénérer les étiquettes « difficile » et « moyennement difficile ». Pour $n = 50$, il était très rare sur le corpus d'apprentissage qu'une majorité se détache. Sur le corpus de test, cela a même conduit à ce qu'il n'y ait jamais d'autre étiquette attribuée que l'étiquette par défaut « Facile ». Nous avons donc produit une *baseline* naïve qui attribue automatiquement l'étiquette par défaut. Les meilleurs résultats ont été obtenus pour $n = 20$. La plus-value vis-à-vis de la *baseline* restant toutefois très faible.

Run	Mesure	Rappel	Précision	F_1-mesure
# 1	Macro	0.273	0.250	0.261
# 1	Micro	0.489	0.489	0.489
# 2	Macro	0.265	0.248	0.256
# 2	Micro	0.465	0.465	0.465
# 3	Macro	0.289	0.281	0.285
# 3	Micro	0.360	0.360	0.360

Tableau 8 – Résultats sur la tâche d'identification de la difficulté.

3.2 Tâche 2 : identification du type de plat préparé

Pour la tâche d'identification du type de plat, nous avons appliqué une technique très proche de celle utilisée pour la détection de la difficulté. Les classes étant équilibrées, l'étiquette choisie étant cette fois simplement l'étiquette majoritaire sur les n premiers appariements triés par nombre d'affinités.

Nous avons considéré que l'on pouvait discriminer plus facilement les desserts à travers certains termes techniques et ingrédients. À l'opposé, nous avons fait l'hypothèse que les indices pour déterminer qu'une recette était un plat principal était plus rare. En cas d'égalité, nous avons gardé l'étiquette supposée la plus difficile à identifier : entrée *vs* plat ou dessert et plat *vs* dessert.

2. Sinon on conserve l'étiquette par défaut

Run	Mesure	Rappel	Précision	F_1-mesure
#1	Macro	0.760	0.741	0.750
#1	Micro	0.746	0.746	0.746
#2	Macro	0.584	0.609	0.596
#2	Micro	0.573	0.573	0.573
#3	Macro	0.381	0.418	0.398
#3	Micro	0.331	0.331	0.331

Tableau 9 – Résultats sur la tâche d'identification du type de plat.

3.3 Tâche 3 : Appariement du texte d'une recette avec son titre

Pour cette tâche, nous n'avons utilisé le corpus d'apprentissage que pour éprouver le système. Elle est basée sur le nombre d'affinités existant entre une recette et chacun des titres de la base de titres. Nous ne conservons que les activités qui sont hapax dans la base de titres. En cas d'égalité entre plusieurs titres, aucun départage n'est effectué.

Quand un appariement est effectué, on continue le processus en ne tenant plus compte du titre déjà utilisé. Notre idée est qu'à l'issue d'une première phase où les appariements les plus évidents sont effectués, on essaie d'apparier les recettes restantes à un des titres non utilisés.

Il s'est avéré que la première phase donnait de très bons résultats sur le corpus d'apprentissage mais que les résultats se détérioraient rapidement par la suite. Une fois les appariements les plus évidents effectués dans la première phase, les appariements ultérieurs étaient de très mauvais qualité. Notre système de donnait qu'un titre possible par recette, ce qui explique que le rappel et la précision soient rigoureusement identiques (Tableau 10).

Run	Mesure	Rappel	Précision	F_1-mesure
#1	Macro	0.127	0.127	0.127
#1	Micro	0.127	0.127	0.127

Tableau 10 – Résultats sur la tâche d'appariement texte-titre, moyenne réciproque des rangs (*Mean Reciprocal Rank*) : 0.1956.

3.4 Tâche 4 : extraction des ingrédients à partir du titre et du texte de la recette

Pour l'extraction des ingrédients, la méthode est très proche de celle utilisée pour l'appariement titre-recette. On suppose qu'un ingrédient est utilisé dans une recette si l'on trouve une sous-chaîne de cet ingrédient dans le texte. De manière à éviter la surgénération d'ingrédients, un filtrage était opéré : on ne conserve un ingrédient que s'il n'est pas une sous-chaîne d'un autre ingrédient extrait. Par exemple si le système extrait « citron » et « jus de citron », on conserve seulement le second terme. C'est une *baseline* assez naïve, nous l'avons utilisé pour le run1.
Le run2 ajoutait une heuristique que nous avions déduite du corpus d'apprentissage : étant donné f la fréquence d'apparition d'un ingrédient dans la série de recettes, on déduit p la probabilité

maximale d'apparition du même ingrédient qui est égale à $2.f$. S'il s'avère qu'un ingrédient a une probabilité d'apparition dans le corpus de test supérieure à cet attendu, on introduit des contraintes. L'ingrédient en question n'est sélectionné que s'il apparaît à des positions déterminées : débuts et fins de texte. Le début et la fin du texte de la recette sont les 20 premiers et les 20 derniers caractères. Cette heuristique permettait d'écarter des ingrédients extraits trop fréquemment tels que « dore », « plie », « rose » ou « renne ». Le run 3 remplace cette heuristique par une heuristique sur la longueur des ingrédients, le système n'extrait pas d'ingrédient dont la longueur en caractères est inférieure à 4.

	Run #1	Run #2	Run #3
MAP	0.4881	0.5556	0.5076

Tableau 11 – Moyenne de la précision moyenne (*Mean Average Precision, MAP*) par run sur la tâche d'extraction des ingrédients.

4 Discussion

L'édition 2013 du DÉfi Fouille de Textes proposait l'analyse automatique de recettes de cuisine en langue française. Nous avons présenté pour chaque tâche une méthode fondée sur une analyse au grain caractère. Les résultats que nous avons obtenus ont été décevants puisque nous avons terminé à la dernière place sur chaque tâche à l'exception de la tâche 4 (extraction des ingrédients). Il été intéressant de tester des techniques d'apprentissage en exploitant les caractéristiques issues de notre analyse au grain caractère.

Remerciements

Nous remercions les organisateurs du DEFT2013 d'avoir proposé une tâche originale.

Références

LEJEUNE, G., BRIXTEL, R., GIGUET, E. et LUCAS, N. (2011). Deft2011 : appariement de résumés et d'articles scientifiques fondé sur les chaînes de caractères. *In Défi Fouille de Textes/TALN 2011*, pages 53–64.

UKKONEN, E. (2009). Maximal and minimal representations of gapped and non-gapped motifs of a string. *Theorie in Computer Science*, 410(43):4341–4349.

Systèmes du LIA à DEFT'13

Xavier Bost[1] Ilaria Brunetti[1,6] Luis Adrián Cabrera-Diego[1]
Jean-Valère Cossu[1] Andréa Linhares[1,5] Mohamed Morchid[1]
Juan-Manuel Torres-Moreno[1,2,3,4] Marc El-Bèze[1,2,4] Richard Dufour[1,4]
(1) LIA, 339, chemin des Meinajariès 84912 Avignon Cedex 09
(2) SFR Agorantic Université d'Avignon et des Pays de Vaucluse, 84000 Avignon Cedex
(3) École Polytechnique de Montréal, 2900 Bd Edouard-Montpetit Montréal, QC H3T1J4
(4) Brain & Language Research Institute, 5 avenue Pasteur, 13604 Aix-en-Provence Cedex 1
(5) Universidade Federal do Ceara Rua Estanislau Frota, S/N, CEP 62.010-560 – Sobral/ CE Brésil
(6) Equipe Maestro INRIA, 2004, Route des Lucioles, 06902 Sophia Antipolis
prenom.nom@univ-avignon.fr

RÉSUMÉ

La campagne Défi de Fouille de Textes (DEFT) en 2013 s'est intéressée à deux types de fonctions d'analyse du langage, la classification de documents et l'extraction d'information dans le domaine de spécialité des recettes de cuisine. Nous présentons les systèmes du LIA appliqués à DEFT 2013. Malgré la difficulté des tâches proposées, des résultats intéressants ont été obtenus par nos systèmes.

ABSTRACT

The 2013 *Défi de Fouille de Textes* (DEFT) campaign is interested in two types of language analysis tasks, the document classification and the information extraction in the specialized domain of cuisine recipes. We present the systems that the LIA has used in DEFT 2013. Our systems show interesting results, even though the complexity of the proposed tasks.

MOTS-CLÉS : Classification de textes ; Extraction d'information ; Méthodes statistiques ; Domaine de spécialité.

KEYWORDS: Text Classification ; Information Extraction ; Statistical Methods ; Speciality domain.

1 Description des tâches et des données

Le Défi Fouille de Textes (DEFT) s'intéresse en 2013 à deux types de fonction d'analyse du langage, la classification de documents (tâches 1 à 3) et l'extraction d'information (tâche 4), ceci dans un domaine de spécialité : les recettes de cuisine. Pour cette nouvelle édition du défi [1], l'équipe du Laboratoire Informatique d'Avignon (LIA) a décidé de participer aux tâches suivantes :

– *T1. L'identification à partir du titre et du texte de la recette de son niveau de difficulté sur une échelle à 4 niveaux : très facile, facile, moyennement difficile, difficile ;*

1. http://deft.limsi.fr/2013/

– *T2. L'identification à partir du titre et du texte de la recette du type de plat préparé : entrée, plat principal, dessert ;*
– *T4. L'extraction à partir du titre et du texte d'une recette de la liste de ses ingrédients.*

Le corpus, fourni par DEFT au format *XML*, est composé de 13 684 recettes. Nous avons choisi d'augmenter notre base de données en extrayant des recettes de cuisine à partir de 4 sites spécialisés : Cuisine AZ (`http://www.cuisineaz.fr`), Madame Le Figaro (`http://madame.lefigaro.fr/recettes`), 750 grammes (`http://www.750g.com/`) et Ptit Chef (`http://www.ptitchef.com/`). Environ 50k recettes supplémentaires ont pu être obtenues à partir de ces sites. Notons cependant que les recettes téléchargées ne comportaient pas toujours les mêmes champs d'information que ceux contenus dans les recettes fournies par les organisateurs de la campagne.

2 Méthodes utilisées

Pour répondre à ce défi, nous avons employé plusieurs méthodes de classification (modèles probabilistes discriminants, *boosting* de caractéristiques textuelles) et d'extraction d'information, ainsi que leurs combinaisons.

2.1 Modèles discriminants

Le système DISCOMP_LIA a été appliqué aux tâches 1 et 2. Son fonctionnement repose essentiellement sur les modèles discriminants décrits dans (Torres-Moreno et al., 2011). Ces modèles servent aussi bien à agglutiner des mots afin de produire des termes composites ayant un fort pouvoir discriminant (sans toutefois passer en dessous d'un seuil préfixé de couverture), qu'à pondérer de façon plus appropriée ces termes dans le calcul de l'indice de similarité (ici Cosine revisitée) entre chaque recette et le sac de mots de chacune des classes (3 ou 4 selon les tâches).

Trois nouveautés ont été introduites pour prendre en compte les particularités du défi :
– nous avons opté pour une approche hiérarchique propre à chacune des 2 tâches :
  – pour la tâche T1 : identification du niveau *facile* ou *difficile* d'une recette, puis dans chacun des 2 groupes, différenciation en 2 sous-groupes (*moyennement* ou *très difficile*) ;
  – pour la tâche T2 : identification du type *dessert* ou *autre* ; dans le cas d'une identification en type *autre*, différenciation entre *plat principal* et *entrée* ;
– nous avons appliqué le même système avec un paramétrage *ad hoc* à 2 jeux de données : le titre de la recette seul pour le premier et pour le second l'ensemble formé par le contenu et le titre de la recette. Une combinaison linéaire entre les scores des 2 systèmes a permis d'obtenir une nouvelle distribution de probabilités sur les hypothèses de classes ;
– la formation par agglutination des termes et ce, surtout dans le titre, a fait apparaître des sous-types de recettes (quiches, soupes, gâteaux) qui ont permis fournir aux modèles de meilleurs points d'appui pour identifier la classe visée. Dans les cas où l'indice de pureté de *Gini* valait 1, et où la couverture était élevée, nous avons enrichi le sac de mots avec des composants factices de façon à compenser le bruit qui pouvait être engendré par le reste de la recette. Ceci a été fait de façon semi-automatique, mais devrait pouvoir être complètement automatisé.

Notons par ailleurs que nous nous sommes servis du critère de pureté de *Gini* pour sélectionner quelques mots erronés ce qui, après correction, a permis d'augmenter leur couverture sans altérer leur pouvoir discriminant.

2.2 Algorithmes de *boosting*

Pour les tâches de classification (1 et 2), nous proposons une manière de combiner différentes caractéristiques textuelles et numériques au moyen d'un algorithme d'apprentissage automatique : le *Boosting* (Schapire, 2003). Son objectif principal est d'améliorer (de « booster ») la précision de n'importe quel algorithme d'apprentissage permettant d'associer, à une série d'exemples, leur classe correspondante. Le principe général du *Boosting* est assez simple : la combinaison pondérée d'un ensemble de classifieurs binaires (appelés classifieurs *faibles*), chacun associé à une règle différente de classification très simple et peu efficace, permettant au final d'obtenir une classification robuste et très précise (classifieur *fort*). La seule contrainte de ces classifieurs *faibles* est d'obtenir des performances meilleures que le hasard. Dans le cas des classifieurs binaires, l'objectif est de classifier plus de 50 % des données correctement.

Nous proposons d'utiliser l'algorithme *AdaBoost* car il présente de nombreux avantages dans le cadre des tâches de classification proposées. En effet, l'algorithme est très rapide et simple à programmer, applicable à de nombreux domaines, adaptable aux problèmes multi-classes. Il fonctionne sur le principe du *Boosting*, où un classifieur *faible* est obtenu à chaque itération de l'algorithme *AdaBoost*. Chaque exemple d'apprentissage possède un poids, chaque tour de classification permettant de les re-pondérer selon le classifieur *faible* utilisé. Le poids d'un exemple bien catégorisé est diminué, au contraire d'un exemple mal catégorisé, pour lequel on augmente son poids. L'itération suivante se focalisera ainsi sur les exemples les plus « difficiles » (poids les plus élevés). L'algorithme a été implémenté dans l'outil de classification à large-marge *BoosTexter* (Schapire and Singer, 2000), permettant de fournir comme caractéristiques au classifieur des données numériques mais également des données textuelles. Les classifieurs *faibles* sur les données textuelles peuvent prendre en compte des *n*-grammes de mots. Nous avons utilisé l'outil *IcsiBoost* (Favre et al., 2007), qui est l'implémentation open-source de cet outil. *IcsiBoost* a notamment l'avantage de pouvoir fournir, pour chaque exemple à classifier, un score de confiance pour chacune des classes entre 0 (peu confiant) et 1 (très confiant).

2.2.1 Niveau de difficulté

Pour l'approche utilisant l'algorithme de classification *AdaBoost*, nous avons retenu différentes caractéristiques qui ont été fournies en entrée pour l'apprentissage du classifieur :

– données textuelles
  1. *n*-grammes de mots du titre de la recette avec taille maximum de 3 mots ;
  2. *n*-grammes de mots du texte de la recette avec taille maximum de 4 mots ;
  3. uni-gramme sur la liste des ingrédients obtenue automatiquement (voir section 2.6).

– données numériques continues
  1. nombre de mots du titre de la recette ;
  2. nombre de mots du texte de la recette ;

3. nombre de phrases du texte de recette ;
4. nombre de séparateurs du texte de la recette (point, virgule, deux-points, etc.) ;
5. taille de la liste des ingrédients obtenue automatiquement.

À la fin de ce processus d'apprentissage, la liste des classifieurs sélectionnés est obtenue tout comme le poids de chacun d'eux, afin d'utiliser les exemples les plus discriminants pour chaque classe (chaque niveau de difficulté est considéré dans cette tâche comme une classe). Nous avons composé un sous-corpus à partir du corpus d'apprentissage fourni par les organisateurs de la tâche afin d'optimiser le nombre de classifieurs *faibles* nécessaires. Ce sous-corpus est composé de 3 863 recettes respectant la distribution des classes, le reste étant utilisé pour l'apprentissage (environ 72% des données d'apprentissage). Notons que tout le corpus d'apprentissage sera néanmois utilisé pour entraîner les classifieurs pendant la phase de test de la campagne. Enfin, une attention particulière a été portée sur la normalisation des données textuelles du titre et de la description des recettes. En effet, les données textuelles « brutes » fournies par les organisateurs possédaient un vocabulaire très bruité principalement à cause des nombreuses abbréviations (par exemple « th » pour « thermostat ») et fautes d'othographe (par exemple « échalote » et « échalotte »). Quatre traitements particuliers ont été appliqués sur le texte :

1. suppression de la ponctuation et isolation de chaque mot (exemple : « et couper l'oignon. » devient « et␣couper␣l'␣oignon ») ;
2. utilisation d'une liste manuelle de normalisation de mots (exemple : « kg » devient « kilogramme ») ;
3. conversion automatique des chiffres en lettres ;
4. regroupement des *n*-grammes de mots les plus fréquents au sein d'une même entité au moyen de l'outil *lia_tagg* [2] (exemple : « il y a » est considéré comme un mot « il_y_a »).

Ce texte normalisé est utilisé par l'algorithme *AdaBoost* pour la recherche des classifieurs *faibles* sur les données textuelles lors de la phase d'apprentissage, mais également pour la recherche du niveau de difficulté associé à une recette lors de la phase de test. Au final, nous obtenons pour chaque recette à classifier, un score de confiance pour chaque niveau de difficulté, permettant de choisir le niveau de difficulté de la recette (score le plus élevé).

2.2.2 Type de plat

Dans ce problème de classification au moyen de l'approche par *Boosting*, nous avons utilisé les mêmes caractéristiques et normalisations que celles décrites dans la tâche de recherche du niveau de difficulté (voir section 2.2.1). Pour chaque recette, nous obtenons donc un score de confiance pour chaque type de plat qui pourra être utilisé pour choisir la classe à associer à une recette.

2.3 SVMs

Le corpus des recettes $\mathbb{X}$ étant donné, il s'agit d'élaborer à partir d'un sous-ensemble $\mathbb{D} \subset \mathbb{X}$ de recettes annotées par classe (niveau de difficulté ou type de plat selon la tâche), une méthode de

2. http ://lia.univ-avignon.fr/fileadmin/documents/Users/Intranet/chercheurs/bechet/download_fred.html

classification γ qui à toute recette associe une classe. En notant $\mathbb{C}$ l'ensemble des classes, nous avons donc :

$$\gamma : \mathbb{X} \longrightarrow \mathbb{C} \tag{1}$$

La troisième des méthodes appliquées a consisté, pour chaque couple de classes $(c, c') \in \mathbb{C} \times \mathbb{C}$ (avec $c \neq c'$), à apprendre et à appliquer un classifieur binaire $\gamma_{(c,c')} : \mathbb{X} \rightarrow \{c, c'\}$ à base de SVMs.

En notant $s_{(c,c')}(r)$ le score obtenu selon le classifieur $\gamma_{(c,c')}$ par la recette $r \in \mathbb{X}$, nous avons alors, avec $c' \in \mathbb{C}$:

$$\gamma(r) = \arg\max_{c \in \mathbb{C}} \left(s_{(c,c')}(r) \right) \tag{2}$$

Lors des phases d'apprentissage et d'application des classifieurs binaires, les recettes ont été représentées vectoriellement dans l'espace du lexique de référence. Le poids $w(t)$ du terme t d'une recette r dans le vecteur associé est donné par :

$$w(t) = tf(t).idf(t) = tf(t).\log\left(\frac{|\mathbb{X}|}{df(t)}\right) \tag{3}$$

où $tf(t)$ désigne le nombre d'occurrences du terme t dans la recette et $df(t)$ le nombre de recettes du corpus $\mathbb{X}$ où le terme t apparaît. L'approche décrite a été appliquée pour les deux tâches de classification.

Dans le cas de la prédiction du type de plat, le lexique a cependant été filtré avant vectorisation des recettes sur le critère de l'information mutuelle entre un terme t et une classe $c \in \mathbb{C}$ (Manning et al., 2008). Après sélection des termes, seuls les 10 000 termes les plus porteurs d'information sur les classes ont alors été retenus.

2.4 Similarité cosinus

La quatrième approche, seulement utilisée pour la prédiction du type de plat, repose elle aussi sur une représentation vectorielle des documents : à chaque recette r, nous faisons correspondre un vecteur v_r dont les composantes $w_r(t)$ dans l'espace du lexique sont données, pour chaque terme t présent dans le corps de la recette, par :

$$\begin{aligned} w_r(t) &= tf(t).idf(t).G(t) \\ &= tf(t).idf(t).\sum_{c \in \mathbb{C}} \mathbb{P}^2(c|t) \\ &= tf(t).idf(t).\sum_{c \in \mathbb{C}} \left(\frac{df_c(t)}{df_{\mathbb{T}}(t)}\right)^2 \end{aligned} \tag{4}$$

où $G(t)$ désigne l'indice de *Gini*, $df_{\mathbb{T}}(t)$ est le nombre de recettes du corpus d'apprentissage $\mathbb{T} \subset \mathbb{X}$ contenant le terme t et $df_c(t)$ correspond au nombre de recettes du corpus d'apprentissage annotées selon le type $c \in \mathbb{C}$.

A chaque classe $c \in \mathbb{C}$, nous faisons par ailleurs correspondre un vecteur dont les composantes $w_c(t)$ sont données dans l'espace du lexique par :

$$w_c(t) = df_c(t).idf(t).G(t) \tag{5}$$

où $df_c(t)$ désigne le nombre de recettes annotées selon le type de plat c où le terme t apparaît.

Une mesure de similarité $s(r,c)$ entre une recette $r \in \mathbb{X}$ et un type de plat $c \in \mathbb{C}$ est alors donnée par le cosinus de l'angle formé par les vecteurs v_r et v_c :

$$s(r,c) = \cos(\widehat{v_r, v_c}) = \frac{\sum_{t \in r \cap c} w_r(t).w_c(t)}{\sqrt{\sum_t w_r(t)^2.w_c(t)^2}} \tag{6}$$

Dans ces conditions la classe $\gamma(c)$ attribuée à une recette r est celle qui maximise la mesure de similarité cosine :

$$\gamma(r) = \arg\max_{c \in \mathbb{C}} (s(r,c)) \tag{7}$$

Le vocabulaire utilisé lors de la phase de vectorisation était constitué des seuls termes t dont l'indice de Gini $G(t)$ était au moins égal à 0.45.

Ce seuil a été fixé empiriquement par maximisation du macro F-score sur un corpus de développement constitué de 3 864 des 13 864 recettes du corpus d'apprentissage.

2.5 Fusion des systèmes

Pour chacune des tâches de classification, plusieurs méthodes ont donc été appliquées : trois pour la tâche 1, quatre pour la tâche 2.

Pour chaque couple $(r,c) \in \mathbb{X} \times \mathbb{C}$ associant une recette du corpus à une classe, nous disposions donc d'un score $s_i(r,c)$ pour chacune des méthodes de classification (avec $i = 1, ..., 3$ ou $i = 1, ..., 4$ selon la tâche).

2.5.1 Normalisation des résultats par méthode

Les scores produits en sortie des différents systèmes de classification ont d'abord été normalisés de manière à ce que la somme des prédictions sur l'ensemble des classes pour une méthode donnée soit égale à 1. En notant $n_i(r,c)$ le score normalisé obtenu selon la $i-$ème méthode pour le couple $(r,c) \in \mathbb{X} \times \mathbb{C}$, nous avons donc :

$$n_i(r,c) = \frac{s_i(r,c)}{\sum_{j=1}^{|\mathbb{C}|} s_i(r,c_j)} \tag{8}$$

où c_j désigne la $j-$ème classe ($j = 1, ..., 4$ ou $j = 1, ...3$ selon la tâche de classification).

Deux méthodes de fusion ont été appliquées après normalisation des scores $s_i(r,c)$.

2.5.2 Combinaison linéaire des scores

La première méthode de fusion a consisté, pour une recette donnée r, à départager les classes candidates par combinaison linéaire des scores obtenus selon les différentes méthodes de classifi-

cation. En notant $\gamma(r)$ la classe conjecturée, on a donc :

$$\gamma(r) = \arg\max_{c \in \mathbb{C}} \left(\sum_{i=1}^{m} n_i(r,c) \right) \tag{9}$$

où m correspond au nombre de méthodes appliquées pour la tâche considérée.

2.5.3 Méthode ELECTRE

Issue du domaine de l'optimisation multi-objectif discrète, la méthode ELECTRE (Roy, 1991) consiste à définir sur l'ensemble $\mathbb{C}$ des classes candidates une relation $\mathscr{S} \subset \mathbb{C} \times \mathbb{C}$ dite de surclassement. De manière informelle, nous pouvons dire qu'une classe c surclasse une classe c' si elle la domine sur un nombre « important » de méthodes et si, sur les éventuelles méthodes restantes où elle est dominée par c', elle ne l'est pas au-delà d'un certain seuil fixé pour chaque méthode. Plus formellement :

$$c \mathscr{S} c' \Longleftrightarrow \begin{cases} conc(c,c') \geqslant sc \\ v(c,c') = 0 \end{cases} \tag{10}$$

où $conc(c,c') \in [0,1]$ désigne l'indice de concordance entre les classes candidates c et c', $sc \in [0,1]$ un seuil dit de concordance fixé empiriquement et $v(c,c') \in \{0,1\}$ un indice binaire dit *de veto*.

L'indice de concordance $conc(c,c')$ évalue le taux de méthodes (éventuellement pondérées) selon lesquelles c domine c' :

$$conc(c,c') = \frac{\sum_{i \in M(c,c')} p_i}{\sum_{i \in M} p_i} \tag{11}$$

où M désigne l'ensemble des méthodes, $M(c,c')$ l'ensemble des méthodes sur lesquelles c domine (non strictement) c' et p_i le poids accordé à chaque méthode $i \in M$.

L'indice binaire $v(c,c')$ (à valeurs dans l'ensemble $\{0,1\}$) fixe la valeur du veto de c' vers c selon les modalités suivantes (en notant $n_i(r,c)$ le score obtenu selon la i-ème méthode par le couple (r,c) associant une recette à une classe) :

$$v(c,c') = \begin{cases} 1 \Longleftrightarrow \exists i \in M : n_i(r,c') > n_i(r,c) \text{ et } n_i(r,c') - n_i(r,c) \geqslant v_i \\ 0 \text{ sinon} \end{cases} \tag{12}$$

où $v(i) \in [0,1]$ est la valeur de veto associée à la $i-$ème méthode.

Les valeurs des différents paramètres ont été fixées empiriquement à partir du corpus de développement par maximisation des métriques appliquées pour évaluer la classification.

Les poids p_i associés aux différentes méthodes ont tous été fixés à $p_i = 1$. Le seuil de concordance a été fixé à $sc = 0,7$ pour la tâche 1 et $sc = 0,6$ pour la tâche 2 ; enfin, la valeur de veto v_i a été fixée à $v_i = 0,5$ pour chacune des méthodes et quelle que soit la tâche considérée.

Le noyau de la relation $\mathscr{S} \subset \mathbb{C} \times \mathbb{C}$ est alors constitué des éventuelles classes candidates non surclassées.

Dans le cas d'un noyau *singleton*, la classe conjecturée pour la recette est l'unique classe élément du noyau.

Dans le cas de noyau vide ou de noyau formé de plusieurs classes concurrentes, c'est la meilleure classe candidate selon la méthode de combinaison linéaire qui a été retenue.

2.6 Extraction d'ingrédients

La tâche 4 est une tâche pilote d'extraction d'information. La difficulté est multiple, car les auteurs des recettes n'utilisent pas tous les ingrédients, ou ils ont introduit librement des ingrédients non listés. Par exemple, il n'est pas rare d'avoir des passages comme « mélanger énergiquement tous les ingrédients » (recette 27174), qui ne permettent pas d'extraire directement les ingrédients.

Nous avons attaqué ce problème en utilisant deux approches : l'une à base de règles et l'autre probabiliste. L'idée de base pour les règles a été la suivante : soit A l'ensemble de mots de la recette i ; soit B l'ensemble de mots de la liste DEFT (références d'apprentissage ou *gold-standard* d'évaluation). Éliminer tous les mots de A qui ne sont pas dans B. Cela produit une liste d'ingrédients candidats L_c.

Cette liste peut contenir des termes génériques comme « VIANDE », « FROMAGE » ou « POISSON », présents dans le texte de la recette. Afin de mieux identifier ce terme, nous avons employé une approche probabiliste naïve. Nous avons calculé la probabilité d'avoir un certain type de viande x, étant donné une liste d'ingrédients $L_c = (L_1, L_2, ..., L_n)$:

$$p(x|L_c) = P(x \cap L_c)/p(L_c); \quad x = \{\text{VIANDE, FROMAGE, POISSON}\} \tag{13}$$

Les termes ainsi identifiés, sont alors injectés dans la liste extraite L_c, ce qui produit la liste deifnitive L.

La liste d'ingrédients L ainsi obtenue a été utilisée dans les systèmes de *boosting* (c.f. section2.2) dans les tâches T1 et T2.

3 Résultats et discussion

Les systèmes ont été évalués en utilisant les mesures proposées par DEFT : le F-score pour les tâches 1 et 2, et la mesure *Mean Average Precision* (MAP) de TREC pour la tâche d'extraction.

3.1 Tâches de classification

Pour la tâche de classification en niveau de difficulté (T1), nos trois runs étaient donnés par :

– Run 1 : SVMs seuls ;
– Run 2 : fusion des trois méthodes par ELECTRE ;
– Run 3 : combinaison linéaire des trois méthodes.

Pour la tâche de classification en type de plat (T2) :

– Run 1 : Modèle discriminant seul ;

– Run 2 : fusion des quatre méthodes par ELECTRE ;
– Run 3 : combinaison linéaire des quatre méthodes.

Les résultats obtenus sont récapitulés dans les tableaux 1 (tâche 1) et 2 (tâche 2).

Pour la tâche 1 et pour chacun des trois runs, nous fournissons, outre les F-scores, la distance moyenne (micro-écart) de l'hypothèse à la référence sur l'échelle des quatre niveaux de difficulté, deux niveaux contigüs étant distants de 1.

T1	**run 1**	**run 2**	**run 3**
Micro F-score	**0.5916**	0.5812	0.5877
Macro F-score	0.4531	0.4531	**0.4569**
Distance moyenne	**0.4409**	0.4586	0.4465

TABLE 1 – *Résultats obtenus sur le corpus de test - Tâche 1*

T2	**run 1**	**run 2**	**run 3**
Micro F-score	0.8760	0.8860	**0.8886**
Macro F-score	0.8706	0.8795	**0.8824**

TABLE 2 – *Résultats obtenus sur le corpus de test - Tâche 2*

Pour la tâche 2, la combinaison des méthodes permet d'obtenir des scores systématiquement supérieurs à ceux que nous obtenons en appliquant une méthode isolée.

Pour la tâche d'évaluation du niveau de difficulté, les bénéfices retirés de l'utilisation d'une méthode de fusion sont moins nets et apparaissent dépendants de la métrique appliquée lors de l'évaluation.

Ceci a permis au LIA de se positionner dans la tâche 1 : 3ème/6 et dans la tâche 2 : 1er/5.

En ce qui concerne la tâche T2, nous croyons que l'évaluation aurait dû prendre en compte la dimension muti-labels de plusieurs recettes. Par exemple la recette 18 052 (présente dans le test) se termine par « servir tiède (aussi bon chaud que froid ou réchauffé), en entrée ou en plat » est annotée uniquement comme une entrée. Si un système produit l'étiquette *plat principal*, faut-il pour autant compter cela comme une erreur ?

Relevons aussi l'aspect subjectif de la tâche T1. En effet, dire qu'une recette est facile ou difficile sans tenir compte qui en est l'auteur c'est faire fi de son niveau d'expertise ! ! !

Nous émettons également l'idée que les macro-mesures devraient être privilégiées aux micro-mesures car ces dernières pousseraient à mettre en œuvre des stratégies négligeant les classes minoritaires. Nous pensons donc que cette façon d'évaluer serait plus adaptée à une utilisation orientée application.

3.2 Tâche d'extraction

L'évaluation de cette tâche est assez subjective, malgré son caractère d'extraction d'information. Par exemple, dans l'ensemble d'apprentissage, la recette 10 514 dit : « 40 cl d'Apremont ou autre blanc de Savoie sec (facultatif, mais donne plus de goût) ». Les références DEFT indiquent **mais** comme ingrédient de la tartiflette. Evidemment nous n'avons pas incorporé l'information des ingrédients d'apprentissage, à la différence de la méthode DEFT ayant généré le *gold-standard*.

Nous avons obtenu un score de MAP=0.6364 en mesure *qrel* dans le corpus d'apprentissage, et dans le classement officiel un score de MAP=0.6287 dans le corpus de test. Cela a positionné l'équipe du LIA au rang 3ème/5. Bien que standard, la mesure MAP pourrait avoir intégré les accents dans l'évaluation des ingrédients extraits : cela aurait permis de lever des ambiguïtés qui ont été inutilement introduites par la désaccentuation.

Nous pensons que les références fournies par DEFT dans l'ensemble d'apprentissage, ont contribué à rendre la tâche floue. En effet, cela n'a pas de sens d'extraire *papier sulfurisé* ou *couteau* (extraite à partir de *pointe de couteau*) comme ingrédients. De même, l'ingrédient *maïs*, incorrectement extrait des recettes d'apprentissage comportant l'expression *mais...*, n'est pas apparu dans les recettes de test comportant ladite expression.

3.3 Comparaison versus les humains

Nous avons mis en place une petite expérience impliquant des êtres humains. Pour la tâche T4, nous avons demandé à 7 annotateurs (experts ou non en matière culinaire), d'effectuer la tâche DEFT. Ceci nous a permis de positionner nos systèmes par rapport aux performances des personnes. Puisque les personnes ont des connaissances extra-linguistiques, les juges humains ont eu comme seule consigne celle de ne pas consulter des ressources externes (par exemple des ressources électroniques ou des livres de cuisine) pour classer les recettes ou pour extraire les ingrédients. Le tableau 3 montre la moyenne MAP pour chaque annotateur A_i sur 50 recettes de la tâche T4, choisies au hasard. La moyenne générale pour les personnes est de MAP=0.5433 et pour le système S de MAP=**0.6570**. Ceci montre que dans ce sous-ensemble, nos systèmes sont au-dessus des performances atteintes par les humains.

A1	A2	A3	A4	A5	A6	A7	S
0.5333	0.6029	0.5271	0.5880	0.5313	0.4679	0.5529	**0.6570**

TABLE 3 – Performance des personnes sur la tâche T4.

Il est clair que, même les humains ayant une expertise culinaire, ont obtenu de piètres notes dans cette tâche. Cela s'explique par la mauvaise qualité du *gold standard* fourni par DEFT. Par exemple, la crème ou le café (cuillère à café, recette 90 806) semblent subir une extraction de nature aléatoire... D'ailleurs, nous avons souvent été surpris par les ingrédients de référence de DEFT utilisés dans telle ou telle recette, qui ne peuvent en aucun cas avoir été employés de façon conjointe. Par exemple, dans la recette 64 761 (dessert), la référence DEFT liste parmi les ingrédients « moules » et « caramel » ! Évidemment il s'agit de l'ustensile *moule à charlotte*. Malheureusement il s'agit de problèmes récurrents. Sans parler de l'eau ou de la pâte (déclinée comme pâté, pâtes fraîches, brisée, sablée, etc.) qui semblent être détectées aléatoirement. Pour répondre au défi, nous avons développé plusieurs systèmes d'extraction d'ingrédients qui ont dû

être modifiés (souvent à l'encontre de ce qui nous semblait logique) afin de rendre une sortie proche de celle des références. Ceci revient en fin de compte à modéliser la machine d'extraction de DEFT. Les références étant de qualité discutable, est-il intéressant de chercher à reproduire la sortie d'une telle machine ?

4 Conclusions

Malgré la quantité d'erreurs présentes dans le corpus d'apprentissage, nos algorithmes ont conduit à des résultats intéressants. Nous voulons ajouter quelques commentaires par rapport à ce défi. Nous avons observé que l'auto-évaluation du niveau de difficulté des recettes par ceux qui les publient est un exercice très subjectif, assez souvent sujet à caution. En ce qui concerne le type de plat, l'évolution du mode de vie rend assez floue la distinction entre plat principal et hors d'œuvre. Les tartes salées, les quiches ou les tartines sont de plus en plus considérées comme des plats « de résistance ». Ceci conduit souvent les internautes à exprimer de façon explicite leur indécision quant à la catégorie à retenir. Ces observations n'ont pas suffi à atténuer notre perplexité lorsque nous avons constaté que certaines méthodes à base de « cuisine » algorithmique réussissaient à faire mieux que des approches sophistiquées. Pour autant, nous n'avons pas cédé à la tentation de les retenir pour en faire une soumission !

Remerciements

Nous remercions les annotateurs qui ont bien voulu nous aider dans cette tâche ! Patricia Velázquez, Sulan Wong, Mariana Tello, Alejandro Molina et Grégoire Moreau. Nous tenons aussi à remercier les organisateurs de la campagne d'évaluation, sans qui nous n'aurions pu participer à ce défi.

Références

Favre, B., Hakkani-Tür, D., and Cuendet, S. (2007). Icsiboost. `http://code.google.com/p/icsiboost`.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Roy, B. (1991). The outranking approach and the foundations of ELECTRE methods. *Theory and Decision*, 31 :49–73.

Schapire, R. E. (2003). The Boosting Approach to Machine Learning : An Overview. In *Nonlinear Estimation and Classification*.

Schapire, R. E. and Singer, Y. (2000). BoosTexter : A boosting-based system for text categorization. In *Machine Learning*, volume 39, pages 135–168.

Torres-Moreno, J., El-Bèze, M., Bellot, P., and F., B. (2011). *Peut-on voir la détection d'opinions comme un problème de classification thématique ?*, chapter 9. Hermes Lavoisier.

Approches hybrides pour l'analyse de recettes de cuisine DEFT, TALN-RECITAL 2013

Luca Dini, André Bittar, Mathieu Ruhlmann
Celi France, 12-14 rue Claude Génin, 38000 Grenoble
dini@celi-france.com, bittar@celi-france.com, ruhlmann@celi-france.com

RESUME

Le Défi fouille de textes (DEFT) 2013 porte sur l'analyse automatique de recettes de cuisine en langue française, une thématique actuellement à la mode et qui a déjà fait l'objet d'une campagne d'évaluation (Computer Cooking Contest, 2012). Dans cet article, nous présentons nos modules pour le traitement de la tâche 1 (la détection du niveau de difficulté d'une recette) et de la tâche 4 (l'extraction des ingrédients d'une recette). Notre système pour la tâche 1 repose sur une combinaison de méthodes d'apprentissage et de logique probabiliste, alors que pour la tâche 4, nous avons utilisé des méthodes hybrides (symboliques et statistiques). Nous posons certaines questions concernant le traitement automatique d'un corpus qui contient des données subjectives, ainsi que la pertinence de la mesure MAP pour l'évaluation de la tâche 4.

ABSTRACT

The Défi fouille de textes (DEFT) 2013 focuses on the automatic processing of cooking recipes in French, a topic that has already been the subject of an evaluation campaign (Computer Cooking Contest, 2012). In this article, we present our systems for the processing of tasks 1 (identifying the level of difficulty of a recipe) and 4 (extracting ingredients from a recipe). Our system for task 1 is based on a combination of machine learning and probabilistic logic, while for task 4, we used hybrid (symbolic and statistical) methods. We raise certain questions concerning the automatic processing of a corpus containing subjective data, as well as the appropriateness of using the MAP measure for the evaluation of task 4.

MOTS-CLES : anaylse de recettes, extraction d'information, classification automatique, méthodes hybrides
KEYWORDS : recipe analysis, information extraction, automatic classification, hybrid methods

1 Introduction

Le Défi Fouille de Textes (DEFT) 2013 porte sur l'analyse automatique de recettes de cuisine en langue française. Cette thématique, actuellement à la mode, a déjà fait l'objet d'une campagne d'évaluation, sous forme du « Computer Cooking Contest » (CCC) (Computer Cooking Contest, 2012) tenu tous les ans depuis 2008. Le but de l'édition 2012 de la campagne CCC était d'encourager l'utilisation de moyens technologiques pour aider à la création de recettes de cuisine. Il s'agissait d'une tâche de *génération* de recettes.

En revanche, cette édition du DEFT consiste en quatre tâches d'*analyse* de recettes de cuisine :

1. 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.
2. Identifier à partir du titre et du texte de la recette le type de plat préparé : entrée, plat principal, dessert.
3. Apparier le texte d'une recette à son titre.
4. Extraire du titre et du texte d'une recette la liste de ses ingrédients.

Nous présentons, dans cet article, notre système d'analyse linguistique HOLMES, et nos travaux pour le traitement des tâches 1 et 4.

2 HOLMES

HOLMES (Hybrid Operable platform for Language Management and Extensible Semantics) est une plateforme pour l'analyse de textes écrits en français, italien et anglais, développée au sein de Celi France. C'est avec certains modules de ce système que nous avons abordé les tâches 1 et 4 du DEFT 2013.

HOLMES permet de faire les traitements TAL de base (détection de phrases, tokenisation, étiquetage morphosyntaxique) ainsi que des traitements de plus haut niveau, tels que l'analyse syntaxique en dépendances, l'identification d'entités nommées et l'extraction d'information. La plateforme permet également l'entraînement de classifieurs pour l'apprentissage. HOLMES est développé entièrement en Java et a été conçu pour maximiser son extensibilité pour des vraies tâches de TAL.

Techniquement, HOLMES est une plateforme pour le traitement automatique des langues basée sur une approche *radicalement incrémentale*. Cela veut dire que toutes les informations ajoutées par le système restent accessibles à travers les différentes étapes de traitement. Actuellement, les principaux traitements disponibles dans HOLMES sont les suivants :

- Segmentation en phrases
- Tokenisation
- Etiquetage morphosyntaxique
- Analyse morphologique
- Règles de patrons linéaires sur des séquences d'objets linguistiques
- Analyse syntaxique en dépendances
- Interrogation de base ontologique
- Extraction d'entités nommées (avec Conditional Random Fields)
- Classification automatique

Le module de règles sur des séquences d'objets linguistiques, que nous avons utilisé pour la tâche 4, s'inspire du module TokensRegexp (CHANG, 2011) du package CoreNLP développé à l'Université de Stanford (STANFORD, 2010). Ces règles permettent l'application d'annotations sur des séquences d'objets linguistiques (ex. tokens). Dans ces règles, tous les traits linguistiques d'un objet donné peuvent être utilisés, ce qui résulte en un système de règles assez puissant. La figure Figure 1: exemple de règle sur séquence d'objets linguistiques. donne un exemple d'une telle règle. Cette règle s'applique sur une séquence commençant par un token ayant le lemme « filet », suivi du token « de » ou « d' », suivi d'un token étiqueté comme nom commun ou nom propre. La règle applique l'annotation

« INGREDIENT » sur la séquence rencontrée dans un texte.

```
{ pattern : ( [{lemma :filet}] /de|d'/ [{tag:/NC|NPP/}] ),
  action : ( Annotate($0,"ner","INGREDIENT") ) }
```

Figure 1: exemple de règle sur séquence d'objets linguistiques.

La nature hybride de HOLMES est apparente dans l'intégration étroite de méthodes symboliques (des règles linguistiques écrites à la main par des experts) avec des méthodes basées sur apprentissage. En effet, la sortie des modules statistiques est accessible au niveau des modules symboliques, et inversement, les modules statistiques peuvent recevoir comme traits d'entrée les résultats des calculs symboliques.

3 Tâche 1 : détecter le niveau de difficulté d'une recette

3.1 Description de la tâche 1

La tâche 1 du DEFT 2013 consiste à déterminer, pour une recette de cuisine donnée en entrée, son niveau de difficulté sur une échelle à 4 niveaux : très facile, facile, moyennement difficile, difficile. Pour cette tâche, un système peut se baser sur toutes les informations qu'il est possible d'extraire à partir du texte de la recette et son titre. La mesure d'évaluation pour cette tâche est la distance entre la réponse du système et la réponse correcte.

Malgré sa simplicité apparente, cette tâche s'avère très intéressante dans la mesure où elle implique un certain type de classification, notamment une classification *fonctionnelle,* tout comme la classification en genre (KARLGREN & CUTTING, 1994) (KESSLER, NUMBERG, & SCHÜTZE, 1997). Dans une tâche de classification thématique « standard », les mots du texte caractérisent son appartenance à une classe ou une autre, ce qui se prête à une approche de classification par « sac de mots ». En revanche, pour une tâche de classification fonctionnelle, ce genre de corrélation n'est pas toujours utile car les critères de bonne classification se trouvent dans les *relations* entre mots. Par exemple, deux recettes qui décrivent le même dessert avec deux niveaux de difficulté différents sont thématiquement plus proches qu'une recette de plat principal et une recette de dessert ayant le même degré de difficulté.

Pour cette tâche, un corpus d'apprentissage a été fourni aux participants. Ce corpus contient un total de 13 865 recettes provenant du site web Marmiton.org et qui sont associées à leur niveau de difficulté. Pour chaque recette, l'appréciation de sa difficulté est donnée par son auteur.

3.2 Approche : classification à deux niveaux

L'approche que nous avons adoptée pour aborder cette tâche est basée sur deux niveaux de classification. Le premier niveau repose sur l'entraînement d'un classifieur basé purement sur le corpus d'apprentissage. Le deuxième niveau consiste en l'application de formules logiques qui confirment ou contredisent les résultats de la première étape de classification, permettant de les réviser. Les deux classifieurs utilisent le même ensemble de traits. Etant donnée la nature fonctionnelle de cette tâche, la sélection des traits, que nous présentons dans la section 3.2.1, a été primordiale.

3.2.1 Sélection de traits

Pour cette tâche, la sélection de traits doit être guidée par la question « qu'est-ce qui rend une recette difficile ? ». Intuitivement, on pourrait considérer la longueur du texte comme un facteur décisif, un postulat basé sur la supposition implicite que plus une recette est longue, plus le nombre d'instructions nécessaire à son accomplissement est important. Si cette supposition peut s'avérer vraie pour un livre de recettes « normalisées », ce n'est pas le cas pour un recueil de recettes générées par une communauté d'internautes. En effet, un auteur peut ajouter à une recette courte, une longue explication de ses origines, des souvenirs personnels concernant sa découverte, des faits intéressants sur les ingrédients, etc., comme c'est le cas du corpus de cette tâche. Pour un corpus contenant de telles recettes, la simple longueur du texte n'est pas un critère approprié et nous l'avons donc exclu come trait dans notre traitement.

Une autre approche serait d'adopter comme trait le nombre d'occurrences de verbes trouvés dans le texte. Typiquement, un verbe dénote une action, et donc une opération à effectuer dans la recette. Plus il y a de verbes, plus il y a d'actions à effectuer, et plus la recette est compliquée et difficile. Cependant, nos expérimentations avec cette approche n'ont donné que des résultats moyens.

Une approche plus raisonnée serait alors d'admettre que, dans le corps de recettes publiées par une communauté d'internautes, peuvent apparaître des verbes qui dénotent des actions liées au domaine de la cuisine, mais aussi des verbes qui sont en dehors de ce champ sémantique et qui ne doivent pas être pris en compte. Dans le domaine des « prédicats de cuisine » on peut faire la distinction entre des prédicats qui dénotent des actions intrinsèquement faciles, telles que « faire chauffer » et « faire bouillir », et des prédicats qui dénotent des actions qui demandent plus d'expérience, telles que « flamber ». Bien que cette distinction ne soit pas pertinente pour un classifieur sac-de-mots (qui apprend de façon autonome les prédicats qui caractérisent les recettes faciles et difficiles), elle peut être très importante pour un classifieur manuellement paramétré. Nous avons donc opté pour cette approche.

Afin de différencier les prédicats « faciles » des prédicats « difficiles » et « neutres » nous avons compté le nombre d'occurrences de chacun des verbes dans les recettes des différents niveaux de difficulté. Un prédicat appartient à la classe pour laquelle il a le plus grand nombre d'occurrences. Nous aurions pu adopter des méthodes plus fines, telles que « log odds ratio » (EVERITT, 1992), pour établir ce genre de correspondance, mais pour des raisons de simplicité de mise-en-œuvre, nous avons opté pour l'approche que nous venons de décrire. Avec cette méthode, nous avons produit une liste de prédicats associés à un seul niveau de difficulté.

La seule détection des prédicats de cuisine est susceptible, dans certains cas, de produire des erreurs dans les cas où le prédicat est modifié par un syntagme prépositionnel ou adverbial. Par exemple, les instructions « mélangez le tout », « mélangez longuement le tout » et « mélangez longuement le tout avec attention » dénotent des actions avec, dans l'ordre, un degré de complexité croissant. Dans notre calcul de traits pour apprentissage, nous supposons que plus un verbe a de modifieurs, plus l'action qu'il dénote est complexe. Bien entendu, un modifieur peut n'impliquer aucune augmentation dans la complexité d'une action (ex. « déposez le mélange **dans un bol** »), mais une vérification manuelle a montré que

cela n'est pas très fréquent dans le corpus utilisé pour cette tâche. Nous avons repéré les modifieurs de verbes à l'aide d'une analyse syntaxique en dépendances et en extrayant la tête syntaxique de syntagmes gouvernés par un prédicat donné.

Une recette peut parfois contenir des assertions sur sa difficulté. Il s'agit, dans ce cas, d'une information qui relève du domaine de l'analyse d'opinion plutôt que de la classification pure et, en tant que telle, nécessiterait un traitement indépendant. Cependant, nous n'avons pas disposé de suffisamment de temps pour mettre en place un système pour analyser les assertions subjectives sur la difficulté des recettes. Nous avons alors simplement étiqueté tous les mots selon leur appartenance à un domaine de difficulté ou son opposé. Nous avons constitué un lexique de ces mots qui dénotent une polarité facile ou difficile. Au moment de l'exécution, les adjectifs de cette liste sont convertis en leur équivalent adverbial.

Pour résumer, les traits que nous avons utilisés pour entraîner notre classifieur sont les suivants :

- Classes de verbes selon le niveau de difficulté d'une action liée à la cuisine
- Nombre de modifieurs des verbes
- Classes de mots indiquant une polarité facile ou difficile

3.2.2 Classifieur simple

La première étape de classification consiste en l'entraînement d'un classifieur d'entropie maximale (MaxEnt) sur l'ensemble de traits mentionné dans la section 3.2.1. Afin d'éviter que le classifieur ne s'oriente vers la tâche de classification thématique (à la place d'une classification fonctionnelle), nous n'avons pas fourni en entrée le texte entier. Pour la même raison, nous n'avons pas non plus fourni la liste des ingrédients détectés, qui était pourtant rendue disponible par la tâche 4. La liste de prédicats « faciles » et « difficiles » repérés dans le texte, ainsi que la liste de mots renvoyant à une polarité facile ou difficile détectés, ont été fournies en entrée au classifieur. Dans cette configuration, nous avons obtenu une « précision »[1] d'environ 67%.

3.2.3 Réseaux logiques de Markov

Le deuxième niveau de classification que nous avons effectué consiste en l'application de réseaux logiques de Markov (RICHARDSON, 2006). Un réseau logique de Markov (RLM) est une logique probabiliste qui permet la modélisation de problèmes qui peuvent être décrits en termes de prédicats et formules logiques, et qui permet de faire des inférences incertaines (BEEDKAR, DEL CORRO, & GEMULLA, 2013). Les RLM ont été utilisés pour obtenir des résultats satisfaisants dans d'autres tâches de TAL, tels que la désambiguïsation sémantique et l'étiquetage de rôles sémantiques (CHE & TING, 2010) (RIEDEL & MEZA-RUIZ, 2008) ou l'analyse sémantique non supervisée (POON & DOMINGOS, 2009). Nous avons utilisé l'implémentation des RLM fourni dans le package open-source Tuffy (NIU, RE, DOAN, & SHAVLIK, 2011). Une des motivations pour avoir choisi d'utiliser les RLM réside dans le fait que les packages tels que Tuffy permettent d'apprendre automatiquement, à partir d'un corpus de référence, les poids à attribuer aux prédicats. Par exemple, prenons les deux

[1] Nous utilisons la formule simple suivante pour calculer la « précision » : *éléments correctement détectés / éléments corrects*. Il ne s'agit donc pas de la mesure de précision habituellement utilisée en extraction d'information.

formules suivantes :

1. Recipe(x) & Contains(x, butter) & Has_difficult_preds(x, y) && y > 10 → DIFF(x)
2. Recipe(x) & Contains(x, butter) & Has_difficult_preds(x, y) && y < 10 → EASY(x)

Le système calcule les probabilités de 1 et 2 sur un corpus de référence contenant les prédicats cibles DIFF(x) et EASY(x). L'intérêt de la pondération de prédicats vient du fait qu'un système, suivant l'approche de (RIEDELL, 2008), peut générer un ensemble très important de prédicats possibles et décider lesquels mèneraient à une bonne conclusion.

Malheureusement, étant donné le corpus de référence à notre disposition, notre méthode de détection automatique du niveau de difficulté a donné des résultats médiocres. En effet, si nous avons obtenu quelques résultats prometteurs (40% de précision pour une échelle de difficulté à 4 niveaux, et 60% pour une échelle à 2 niveaux de difficulté), une inspection manuelle des poids attribués aux prédicats a montré que l'algorithme a résulté en un surapprentissage, sans montrer de réelle connexion avec l'appréciation humaine de la difficulté des recettes.

Après cette phase d'expérimentation, nous avons alors décidé d'omettre la pondération automatique sur la base du corpus de référence pour favoriser des règles écrites à la main. Nous avons écrit des règles pour distinguer les recettes sur une échelle de difficulté bipartite – facile et difficile – la distinction sur une échelle à 4 niveaux étant en dehors de la portée de notre système. Nous donnons ci-dessous quelques exemples des types de règles que nous avons écrites. Pour une recette r donnée :

1. Si r contient plus de x prédicats « difficiles » et plus de y modifieurs → DIFF(r)
2. Si r contient plus de x prédicats et contient le mot « long » ou contient le mot « difficile » → DIFF(r)
3. Si le ratio de mots à polarité « facile » aux mots de polarité « difficile » > y → EASY(r)
4. Si le ratio de mots à polarité « difficile » aux mots de polarité « facile » > y → DIFF(x)
5. Si le nombre d'ingrédients identifiés > x et le nombre de prédicats > y → DIFF(x)
6. Si l'ingrédient i est détecté et que r contient au moins un mot à polarité « difficile » → DIFF(x)

L'application de ces règles a encore entraîné une légère baisse de précision. Mais encore, un examen manuel des résultats a montré que ces erreurs étaient souvent dues à des imprécisions dans le corpus de référence où une fausse appréciation avait été attribuée. Sans prendre en compte ces erreurs, les règles manuelles donnaient une nette amélioration des résultats.

3.2.4 Approche hybride

L'approche finale que nous avons adoptée pour cette tâche se compose de deux parties :

1. Un classifieur d'entropie maximale (MaxEnt) entraîné sur des prédicats pertinents et des mots renvoyant aux 4 niveaux de difficulté
2. Un classifieur RLM (à base de règles) capable de distinguer uniquement entre des recettes faciles et difficiles

Comme la tâche demande une classification sur 4 niveaux de difficulté, nous avons gardé le

classifieur MaxEnt comme décideur principal et nous sommes servis du classifieur RLM pour effectuer un ré-ranking des résultats. Dans le cas où le classifieur RLM attribuait la catégorie « facile », les scores du classifieur MaxEnt pour les catégories « facile » et « très facile » se voyaient augmenter. Inversement, si le classifieur RLM attribuait la classe « difficile », les scores MaxEnt pour « difficile » et « très difficile » étaient revus à la hausse.

L'amélioration apportée par l'utilisation du classifieur RLM était minime, de l'ordre de 1% de précision (une augmentation de 67% à 68%). Cependant, après une inspection manuelle des résultats, nous avons décidé de garder ce module dans le système car les erreurs étaient dues, pour la plupart, à une « mauvaise » évaluations de la part de l'auteur de la recette dans le corpus de référence.

3.3 Discussion: mes recettes ne sont jamais difficiles!

Sur l'ensemble du corpus, nous avons constaté que l'ensemble des recettes « difficiles » et « moyennement difficiles » ne constitue qu'environ un dixième (1 148) des recettes des classes « facile » et « très facile » (12 714). Cette différence émerge-t-elle simplement du fait d'une prévalence « objective » de recettes faciles, ou relève-t-elle plutôt d'un effet des intentions des auteurs ? Nous tentons, par la suite, de répondre à cette question.

Nous estimons que le concept de la difficulté d'une recette de cuisine peut se définir de deux façons. Dans le cadre d'une publication contrôlée, il existe des critères définis par l'équipe éditoriale pour déterminer la difficulté. Dans ce cas-là, une certaine homogénéité est imposée aux auteurs. Cependant, dans le cas de recettes éditées par des utilisateurs, l'évaluation de la difficulté se base sur trois critères :

1. Les capacités culinaires de l'auteur
2. L'intention de l'auteur : souhaite-t-il que sa recette devienne populaire auprès du grand public ou plutôt qu'elle reste au sein d'une communauté de « passionnés de la cuisine » ?
3. L'interface graphique

Contrairement à d'autres tâches d'évaluation, par exemple l'analyse d'opinions (cf. DEFT 2007 et 2009), les appréciations de la difficulté des recettes sur le site Marmiton.org ne représentent pas une sorte d' « intelligence collective » de gens qui ont essayé et évalué les recettes. Dans notre cas, le seul juge est l'auteur de la recette lui-même. Nous estimons que les 3 points que nous mentionnons ci-dessus, que nous allons examiner de plus près par la suite, représentent un fort biais pour le niveau « facile ».

Concernant les capacités culinaires, un sentiment de fierté peut conduire les auteurs à favoriser la note « facile ». Lorsqu'un auteur évalue sa recette comme étant « difficile », il ou elle s'expose implicitement à être considéré comme un cuisinier moyen par d'autres qui considéreraient la recette « facile ». Par conséquent, les auteurs auraient tendance à attribuer l'appréciation « facile » pour éviter de ternir leur réputation.

Il existe également certains indices qui révèlent l'intention de l'auteur quant au public visé. Un parcours un peu spéculatif du site révèle, par exemple, la recette de « quenelles de volaille, sauce financière », généralement considérée comme difficile. L'auteur a évalué cette recette comme étant facile, malgré la suite élaborée d'instructions qu'il a décrites et qui sont manifestement non triviales à accomplir. Les commentaires sur cette recette, laissés par des

internautes, témoignent du fort désaccord qui peut exister entre les différents jugements :

louve_2: *C'est bon, simple et rapide*

laure04130: *Bien compliqué pour un résultat insatisfaisant.*

La subjectivité du jugement se retrouve également dans le fait que l'on trouve deux versions d'un même plat marquées avec des niveaux de difficulté différents. Par exemple, on trouve « bouillabaisse facile et sa rouille » étiqueté « facile » et « bouillabaisse comme à Marseille » annoté « difficile ». En lisant la recette, on se rend compte que cette différence d'appréciation ne s'explique pas par une différence dans les séquences d'instructions (qui sont, en fait, comparables pour ces deux recettes). On peut supposer qu'il s'agit plutôt d'une position « idéologique » que chacun des auteurs adopte dans la présentation de sa recette : dans le premier cas « accessible à tous » et, dans le deuxième, « recette traditionnelle régionale ».

Enfin, lors de la saisie de l'appréciation par l'auteur, la valeur par défaut dans l'interface est « facile ». Il ne semble pas irraisonnable de supposer que parfois un auteur, que ce soit par paresse ou inadvertance, ne change pas cette valeur par défaut.

En vue de ces observations, nous concluons que ces trois critères rendent difficile l'évaluation de la difficulté de recettes de cuisine sur la base d'éléments purement textuels, au moins sur un corpus comme celui de Marmiton.org.

3.4 Conclusions sur la tâche 1

Nous avons présenté une approche pour la détection du niveau de difficulté de recettes de cuisine basée sur un mélange de méthodes d'apprentissage et de logique probabiliste. Les résultats préliminaires, obtenus sur une partie de 20% du corpus non utilisée pour l'apprentissage, sont assez décevants. De tels résultats médiocres pourraient s'expliquer par un mauvais choix d'algorithmes de classification. Cependant, une analyse manuelle du corpus d'évaluation nous laisse penser que l'évaluation du niveau de difficulté d'une recette est une tâche hautement subjective et que le résultat ne peut être déterminé de façon fiable sur la base du simple texte de la recette. Il serait intéressant de vérifier cette impression par une analyse approfondie des données structurées du site Marmiton.org qui prendrait en compte non seulement les recettes, mais aussi les utilisateurs (les auteurs et les gens qui laissent des commentaires). Nous proposons, si une expérience analogue à la tâche 4 de DEFT 2013 était proposée, que la classification de la recette prenne en compte un paramètre supplémentaire, notamment l'utilisateur et ses comportements.

4 Tâche 4 : détecter les ingrédients d'une recette

4.1 Description de la tâche 4

La tâche 4 du DEFT 2013 concerne l'identification des ingrédients dans les recettes de cuisine. Il s'agit d'une tâche d'extraction d'information dans laquelle un système doit fournir en sortie une liste des ingrédients repérés dans le titre et le texte d'une recette donnée en entrée. L'évaluation s'effectue en comparant la liste d'ingrédients obtenue par le système avec celle fournie par l'auteur de la recette. La mesure de « Mean Average Precision » (MAP) est utilisée pour l'évaluation.

4.2 Corpus et ressources fournis

Le corpus utilisé pour cette tâche (et pour les autres tâches de cette campagne) provient du site web Marmiton (MARMITON). Ce site, ouvert gratuitement au grand public, recense plus de 55 000 recettes de cuisine proposées par ses membres. Les recettes ont été extraites automatiquement du site pour constituer un corpus de test et un corpus d'apprentissage. Les organisateurs ont veillé à ce que les proportions des classes de recettes présentes dans les deux corpus soient les mêmes, mais sans vérifier que les fichiers de chaque corpus soient de la même taille. Pour la tâche 4, le corpus d'apprentissage contient 13 864 fichiers. Le corpus de test, quant à lui, contient 2 306 fichiers.

Une liste globale normalisée contenant (au moins) tous les ingrédients présents dans le corpus global (et éventuellement d'autres qui n'y figurent pas) a également été fournie. La figure Figure 2: extrait de la liste d'ingrédients. montre un extrait de cette liste.

```
2965 0 thon 1
3401 0 aneth 1
3401 0 carre-frais 1
3401 0 citron 1
3401 0 pate-feuilletee 1
3401 0 poivre 1
```

Figure 2: extrait de la liste d'ingrédients.

4.3 Une approche hybride

L'approche que nous avons adoptée pour cette tâche repose sur une combinaison de méthodes symboliques (à base de règles) pour l'extraction d'information et de techniques statistiques souvent utilisées dans les domaines de la recherche d'information et de l'apprentissage. Notre choix de méthodes a été motivé par les facteurs suivants :

1. Il s'agit d'une tâche typique du domaine de l'extraction d'information et, étant donnée la liste d'ingrédients fournie par les organisateurs, dans un monde idéal, le recours à des méthodes uniquement symboliques permettrait d'obtenir des résultats très satisfaisants ;
2. Cependant, le contenu du corpus de test n'a pas été filtré et il reste certains cas où la détection correcte des ingrédients est impossible sur la simple base du texte lui-même. Dans ces cas, et en vue de l'évaluation par la mesure MAP, il est toujours mieux de fournir une réponse « raisonnable » que de ne proposer aucun ingrédient. Les méthodes statistiques ont l'avantage de permettre ce type de robustesse.

Les cas que nous mentionnons au point 2 apparaissent dans les situations suivantes :

- Lorsque la préparation consiste simplement à mélanger les ingrédients, sans qu'ils ne soient mentionnés dans le texte lui-même. Par exemple, « mixer le tout », « mélanger tous les ingrédients », « mélanger l'ensemble », etc. Ce sont des cas de coréférence nominale pour lesquels on ne peut trouver l'antécédent sans considérer la liste d'ingrédients et le corps de la recette comme un texte uni.
- Lorsqu'un ingrédient figure dans la liste des ingrédients avec son nom complet, mais

qu'il est repris dans le corps de la recette par un terme plus générique. Par exemple, la liste d'ingrédients contient « filet de rascasse » alors que le corps de la recette mentionne « les filets » ou « le poisson », encore des cas de coréférence nominale qui sont souvent difficiles à résoudre de façon automatique[2].

- Lorsqu'un ingrédient est implicite dans une des opérations nécessaire au déroulement de la recette. L'auteur présuppose que le cuisinier saura à quel moment utiliser cet ingrédient. Par exemple, la liste d'ingrédients contient « huile d'olive » qui est inclus dans la recette dans l'instruction « faites revenir les oignons ».

Dans la section 4.3.1, nous décrivons les méthodes symboliques utilisées pour traiter les mentions « explicites » d'ingrédients. Dans la section 4.3.2, nous présentons le traitement des cas plus « incertains » par des méthodes statistiques. Finalement, nous présentons, à la section 4.4, nos conclusions sur cette tâche.

4.3.1 Traitement symbolique

Pour la tâche 4 du DEFT 2013, nous avons employé deux couches de traitement symboliques : une couche lexicale et une couche de règles. Pour la couche de traitement lexical, nous avons d'abord construit un lexique de toutes les expressions contenues dans la liste d'ingrédients fournie par les organisateurs. Les entrées du lexique sont représentées sous forme d'expressions régulières Java. Pour chaque élément de la liste d'origine, notre lexique stocke la version lemmatisée de cet élément.

Les entrées lexicales permettent un match « flou », c'est-à-dire que la recherche ne prend en compte ni la casse des caractères, ni les accents (tous les accents dans une position donnée sont considérés comme corrects). La figure 3 illustre la composition de notre lexique.

```
thon                          INGREDIENT
aneth                         INGREDIENT
carr[eéèëê] frais             INGREDIENT
citron                        INGREDIENT
p[aâäà]te feuillet[eéèëê]e    INGREDIENT
poivre                        INGREDIENT
```

Figure 3: extrait du lexique des ingrédients.

La recherche lexicale constitue la première étape du traitement. Les expressions trouvées dans le texte sont alors étiquetées `INGREDIENT`.

Après la recherche lexicale, s'applique la couche de règles sur des patrons linéaires de tokens. Nous avons implémenté des règles pour les situations suivantes :

- **Normalisation de la forme lemmatisée :** le système repère les ingrédients dans le texte sur la base du lemme. Cet ensemble de règles détecte et normalise les cas où les ingrédients doivent apparaître au pluriel dans la sortie, par exemple « marrons », « olives », « lasagnes ».
- **Expansion de termes génériques :** repérage d'ingrédients absents de la liste « gold

[2] En effet, la résolution des relations de coréférence constitue une tâche TAL à part entière (MUC-6, 1995) (MUC-7, 1997).

standard » sous forme simple, mais qui peuvent être utilisés pour expansion de termes. Par exemple, le mot « sauce » n'est pas un ingrédient possible en soi. Cependant, il figure parfois dans le corps des recettes, par exemple, dans une référence générique à un ingrédient de la liste d'ingrédients. Ces règles font l'expansion du mot « sauce » pour retrouver la forme complète la plus plausible dans le contexte.

- **Expansion de formes contractées :** pour trouver une forme complète. Par exemple, « chantilly » donne « creme-chantilly ».
- **Extraction d'ingrédients « implicites » :** pour dériver les noms d'ingrédients à partir de certains verbes. Par exemple, « fariner » et « farine », « poivrer » et « poivre », « saler » et « sel », etc.
- **Correction de l'étape de recherche lexicale :** pour corriger l'application trop simpliste des lexiques. Par exemple, dans l'expression « noix de beurre », seul « beurre » (et pas « noix ») est un ingrédient.

A la fin des étapes de traitement symbolique, la sortie de notre système consiste en une liste d'ingrédients « sûrs ».

4.3.2 Traitement statistique

Le fait d'utiliser MAP comme mesure d'évaluation laisse la place à l'incertitude dans les résultats (du fait de demander une liste ordonnée de possibilités). Paradoxalement, avec cette mesure, un système qui ne liste que des ingrédients 100% sûrs (des ingrédients correctement détectés dans un texte) aurait de moins bons résultats qu'un système qui énumérerait également tous les autres ingrédients possibles. Une partie importante de l'expérience est de dresser une liste ordonnée des ingrédients « incertains » selon leur plausibilité.

Pour enrichir la liste de résultats fournie par les règles, nous avons employé trois heuristiques.

La première a consisté à faire l'expansion de termes génériques. Par exemple, certaines expressions d'ingrédients génériques sont employées à la place de formes plus spécifiques. Tableau 1 ci-dessous donne quelques exemples.

Forme générique	Formes spécifiques
sauce	sauce-tomate, sauce-soja, sauce-worcestershire
porc	roti-de-porc, echine-de-porc, cotes-de-porc

TABLE 1 – Exemples d'expressions d'ingrédients génériques et de formes spécifiques correspondantes.

Une liste de formes complètes est générée à partir de la liste d'ingrédients fournie par les organisateurs en faisant l'expansion des termes qui apparaissent soit au début d'une mention d'ingrédient (ex. « farine » dans « farine-de-ble »), soit à la fin (ex. « agneau » dans « cote-d-agneau »). Les expansions obtenues sont ensuite ordonnées selon leur fréquence

dans le corpus de référence. Il est important de noter que nous n'avons disposé d'aucune ontologie ou taxonomie sur le domaine en question. Par conséquent, certains types d'expansion n'ont pas pu être opérés. Par exemple, l'expansion de « poisson » en une liste de tous les types de poissons (« cabillaud », « morue », « perche », etc.), ou bien l'expansion de « légume » en « chou », « carotte », « potiron », etc.

La deuxième heuristique, correspondant également à une expansion de termes, vise à identifier les ingrédients dans les recettes avec des instructions peu informatives, telles que « mélangez tous les ingrédients ». Pour effectuer cette étape, nous avons utilisé une autre information sur les recettes, notamment le type de plat préparé (entrée, plat principal ou dessert). Un simple classifieur d'entropie maximale nous a permis d'obtenir des résultats de 88% de précision pour la tâche de classification de recettes selon le type de plat préparé. Il est clair que la liste d'ingrédients possibles est en partie restreinte par le type de plat en question. Il est très peu probable qu'un dessert contienne une épaule d'agneau ou qu'un plat principal contienne de la crème Chantilly. Sur la base de cette distinction, nous avons réordonné tous les ingrédients en fonction de leur fréquence d'apparition dans les différents types de plats. Lors du traitement (après toutes les autres étapes d'extraction et d'enrichissement), nous avons classifié la recette en fonction du type de plat et enrichi la liste d'ingrédients possibles avec les ingrédients les plus probables pour ce type de plat.

La troisième et dernière heuristique était d'ajouter tous les autres ingrédients selon leur ordre d'apparition dans le corpus.

4.4 Conclusions sur la tâche 4

Le système que nous avons développé pour cette tâche repose sur un lexique fourni par les organisateurs, des règles écrites manuellement et spécifiques au domaine, et un ensemble d'heuristiques pour trouver des ingrédients « implicites ». Il est clair que l'approche est biaisée pour accommoder la mesure d'évaluation MAP, qui privilégie le rappel sur la précision. Alors que l'utilisation de cette mesure permet la comparaison des résultats avec d'autres campagnes qui utilisent la même mesure, nous estimons que la MAP n'est pas la mesure optimale pour une tâche d'extraction d'information pure. En outre, il est clair que l'utilisation de la mesure MAP, adaptée dans un contexte académique, ne donne pas d'idée réelle de l'utilité de l'application. En effet, nous ne connaissons aucune application réelle pour laquelle il est plus avantageux de fournir un ensemble de faits « plausibles » que de fournir une liste de faits « sûrs ».

5 Références

(s.d.). Consulté le 5 6, 2013, sur MARMITON: http://www.marmiton.org

(2012). Consulté le 5 6, 2013, sur Computer Cooking Contest: http://computercookingcontest.net

BEEDKAR, K., DEL CORRO, L., & GEMULLA, R. (2013). Fully Parallel Inference in Markov Logic Networks. *BTW*, 205-224.

CHANG, A. X. (2011, 9 14). Consulté le 5 6, 2013, sur Tokens Regexp: http://nlp.stanford.edu/software/tokensregex.shtml

CHE, W., & TING, L. (2010). Jointly Modeling WSD and SRL with Markov Logic. (A. f. Linguistics, Éd.) *Proceedings of the 23rd International Conference on Computational Linguistics*, 161-169.

EVERITT, B. (1992). *The analysis of contingency tables* (Vol. 45). Champman & Hall/CRC.

KARLGREN, J., & CUTTING, D. (1994). Recognizing Text Genres with Simple Metrics Using Discriminant Analysis. *Proceedings of the 15th Conference on Computational Linguistics. 2*, pp. 1071-1075. Association for Computational Linguistics.

KESSLER, B., NUMBERG, G., & SCHÜTZE, H. (1997). Automatic Detection of Text Genre. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics* (pp. 32-38). Association for Computational Linguistics.

NIU, F., RE, C., DOAN, A., & SHAVLIK, J. (2011). Tuffy: scaling up statistical inference in Markov logic networks using an RDBMS. *Proceedings of the VLDB Endowment, 4*(6), 373-384.

POON, H., & DOMINGOS, P. (2009). Unsupervised Semantic Parsing. (A. f. Linguistics, Éd.) *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, 1*, 1-10.

RICHARDSON, M. D. (2006). Markov Logic Networks. *Machine Learning, 62*(1-2), 107-136.

RIEDEL, S., & MEZA-RUIZ, I. (2008). Collective Semantic Role Labelling with Markov Logic. (A. f. Linguistics, Éd.) *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, 193-197.

RIEDELL, S. (2008). Improving the Accuracy and Efficiency of Map Inference for Markov Logic. *UAI '08: Proceedings of the Annual Conference.*

STANFORD. (2010, 1 10). *Stanford CoreNLP: A Suite of Core NLP Tools*. Consulté le 5 6, 2013, sur http://nlp.stanford.edu/software/corenlp.shtml

Participation de Orange Labs à DEFT 2013

Olivier Collin, Aleksandra Guerraz, Yannick Hiou, Nicolas Voisine
Orange Labs
2, avenue Pierre Marzin, 22307 Lannion Cedex, France
{olivier.collin,aleksandra.guerraz}@orange.com
{yannick.hiou,nicolas.voisine}@orange.com

RESUME

Cet article décrit notre participation à DEFT2013. Nous détaillons tout d'abord nos réponses et résultats pour les tâches 1 et 2 qui sont des problèmes de classification de recettes : classification en niveaux de difficulté (*facile, difficile…)* pour la tâche 1 et vers une typologie (*entrée, plat principal, dessert*) pour la tâche 2. Nous poursuivons ensuite par le traitement de la tâche 3 dont le but était de mettre en correspondance les titres des recettes en regard de leur description. Pour cette tâche, nous avons testé l'utilisation d'une distance sémantique entre mots, distance pré-calculée à partir de Wikipédia français.

ABSTRACT

Orange Labs participation to DEFT 2013

This paper describes our participation to DEFT2013. On the one hand we first detail our answers and results related to the tasks 1 and 2 which are classification tasks applied to recipes. Task 1 focused on the difficulty level of each recipe *(facile, difficile…)* and task 2 to the natural typology of recipes: *entrée, plat principal, dessert*. On the other hand, the goal of task 3 was to find the link between a recipe description and a title list. For this task, we applied a semantic distance calculus between words, pre-calculated on Wikipedia French corpus.

MOTS-CLÉS : classification supervisée, co-clustering, distance sémantique, Wikipédia, Khiops, TiLT
KEYWORDS : supervised classification, co-clustering, semantic distance, Wikipedia, Khiops, TiLT

1 Introduction

Le défi 2013 proposait 4 tâches, nous avons participé aux tâches 1, 2 et 3. Les deux premières tâches ont été traitées par des algorithmes de classification supervisée. La tâche 3 a été traitée par des calculs de similarité textuelle s'appuyant sur une métrique de distance entre mots. La tâche 4 n'a pas été traitée. Si les tâches 1 et 2 sont plutôt classiques, la nature des données et la confusion des étiquettes ont apporté une difficulté supplémentaire à ce type de tâche. Nous pensons avoir apporté un éclairage intéressant sur la tâche 2 par l'utilisation d'un outil de co-clustering permettant une factorisation utile des variables. En ce qui concerne la tâche 3, nous avons choisi d'explorer une métrique basée sur une distance sémantique inter mots introduite par Cilibrasi (Cilibrasi et Vitanyi 2006, 2007), (Cilibrasi *et al.*, 2009). Nous nous sommes placés dans un cadre applicatif que nous avons repris dans le challenge « AI Mashup Challenge » (Belaunde *et al.*, 2013). Ce cadre consiste à rapprocher un document (recette) d'un ensemble de titres issus de données vidéos (présentations vidéos

d'une recette).

Dans la section 2, nous décrivons le corpus et les prétraitements généraux que nous avons effectués pour les trois tâches. Notre contribution aux tâches 1 et 2 est présentée dans la section 3. La section 4, quant à elle, détaille l'approche que nous avons utilisée pour la tâche 3. Nous terminons enfin par une discussion sur les résultats et une conclusion sur notre participation au défi.

2 Prétraitements généraux

Le corpus d'entrainement est constitué de 13864 recettes de cuisine. Les recettes sont au format XML. Pour chaque recette, les champs « id », « titre », « type », « niveau », « coût », « ingrédients » et « préparation » sont renseignés. De plus, un fichier récapitulatif de ce corpus a été fourni. Il regroupe des informations collectées dans chaque recette ; on y trouve pour chaque recette un champ comportant les ingrédients normalisés (sans accent, en minuscule, en multi mots) et un champ composé du nombre de ces ingrédients normalisés. Au final, nous avons conservé chacun des champs des recettes mais nous n'avons pas utilisé les ingrédients normalisés. Nous n'avons pas non plus utilisé le fichier fournissant la liste des titres séparément des recettes.

Outre le prétraitement linguistique que nous avons effectué pour la tâche 3 (et décrit en tâche 3), nous avons dû tout d'abord traiter l'encodage des recettes. Cet encodage annoncé comme étant de l'UTF8 était ponctuellement bruité notamment par un encodage parasite de caractères HTML. L'encodage des données XML issues du WEB et notamment des flux RSS est très souvent bruité. Nous avons d'autre part converti en minuscules l'ensemble des données des corpus d'apprentissage et de test.

3 Tâche 1 et 2, généralités

Pour ces tâches, nous avons testé deux méthodes de classification différentes : une méthode utilisant le principe de maximum d'entropie et une autre qui exploite l'hypothèse Bayesienne naïve. Rappelons que la tâche 1 consistait 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*. Quant à la tâche 2, elle consistait à classifier la recette selon le type de plat parmi *entrée*, *plat principal* et *dessert*. Des premiers essais de classification nous ont permis d'analyser les caractéristiques de chacune de ces tâches. La tâche 1 se caractérise par un étiquetage subjectif de la notion de difficulté d'un plat. Un calcul d'accord inter-annotateur aurait probablement montré une forte divergence des étiquettes. D'autre part, le corpus d'apprentissage est fortement déséquilibré et contient relativement peu d'exemples de la classe *difficile*. La tâche 2 se caractérise aussi par une confusion importante portant sur les classes *entrée* et *plat* alors que la classe *dessert* est plutôt bien différenciée des deux autres classes. Dans les deux cas, on se retrouve donc assez loin d'un corpus d'apprentissage idéal.

Concernant ces deux tâches, le premier essai nous a permis de réaliser une référence en utilisant une technique standard de type « Maximum d'Entropie » (Nigam *et al.*, 1999), notre but étant ensuite de tester un outil développé par Orange Labs : Khiops (Boullé, 2008). Pour réaliser notre classifieur de type Maximum d'Entropie, nous avons simplement téléchargé et adapté les entrées/sorties du code mis à disposition par (Tsuruoka *et al.*, 2009) à l'adresse

http://www.nactem.ac.uk/tsuruoka/maxent/.

Pour ces deux tâches, pour les autres essais, nous avons donc utilisé Khiops, un outil de préparation des données et de modélisation pour l'apprentissage supervisé et non-supervisé. Cet outil est basé sur une méthode de classification qui exploite l'hypothèse Bayesienne naïve (Boullé, 2007). Cette méthode estime les probabilités conditionnelles univariées. Elle effectue des discrétisations et groupements de valeurs optimaux pour les variables numériques et catégorielles. Elle recherche un sous-ensemble de variables consistant avec l'hypothèse Bayesienne naïve, en utilisant un critère d'évaluation selon une approche Bayesienne de la sélection de modèles et des heuristiques efficaces d'ajout/suppression de variables. Enfin, elle moyenne l'ensemble des modèles évalués en utilisant un lissage logarithmique de la distribution a posteriori des modèles. L'outil Khiops ne nécessite aucun paramétrage, ce qui constitue son point fort. Cet outil, développé par Orange Labs, est utilisé intensivement pour de nombreux problèmes d'analyse et de classification de données mais moins souvent sur des données textuelles. Il nous a permis de réaliser la classification supervisée des documents pour les tâches 1 et 2, ainsi qu'une analyse des variables pour chacun des modèles générés. Il offre, d'autre part, un algorithme de « co-clustering » que nous avons appliqué à des regroupements documents/mots. Ces regroupements sont à la fois utiles par la sémantique portée mais constituent aussi de nouvelles variables synthétiques permettant de réaliser, voire améliorer les tâches de classification supervisées. Khiops fournit d'autre part un ensemble d'outils d'analyse des variables qui permettent notamment d'ordonner et visualiser le poids relatif de chacune d'entre elles.

Nous n'avons utilisé aucun prétraitement de type linguistique (lemmatisation…) pour ces deux tâches. Par contre, différents types de variables ont été utilisés conjointement : des variables catégorielles comme le champ « coût » ainsi que des variables numériques comme le nombre de mots du champ « préparation ».

Pour pré-évaluer ces tâches, nous avons partagé le corpus d'apprentissage fourni en deux sous-corpus : 70% pour l'apprentissage et 30% pour le test.

Pour la tâche 1, nous observons un fort déséquilibre du nombre d'exemples par classe, la classe *difficile* étant particulièrement sous représentée.

3.1 Tâche 1

3.1.1 Essai 1

L'essai 1 utilise le classifieur de type « Maximum d'Entropie ». Les variables utilisées sont les variables catégorielles « coût » (*bon marché, moyen, assez cher*) et « type » (*entrée, plat, dessert*). Nous prenons également en compte comme variables les mots qui sont présents dans le titre, la liste d'ingrédients et dans la préparation. Certaines catégories de mots ont été filtrées en utilisant notamment la liste de « stop words » fournie par NLTK (http://nltk.org/). Ce sont notamment les articles, prépositions, pronoms personnels, adjectifs possessifs, négations, conjonctions, auxiliaires être et avoir. Nous avons en plus supprimé la plupart des termes représentant des valeurs numériques, des symboles de mesure de contenance, de poids… Les petits mots ne contenant qu'une ou deux lettres ont été aussi éliminés. Les variables numériques utilisées sont le « nombre de mot dans la

préparation » et le « nombre d'ingrédients ». Ces variables ont été discrétisées linéairement.

Les résultats sur notre sous-corpus de test sont très mauvais pour cette tâche : toutes les recettes sont classées comme "facile", soit un taux d'erreur de 66%.

Les fréquences relatives des classes sont probablement la cause de ce problème. Nous ne nous attendions donc pas à de bons résultats pour cet essai.

3.1.2 Essai 2

Pour la tâche 1 utilisant Khiops (essai 2), nous utilisons les variables catégorielles suivantes : « coût » (*bon marché, moyen, assez cher*) et « type » (*entrée, plat, dessert*). Les variables numériques utilisées sont : « nombre de mot dans la préparation » et « nombre d'ingrédients ». Nous prenons également en compte comme variables les mots qui sont présents dans le titre de la recette et dans sa préparation. Les variables « mots » liées aux ingrédients n'ont pas apporté une information utile pour cette tâche avec cet outil. Ce choix de variables a été réalisé suite à l'analyse de variables fournie par Khiops.

Les graphes suivants produits par l'environnement de Khiops illustrent cette analyse pour les variables les plus pertinentes pour notre tâche.

Nous observons ainsi une corrélation entre le niveau de difficulté d'une recette et son coût (figure 1). Cette figure illustre le fait que lorsqu'une recette est « *assez cher* » il est moins probable qu'elle soit très facile à réaliser.

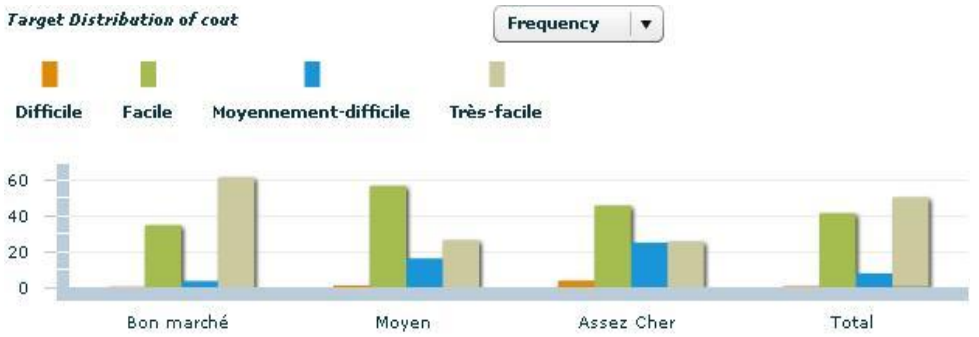

FIGURE 1 – Distribution des valeurs de la variable « coût ».

Le nombre de mots de la préparation est aussi une variable importante (figure 2).

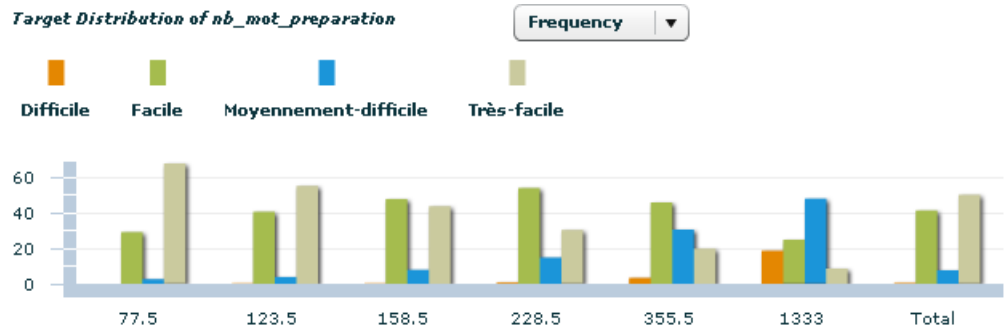

FIGURE 2 – Distribution des valeurs de la variable « nombre de mots de la préparation ».

Plus la recette est longue, plus la difficulté de la réalisation augmente.

Parmi les varibles sélectionnées par Khiops, nous trouvons des mots comme « poche » et

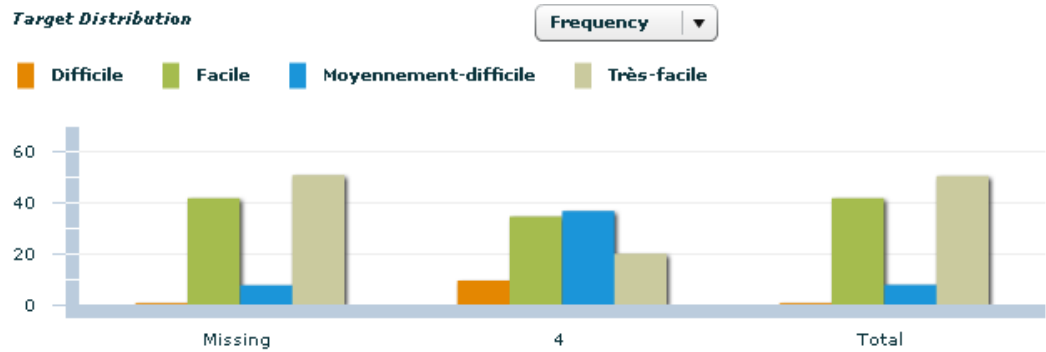

FIGURE 3 – Distribution pour le mot « poche ».

« salade ». Les figures 3 et 4 présentent la distribution de ces mots. Le mot « poche » est absent de la plupart de recettes. En revanche, quand il apparaît il est significatif : il est plus probable que la recette soit difficile à réaliser. Dans le corpus le mot « poche » apparaît dans les contextes « poche à douille » ou « poche à encre », il s'agit, en effet, de termes techniques. En ce qui concerne le mot « salade » (figure 4), il est plus probable qu'une recette soit facile à réaliser quand le « mot » salade apparaît.

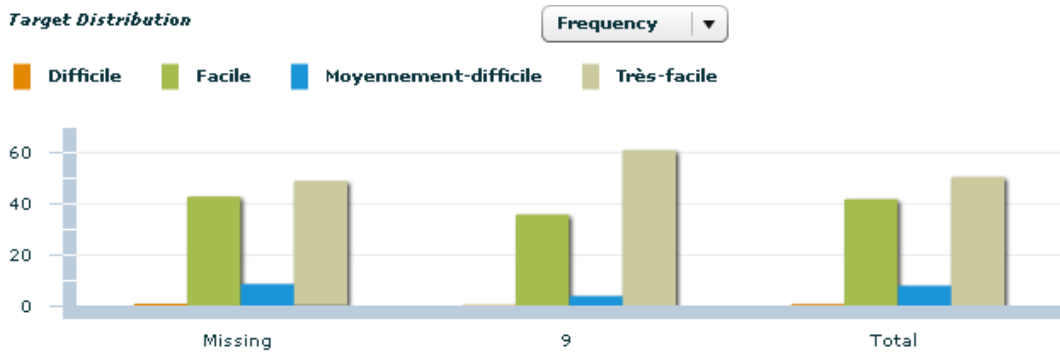

FIGURE 4 – Distribution pour le mot « salade ».

Outre la performance atteinte, cet environnement d'analyse s'avère très utile dans la compréhension du problème. Les tests de mise au point ont montré que Khiops proposait un résultat intéressant pour cette tâche. Mais, même si cet outil souffre moins du manque de données associées à la classe *difficile* et propose une modélisation et une analyse de variables pertinentes, la tâche reste délicate.

3.2 Tâche 2

3.2.1 Essai 1

L'essai 1 utilise le classifieur de type « Maximum d'Entropie », prenant à peu près les mêmes variables d'entrée que pour la tâche 1. Bien évidemment, la variable « type » n'est pas présente, mais nous avons utilisé la variable « niveau». Pour la mise au point de cet essai nous avons constitué un corpus d'apprentissage équi-réparti contenant 2268 exemples par classe, soit environ 50% des exemples disponibles. Les tests préliminaires ont donc été réalisés sur les exemples restants.

Contrairement à la tâche 1, les résultats sont utilisables. Nous avons notamment observé que des mots comme "tarte" et "quiche" sont « confondus » pour ces deux classes. Nous avons constaté que la classe « *dessert* » est plutôt bien discriminée, un *dessert* étant confondu avec un *plat* ou une *entrée* moins de 1% des cas. La confusion provient essentiellement des classes *entrée* et *plat principal*, une *entrée* étant confondue avec un *plat* presque une fois sur deux, un *plat* étant parfois confondu avec une *entrée* dans près de 8% des cas.

Pour notre test final, nous avons donc conservé une equi-répartition des données d'apprentissage de 3200 exemples par classes.

3.2.2 Essai 2

Pour la tâche 2 utilisant Khiops (essai 2), nous utilisons les variables catégorielles « coût » et « niveau » ainsi que les variables numériques « nombre de mot dans la préparation » et « nombre d'ingrédients », ainsi que les variables « mots » présents dans le titre et dans la préparation.

Pour cette tâche, nous utilisons Khiops pour une modélisation non-supervisée et supervisée. Nous avons donc aussi recours à sa fonctionnalité de co-clustering (Boullé, 2011). Les deux dimensions que nous utilisons sont les textes, et les mots. Après convergence de l'algorithme, nous obtenons 102 clusters de textes et 219 clusters de mots. Nous utilisons ensuite les clusters de textes pour la modélisation supervisée. Les recettes ne sont plus représentées par un ensemble de mots, mais sont classées dans un des 102 clusters. Nous utilisons également les variables catégorielles « coût » et « niveau » pour la modélisation supervisée et les variables numériques. Cependant ces variables n'ont pas de gros impact sur le modèle. La variable « cluster » a un pouvoir prédictif beaucoup plus significatif que les autres variables. L'analyse des regroupements de documents réalisés par le co-clustering permet d'expliquer ce résultat : les regroupements effectués de manière non-supervisée correspondent globalement à des regroupements d'entrées, de plat et de desserts. Si nous examinons plus finement les clusters de textes, nous observons en réalité une spécialisation par « cluster » dont voici quelques exemples :

Les recettes *Gâteau pomme-rhubarbe crousti-moelleux, Gâteau automnal, Gâteau aux fruits jaunes et aux pignons, Clafoutis corrézien* appartiennent au même « cluster », caractérisant un dessert de type gâteau. Nous trouvons également des clusters de desserts à base de fruits ou encore à base de chocolat.

De la même manière, les entrées *Velouté de printemps, Soupe aux marrons et aux carottes, Crème de céleri glacée à la menthe, Crème de cresson au caviar de Sevruga, Soupe d'endive à la bière* ont été regroupées au sein d'un même cluster qui semble correspondre à des entrées chaudes de type soupe ou crème.

Un autre exemple concerne des entrées à base de fruits de mer ou de poissons: *Pamplemousse de la mer, Mini-concombres farcis du pêcheur, Miettes de crabe et pamplemousse, Brochette des îles, Salade d'été au saumon fumé, feta, pommes et carottes…*

Pour les plats principaux, nous avons par exemple un cluster contenant *: Coq au vin de la mère Michèle, Bolognaise façon Mag, Bœuf mijoté façon bourguignon, Garbure landaise, Daube de canard au madiran, Daube de sanglier à la provençale …*

Finalement, examinons un cluster qui contient des entrées et des plats principaux et montre la difficulté de dissocier les classes *entrée* et *plat*: *Quiche à la truite fumée, aux courgettes et à la mozzarella (plat), Tarte à l'aubergine et à la brousse (entrée), Quiche à la tomate et aux oignons (plat), Quiche chorizo et tomate (entrée), Tarte moutardée au thon, tomates et poireaux (entrée), Tarte légère courgettes, jambon et chèvre gratiné (entrée), Tarte lardons champignons camembert (plat), Tarte à la tomate et au roquefort (plat), Quiche au poulet et à l'estragon (plat), Quiche au parmesan et tomates (entrée), Quiche au crabe (entrée).* On retrouve dans ces titres et dans ces recettes les termes « quiche » et « tarte » qui conviennent à un plat ou une entrée.

La figure 6 illustre la distribution des classes cibles (*entrée, plat, dessert*) pour un regroupement de cluster réalisé par Khiops. On visualise assez nettement la spécialisation réalisée par le co-clustering.

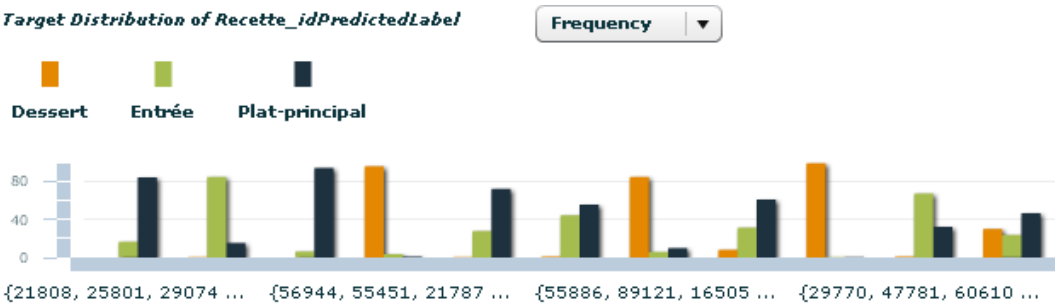

FIGURE 6 – Distribution de clusters

3.2.3 Essai 3

Le 3ème essai reprend le même algorithme que l'essai 2, mais le co-clustering a été réalisé non pas sur les données d'apprentissage mais sur l'ensemble des données d'apprentissage et de test, c'est à dire sur environ 16000 recettes.

4 Tâche 3

Cette tâche a été l'occasion de tester une forme de distance sémantique mot/mot généralisée de manière simple à une distance document/document. Nous nous sommes placés dans un cadre applicatif dans lequel les documents courts (titres) sont en quantité limitée mais gagnent à être ordonnés de manière pertinente.

4.1 Prétraitements

Pour cette tâche, l'espace de représentation des documents a été modifié. Nous avons utilisé notre outil de Traitement Automatique des Langues, TiLT (Heinecke *et al.*, 2008), pour effectuer une lemmatisation associée à un étiquetage syntaxique des documents. Certains traitements ont été réalisés pour tous les types de champs (titre, ingrédients, préparation). Les termes liés aux catégories suivantes été retirés pour tous les types de champs : pronoms, prépositions, déterminants, conjonctions, adverbes, interjections, ainsi que les quantités numériques de tous types. Dans tous les cas, nous avons aussi cumulé, dans un même vecteur, les formes obtenues ainsi que les lemmes correspondants, de manière unitaire, afin d'obtenir une liste de termes uniques.
Nous avons ensuite réalisé des filtrages différenciés en fonction des types de champs (titre, ingrédients, préparation). En ce qui concerne les titres, seules les formes ont été conservées, pas leurs lemmes. Pour les ingrédients, ce sont les formes et les lemmes qui ont été retenus. Les préparations sont aussi représentées par leurs formes et leurs lemmes mais elles ont subi un filtrage supplémentaire puisque nous avons aussi ôté les adjectifs et les verbes. En effet, le rapprochement entre titres et recettes ne fait pas ou peu intervenir adjectifs et verbes. Toutefois, nous avons conservé les adjectifs et les verbes des titres et des ingrédients car le contexte de lemmatisation générait des erreurs du type nom/adjectif telle que « saumon » qui aurait été filtré suite à une erreur l'étiquetant comme « adjectif ». Au final, les ingrédients et les préparations correspondantes sont regroupés en un seul vecteur constitué d'éléments uniques. L'utilisation des lemmes est surtout destinée à étendre la représentation de la recette afin d'augmenter l'intersection entre le vecteur ingrédients-préparation et les mots des titres. Les multi-mots détectés ne sont pas utilisés mais décomposés en mots simples. Le résultat de ce traitement appliqué, par exemple, à la recette « Risotto de quinoa aux champignons » donne la représentation suivante :
Titre : *risotto quinoa champignons*
Document (ingrédients et préparation) : c*hampignons champignon frais quinoa noix oignon sel poivre poivrer persil morceaux morceau sauteuse sauteur huile eau temps risotto poêle bouillon sauce*

Cet exemple montre un taux de recouvrement des mots du titre avec le contenu de la recette de 100%.

Cette représentation a été testée parmi d'autres et s'est avérée nécessaire et suffisante pour cette tâche. Nous limitons ainsi la taille des vecteurs à comparer ce qui limite le temps de calcul notamment le temps nécessaire pour appliquer la mesure de distance présentée au chapitre suivant.

4.2 Distance sémantique

Nous avons utilisé et adapté la mesure proposée par (Cilibrasi et Vitanyi, 2006). Pour rappel, si x et y sont deux termes, f(x) la fréquence de x , f(y)la fréquence de y, f(x,y) la fréquence de x ET y, et M une valeur de normalisation (nombre de documents), la « distance » est la suivante :

$$D(x,y) = \frac{max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log M - min\{\log f(x), \log f(y)\}}$$

Cette mesure vérifie les deux premiers critères d'une distance mathématique (symétrie et séparation) mais pas le critère d'inégalité triangulaire :

D(x,y) = D(y,x) (symétrie)
D(x,y) = 0 ⇔ x=y (séparation)
D(x,z) <= D(x,y)+D(y,z) non vérifié

Cette mesure donne donc une forme de proximité sémantique qui varie entre 0 et 1, 0 indiquant que x est « sémantiquement » proche de y et 1 que x est « sémantiquement très éloigné de y ». Les logarithmes et la constante de normalisation donnent une valeur qu'il faut plutôt interpréter comme une relation d'ordre : x est plus proche de y que de z.

Cette mesure a été initialement proposée en utilisant les comptages fournis par Google. Pour une utilisation concrète faisant appel à un accès intensif à des centaines de millions de distances, l'utilisation de requêtes vers Google n'est pas possible. Nous avons choisi de réaliser ce calcul à partir des documents de Wikipédia français. Une fois le calcul terminé, nous disposons d'une ressource de « distances sémantiques entre termes » accessible en temps réel.

Ce calcul a été réalisé en 2010 à partir d'une version de Wikipédia datant de fin 2009. Les documents utilisés sont les paragraphes issus naturellement des pages de Wikipédia lors du filtrage des méta-données de formatage des pages de Wikipédia. Le calcul fréquentiel f(x,y) est donc effectué dans le contexte d'une phrase ou d'un paragraphe. Après traitement, nous avons extrait environ 6 000 000 paragraphes et constitué une matrice de 193 999 258 distances associées à autant de couples de mots simples. La matrice utilisée ici n'intègre ni les entités nommées, ni les multi-mots.

C'est cette ressource que nous avons voulu tester sur la tâche 3, l'idée étant de calculer une distance globale entre chaque titre et chaque recette par le biais des distances individuelles entre chaque mot des titres et de la recette. Ces distances étant issues d'une ressource initialement indépendante de la tâche mais relativement importante par sa taille et la variété de ses contenus, il est intéressant de pouvoir la tester dans un domaine de spécialité.

Autre point important, ce type de mesure permet d'ordonner la liste des titres en fonction des termes de la recette même si les termes du titre n'apparaissent pas parmi les termes de la recette. Il suffit qu'un terme du titre possède une entrée dans notre matrice en relation avec un mot de la recette pour avoir une distance utilisable.

En ce qui concerne la recette du « Risotto de quinoa aux champignons », nous pouvons ainsi potentiellement récupérer les distances entre les mots du titre (*risotto quinoa champignons)* et tous les mots de la recette (c*hampignons champignon frais quinoa noix oignon sel poivre*

poivrer persil morceaux morceau sauteuse sauteur huile eau temps risotto poêle bouillon sauce).

Nous pouvons donc réaliser ce calcul pour chaque couple (titre, recette). Si T est le nombre de termes du titre et R le nombre de termes de la recette, le nombre de distances à considérer est T×R. Nous avons ensuite choisi une métrique globale faisant intervenir les distances unitaires et synthétisant la proximité entre deux documents, en l'occurence un titre et une recette. Des essais préliminaires ont montré qu'il était plus pertinent de retenir, pour chaque terme d'un titre, uniquement le terme de la recette le plus proche, donc de distance minimum. En ce qui concerne la recette précédente, nous obtenons donc naturellement une distance minimum de 0 pour les termes communs au titre et à la recette :

D_{min}(risotto,risotto) = 0.0
D_{min}(quinoa,quinoa) = 0.0
D_{min}(champignons,champignons) = 0.0

Pour d'autres titres, nous obtenons des valeurs comprises en 0 et 1 pour chaque terme en relation avec l'élément le plus proche de la recette. Par exemple pour le titre représenté par (gâteau chocolat noix coco) :

D (gâteau,*noix*) = min{ D (gâteau,*m*)| *m* mot de la recette}
= 0.182777617404
D (chocolat,*noix*) = min{ D (chocolat,*m*)| *m* mot de la recette}
= 0.192853987214
D (noix,*noix*) = min{ D (noix,*m*)| *m* mot de la recette}
= 0.0 (terme commun)
D (coco,*noix*) = min{ D (coco,*m*)| *m* mot de la recette}
= 0.131960109813

Pour le titre représenté par (gratin courgettes pain ail)

D (gratin, *champignons*) = min{ D (gratin,*m*)| *m* mot de la recette}
= 0.200901512394
D (courgettes, *sauteur*) = min{ D (courgettes,*m*)| *m* mot de la recette}
= 0.190081305148
D (pain, *persil,*) = min{ D (pain,*m*)| *m* mot de la recette}
= 0.17466164279
D (ail, *oignon*) = min{ D (ail,*m*)| *m* mot de la recette}
= 0.0799472060861

Ces valeurs sont donc le reflet du calcul issu du corpus de Wikipédia. Si on peut établir facilement le lien (ail, oignon) ou (coco, noix) il est plus difficile d'expliquer la relation (courgettes, sauteur) !

La mesure globale que nous avons ensuite retenue est la moyenne de ces distances minimales. Si T est un titre de N termes et R une recette, la distance globale est :

$D(T,R) = \frac{1}{N} \sum_i^j D_{min}(T_i, R_j)$

Nous obtenons une mesure globale de proximité qui vaut 0 si les termes du titre sont

contenus dans les termes de la recette. Cette mesure tend vers 1 si aucun des termes du titre n'est contenu dans la recette et si de plus ces termes sont « éloignés » des termes de la recette. La moyenne permet de normaliser la mesure par rapport à la longueur du titre. Nous avons bien une mesure qui permet d'ordonner les titres de manière relative dans tous les cas, même si aucun terme du titre n'apparaît dans la recette, ou presque… En effet, si cela est vrai pour la majorité des titres, certains titres dont les termes ne sont jamais apparus dans Wikipédia ne possèdent aucun lien avec d'autres termes. Un exemple est le titre « Tiou Yap ». Dans ce cas la distance à chaque recette est de 1 ce qui reste cohérent.

Les 3 essais portant sur cette tâche sont des variantes de cette mesure :

- le premier essai n'utilise pas la matrice de distance issue de Wikipédia mais fixe à 0 la distance d'un terme du titre apparaissant dans la recette et à 1 s'il n'apparaît pas. C'est une sorte de limite binaire par défaut de la distance utilisée.

- le deuxième essai utilise la métrique décrite calculée à partir de Wikipédia français.

- le troisième essai utilise aussi la métrique décrite mais nécéssite deux matrices. La première est toujours celle issue de Wikipédia français. La deuxième a été calculée à partir du corpus d'apprentissage de la tâche. Toutes les recettes (titre, ingrédients, préparation) ont servi à calculer une matrice de distances mot/mot relative à ces données. Au final, nous utilisons une forme de vote qui consiste à choisir la distance minimale issue de l'une ou l'autre des deux matrices. Nous faisons l'hypothèse que la matrice calculée sur les recettes des données d'apprentissage est plus spécifique et pertinente que celle de Wikipédia mais moins complète.

Pour des raisons de temps les essais 2 et 3 ont été appliqués sur les 50 premiers résultats de l'essai 1 ce qui correspond plutôt à un ré-ordonnancement du premier essai.

5 Résultats et discussions

5.1 Tâche 1

Pour notre premier essai, utilisant le classifieur de type Maximum d'Entropie, les résultats n'ont pas été satisfaisants.

Concernant l'essai 2 utilisant Khiops, notre 2ème place semble confirmer les qualités de cet outil. Khiops gère plutôt bien cette tâche que nous estimions difficile compte tenu du type et des quantités relatives des données d'apprentissage. Une technique discriminante appliquée à une sélection de variables effectuée par Khiops aurait peut être pu nous hisser plus haut !

5.2 Tâche 2

Pour cette tâche, deux constats sont à faire. Tout d'abord, les résultats des deux premiers essais sont assez proches avec des techniques différentes. L'algorithme de type Maximum d' Entropie s'en sort plutôt bien avec le jeu de données d'apprentissage fourni.

Par contre, la puissance de modélisation de Khiops ne semble pas avoir permis de surpasser les algorithmes concurrents. Nous sommes un peu surpris du positionnement, compte tenu des résultats habituels de Khiops. Son utilisation sur des variables majoritairement textuelles, ce qui n'est pas son domaine applicatif standard, nécessite peut être des avancées

algorithmiques ou bien des prétraitements amont plus adaptés.

Pour le 3ème essai nous avons réalisé un co-clustering sur les données d'apprentissage et de test. Cette technique améliore généralement, au moins légèrement, les performances d'un co-clustering réalisé uniquement sur les données d'apprentissage (essai 2). Compte-tenu des résultats inférieurs du 3ème essai, nous suspectons un problème technique lors de notre essai.

Toujours est-il que Khiops nous donne une vision synthétique des données qui peut être utile dans d'autres cadres applicatifs : sous-catégorisation plus spécialisée des types de plats ou rapprochement de recettes par rapport à peu d'exemples de recettes prototypiques. Il nous a aussi permis d'analyser rapidement le chevauchement des classes *entrée*/*plat* et des variables associées.

5.3 Tâche 3

Pour cette tâche, nous avions choisi de tester une métrique particulière. À la vue de nos résultats, il semblerait que ce choix soit pertinent. Mais si nous regardons de plus près les résultats des différents essais, ce choix est moins évident. En effet, la distance binaire naïve du 1er essai produit un meilleur résultat que le 2ème essai qui utilise les distances mot/mot générales issues de Wikipédia. Il semblerait donc que la prise en compte de ces valeurs réelles entre 0 et 1 agisse plutôt comme du bruit. On peut aussi penser que notre stratégie qui a consisté à utiliser un pré-ordonnancement des cinquante premiers titres avec une métrique naïve, puis à réordonner ces titres avec une deuxième métrique élimine des bonnes solutions qui sont en quelques sorte oubliées…

Le 3ème essai qui donne les meilleurs résultats en utilisant l'information spécifique au domaine (corpus d'apprentissage) nous encourage toutefois à continuer dans l'exploration de cette voie.

6 Conclusion

Cette participation à DEFT 2013 nous a permis de tester des algorithmes développés par Orange Labs sur des données qui peuvent paraître loin de nos préoccupations habituelles. La spécificité des problèmes de classification des tâches 1 et 2 nous intéressait justement par leur production peu classique. L'outil testé (Khiops) montre ses capacités à s'adapter à de nombreuses tâches et particulièrement ici à des variables majoritairement textuelles. Il a montré ses performances en classification supervisée sur la tâche 1. Il se montre un peu moins performant sur la tâche 2, bien que présentant des résultats tout à fait honorables. Ses capacités intégrées d'analyse non supervisée en font un outil globalement très performant et utile pour de nombreuses tâches. La tâche 3 a été l'occasion d'expérimenter une métrique document/document basée sur une métrique individuelle mot/mot. Ces premiers essais nous paraissent intéressants et demandent à être confirmés sur d'autres types de données, particulièrement sur des documents possédant peu de mots communs. Finalement, la tâche 3 s'est avérée être presque un cas d'usage que nous allons exploiter dans le cadre du « AI Mashup Challenge ».

Remerciements

Nous remercions Marc Boullé pour ses conseils sur les tâches de classification ainsi que pour son aide avec Khiops.

Références

BELAUNDE, M., PINSON F., COLLIN, O. (2013). Cooking Assistant mashup. *AI Mashup Challenge (ESCW 2013),* Accepted

BOULLÉ, M. (2007). Compression-Based Averaging of Selective Naive Bayes Classifiers. *Journal of Machine Learning Research,* 8:1659-1685.

BOULLÉ, M. (2008). Khiops: outil de préparation et modélisation des données pour la fouille des grandes bases de données. *In Extraction et gestion des connaissances, EGC'2008,* Sophia-Antipolis, France

BOULLÉ, M. (2011). Data grid models for preparation and modeling in supervised learning. *In Hands-On Pattern Recognition: Challenges in Machine Learning, volume 1, I. Guyon, G. Cawley, G. Dror, A. Saffari (eds.),* pages 99-130*,* Microtome Publishing.

CILIBRASI, R., VITANYI, P. (2006). Similarity of Objects and the Meaning of Words. *In Proceedings of the Third international conference on Theory and Applications of Models of Computation, TAMC'06*, Beijing, China, pages 21-45.

CILIBRASI, R., VITANYI, P. (2007). The Google similarity distance. *In IEEE Transactions on Knowledge and Data Engineering,* 19(3), 370–383 (2007). Preliminary version: "Automatic Meaning Discovery Using Google", Arxiv preprint cs.CL/0412098, 2004, arxiv.org

CILIBRASI, R., BALBACH, F. J., VITANYI, P., LI, M. (2009). Normalized Information Distance *Information Theory and Statistical Learning (*Franck Emmert-Streib, Matthias Dehmer) ISBN: 978-0-387-84815-0 (Print) 978-0-387-84816-7 (Online), pages 45-82, Chapitre 3

HEINECKE, J., SMITS, G., CHARDENON, C., GUIMIER DE NEEF, E., MAILLEBUAU, E., BOUALEM, M. (2008). TiLT : plate-forme pour le traitement automatique des langues naturelles. *Traitement automatique des langues ,* 49(2):17-41.

NIGAM, K., LAFFERTY, J., MCCALLUM, A. (1999). Using Maximum Entropy for text classification. *In Proceedings of the IJCAI Workshop on Information Filtering*, pages 61-67

TSURUOKA, Y., TSUJII, J., ANANIADOU, S. (2009). Stochastic Gradient Descent Training for L1-regularized Log-linear Models with Cumulative Penalty. *In Proceedings of ACL-IJCNLP*, pages 477-485

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, oulalajamaisjepourrai-jamaismangertoutçajenaidéjàpleinlesdoigts...*".

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.

Index des auteurs

Index des mots-clés