

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 : analyse 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 comme 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 modificateurs, plus l'action qu'il dénote est complexe. Bien entendu, un modificateur 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 modificateurs 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 modificateurs 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, & SHAFLIK, 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. $\text{Recipe}(x) \ \& \ \text{Contains}(x, \text{butter}) \ \& \ \text{Has_difficult_preds}(x, y) \ \& \ y > 10 \rightarrow \text{DIFF}(x)$
2. $\text{Recipe}(x) \ \& \ \text{Contains}(x, \text{butter}) \ \& \ \text{Has_difficult_preds}(x, y) \ \& \ y < 10 \rightarrow \text{EASY}(x)$

Le système calcule les probabilités de 1 et 2 sur un corpus de référence contenant les prédicats cibles $\text{DIFF}(x)$ et $\text{EASY}(x)$. L'intérêt de la pondération de prédicats vient du fait qu'un système, suivant l'approche de (RIEDEL, 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 modificateurs $\rightarrow \text{DIFF}(r)$
2. Si r contient plus de x prédicats et contient le mot « long » ou contient le mot « difficile » $\rightarrow \text{DIFF}(r)$
3. Si le ratio de mots à polarité « facile » aux mots de polarité « difficile » $> y \rightarrow \text{EASY}(r)$
4. Si le ratio de mots à polarité « difficile » aux mots de polarité « facile » $> y \rightarrow \text{DIFF}(r)$
5. Si le nombre d'ingrédients identifiés $> x$ et le nombre de prédicats $> y \rightarrow \text{DIFF}(x)$
6. Si l'ingrédient i est détecté et que r contient au moins un mot à polarité « difficile » $\rightarrow \text{DIFF}(x)$

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 automatique².

- 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[eéèèè] frais	INGREDIENT
citron	INGREDIENT
p[aââà]te feuillet[eéèèè]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/tokensregexp.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). Chapman & Hall/CRC.
- KARLGRÉN, 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.
- RIEDEL, 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>