Introduction

Ce projet a pour objectif d’écrire un programme permettant de calculer automatiquement la complexité en temps et en espace d’une recette de cuisine. Cette complexité sera calculée à la manière de la complexité d’un algorithme : les données d’entrée sont les ingrédients, l’espace mémoire les récipients utilisés.

Pour cette tâche, nous disposons d’un corpus de 23 071 recettes au format XML. Chaque recette est structurée de la manière suivante : titre, type, niveau, coût, ingrédients (chaque ingrédient est ensuite entouré d’une balise ***p***), et enfin le texte détaillant les étapes de préparation de la recette. Ce texte n’est pas balisé, or il est nécessaire de savoir quelles opérations sont effectuées sur quels ingrédients. La première étape de notre travail consistera donc à annoter ce texte et à relier chaque ingrédient annoté à l’opération qui lui est associée (en général le verbe qui régit le syntagme désignant l’ingrédient). Concernant l’annotation des ingrédients, nous évaluerons la précision et le rappel de notre fonction à l’aide de la liste des ingrédients en début de recettes. Cette tâche nous semble cependant très difficile dans la mesure où il faudra gérer les coréférences. De plus le rappel sera difficilement calculable puisqu’un même ingrédient, n’apparaissant qu’une fois dans la liste, peut apparaître à de multiples reprises dans le texte de la préparation et rien ne nous permet de connaître ce nombre d’apparitions à partir de la liste d’ingrédients seule.

Pour calculer la complexité, nous avons besoin de connaître les quantités des ingrédients pour en déduire les proportions. Or, étant donné que le texte de la recette ne précise que très rarement la quantité des ingrédients qu’il mentionne, il faudra également relier les ingrédients mentionnés dans le texte à la liste des ingrédients donnée au début de la recette. Comme pour l’évaluation, l’un des principaux problèmes pouvant se poser est celui des coréférences. En effet, plusieurs ingrédients peuvent être regroupés sous une seule appellation générique (par exemple, « 4 fraises » et « 5 pommes » dans la liste des ingrédients peuvent être désignés par l’unique syntagme « les fruits » dans les étapes de préparation), et il peut être nécessaire de savoir à quel ingrédient un pronom renvoie. De fait, cette tâche est proche de celle de l’évaluation dans sa mise en œuvre et nous nous demandons si effectuer cette évaluation (dont les résultats risquent de n’être pas probants) ne serait pas en un sens une perte de temps.

Pour nous aider dans ces premières tâches, nous constituerons un lexique d’ingrédients classés par catégories et utiliserons un étiqueteur morphosyntaxique (Spacy). Cependant la tâche de l’étiqueteur sera compliquée par les nombreuses coquilles que nous avons pu constater dans le corpus (espaces manquants, lettres en trop, fautes d’orthographe...), ce qui risque de générer des erreurs d’étiquetage et donc de limiter la qualité de nos résultats par la suite.

Concernant le calcul de la complexité en espace, la principale difficulté à laquelle il nous faudra trouver une solution réside dans le fait que la plupart des récipients ne sont pas explicitement mentionnés dans le texte de la recette mais seulement implicitement induits par la nature des ingrédients (de la crème sera stockée dans un pot avant d’être utilisée) ou le sens des verbes désignant les opérations effectués sur les ingrédients (mélanger nécessite un contenant comme par exemple un saladier). Il nous faudra donc produire manuellement une liste des ingrédients et opérations concernées.

Une fois la complexité en temps et en espace calculé pour l’ensemble des recettes du corpus, nous analyserons la corrélation entre cette complexité, d’une part, et, d’autre part, le niveau de difficulté de la recette. Il sera intéressant de voir si cette corrélation est forte ou si la difficulté subjective d’une recette en appelle à d’autres critères.

État de l’art

L’extraction automatique d’ingrédients dans les recettes a fait l’objet de plusieurs travaux que nous avons mentionnés dans la bibliographie. Nous nous sommes particulièrement intéressés à l’article de T. Hamon et N. Grabar, *Extraction of ingredient names from recipes by combining linguistic annotations and CRF selection* ainsi qu’à celuide R. Agarwal & K. Miller, *Information Extraction from Recipes*, dont le corpus est comme le nôtre semi-structuré en XML, nous a également fourni quelques pistes pour notre travail.

Hamon et Grabar traitent comme nous des données en français, ce qui est particulièrement intéressant pour nous, et leur travail est divisé en deux étapes. La première étape est un travail de reconnaissance d’entités nommées et utilise pour cela de nombreuses ressources lexicales, non seulement pour les ingrédients mais aussi pour d’autres entités propres aux recettes de cuisine (ustensiles, actions quantités, durées...) que les auteurs ont ensuite projetées sur leur corpus à l’aide d’un outil (le module Perl Alvis ::TermTagger). Les sites web ayant servi à constituer leurs ressources sont mentionnés dans l’article et nous avons pu nous en servir pour constituer notre lexique d’ingrédients que nous avons scrappé sur le site http://les.calories.free.fr/. Les auteurs ont ensuite utilisé un système à base de règles puis un classifieur automatique de type CRF pour distinguer les ingrédients les plus importants.

Agarwal et Miller travaillent quant à eux sur un corpus semi-structuré en XML, dont la structure est très similaire à celle de notre corpus. Leur démarche consiste à générer automatiquement une représentation de recettes interprétable par une machine en réduisant chaque étape de préparation à un prédicat (action) à un à deux arguments (ingrédient/patient, ustensile/agent), ce qui représente précisément ce dont nous avons besoin pour nos calculs de complexité. Leur travail est également réparti en deux étapes : une étape de reconnaissance d’entités nommées utilisant un modèle MEMM (Maximum Entropy Markov Model) puis un travail d’étiquetage en rôles sémantiques effectué à l’aide d’un algorithme à base de règles. Cette dernière étape s’appuie sur des arbres syntaxiques, ce qui nous semble en effet une solution très intéressante pour relier les actions aux ingrédients et aux ustensiles. Il est cependant fort dommage que les auteurs n’expliquent pas le fonctionnement du module qu’ils utilisent pour régler les problèmes de coréférences que nous avons évoqués en introduction.

Concernant le calcul de complexité des recettes, qui est le sujet principal de notre projet, les travaux sont plus difficiles à trouver mais il nous faut mentionner l’article de Yun-Jung Lee, Seong-Min Yoon et Hwan-Gue Cho, *Complexity and Similarity of Recipes based on Entropy Measurement*. Les auteurs y mesurent la complexité des recettes ainsi que la similarité entre recettes (il existe plusieurs autres travaux sur ce sujet, qui n’est pas celui qui nous intéresse ici), en utilisant l’entropie. Pour effectuer ce travail, ils commencent eux aussi par une extraction des ingrédients (en utilisant la liste des ingrédients et en la nettoyant de manière semi-automatique) et des actions effectuées sur eux (peu de précisions sont apportées sur cette étape). L’entropie permet de calculer le degré d’information apporté par ces éléments (si un ingrédient est utilisé dans une majorité de recettes, comme par exemple le sel, sa présence n’apporte que peu d’information) et sert à calculer deux types de complexité : une complexité de préparation basée sur la somme des entropies des ingrédients de la recette (la recette est complexe si les ingrédients sont nombreux et/ou peu communs) et une complexité de procédure basée sur la somme des entropies des verbes d’action (« cooking verbs »). Une visualisation du travail est apportée sous forme de graphes. Bien que ne correspondant pas à la méthode que nous avons prévu d’utiliser pour ce projet, cette article est très intéressant et son approche nous a semblé pertinente.