

Rapport du Projet Tutoré

Création de recettes de cuisine sur la base de technique d'apprentissage

Projet Tutoré IR 4^{ème} année
le 31 mai 2017

Années 2016-2017
Institut National des Sciences Appliquées de Toulouse

A l'attention de M. Eric Alata et M. Roberto Pasqua

ARTIGOUHA Noémy
FICAT Timothée
GRASA Guillaume
SHIN Sohun

Sommaire

Introduction	3
Une base de données solide	4
Des ingrédients dans l'ordre	5
Matrice d'affinité	6
Génération d'une liste d'ingrédients	7
Gestion des quantités	9
Mise en place d'une application	10
Pour aller plus loin	12
Conclusion	13

Introduction

Le sujet de notre projet tutoré est de générer aléatoirement des recettes de cuisine sur la base de l'apprentissage. Ce rapport retrace toutes les étapes qui nous ont permis de créer une application qui génère des recettes de cuisine. Nous présenterons donc tout d'abord la phase d'apprentissage puis l'exploitation de cet apprentissage pour générer des nouvelles recettes de cuisine. Le projet a commencé par la rédaction d'un état de l'art, dans lequel nous avons fait des recherches pour mieux appréhender l'implémentation de l'algorithme.

Parmi ces recherches, nous nous sommes beaucoup penchés sur le Machine Learning et sur les différentes méthodes d'apprentissage comme les algorithmes génétiques, les réseaux de neurones ou encore les réseaux bayésiens. Nous avons en effet pensé qu'il était primordial d'apporter un intérêt fort à ces nouvelles méthodes car elles constituent une partie importante de notre projet.

Nous avons de plus recherché si des projets similaires avaient déjà été implémentés, comme c'est le cas avec l'intelligence artificielle d'IBM qui était capable de générer des recettes de cuisine respectant les critères donnés par l'utilisateur. Cela nous a conduit à la dernière partie de notre état de l'art : les perspectives de notre projet. L'intitulé du sujet nous laisse en effet la liberté d'implémenter différentes fonctionnalités qui pourraient être associées à la génération de recettes, comme par exemple, prendre en compte la météo du jour, l'humeur de l'utilisateur, la création de recettes originales aux associations d'aliments peu conventionnelles, etc...

Une fois nos recherches complétées, nous nous sommes lancés dans l'implémentation de notre programme.

Une base de données solide

La phase d'apprentissage est primordiale car elle va apprendre à notre logiciel comment s'agence les ingrédients entre eux et quels sont les liens entre les divers produits. Pour que cet apprentissage ait du sens il faut mettre à disposition de notre logiciel un très grand nombre de recettes. En effet, si l'échantillon à analyser est trop petit, les données analysées ne peuvent pas être considérées comme exploitables.

Nous avons choisi le site Marmitton.org et en avons extrait tout d'abord mille recettes afin de pouvoir tester rapidement les autres parties de notre logiciel. Nous avons défini une liste de mots clés à rechercher sur le site et avons accédé aux pages renvoyées par la recherche.

Afin d'extraire des recettes du site, nous avons créé un script bash faisant des requêtes sur Marmitton.org. Ensuite, ce script transforme la recette en une liste d'ingrédients et en un paragraphe décrivant les étapes de la recette grâce à l'analyse des balises HTML de la page de la recette. Ceci est finalement enregistré dans un format XML.

Afin que l'apprentissage soit encore plus conséquent nous avons finalement récupéré de cette manière plus de quinze-mille recettes. Un aussi grand nombre de requêtes nous a valu plusieurs fois l'exclusion du site Marmitton.org ralentissant seulement un peu notre travail.

Nous avons ensuite normalisé les recettes en enlevant les accents ou les caractères spéciaux. Nous avons établi une liste d'ingrédients comprenant plus de mille ingrédients afin de vérifier que les ingrédients analysés existent bien et d'être sûr qu'un même ingrédient est écrit de la même façon dans toutes les recettes. De cette manière notre logiciel pourra reconnaître deux ingrédients comme étant le même mais n'étant pourtant pas orthographier de la même façon. Cette liste nous permet aussi de détecter un ingrédient caché dans une phrase comme par exemple "Un cuillère à café de sucre". Ici il faut trouver l'ingrédient sucre et non le café. La liste des ingrédients est donc ordonnée et on trouvera le café en bas de la liste car il est utilisé dans des phrases décrivant d'autres ingrédients.

Des ingrédients dans l'ordre

Afin de procéder à une première analyse des étapes de la recette, nous avons choisi d'ordonner les ingrédients dans leur ordre d'apparition de la recette. Pour ce faire, un algorithme simple parcourt les étapes et réordonne la liste des ingrédients. Si certains ingrédients n'apparaissent pas dans les étapes, ils sont replacés à la fin et dans leur ordre d'apparition dans la liste d'ingrédients. Cette première analyse montre qu'il est possible d'analyser aussi les étapes. Mais ceci peut s'avérer très compliqué à cause de la complexité de la langue française.

Par exemple, avec la liste d'ingrédients suivante :

- 1 aubergine
- 2 tomates
- 1 cuillerée à dessert de moutarde
- 10 cl de crème liquide
- 50 g de gruyère
- herbes de Provence

Et dans les étapes suivantes :

Couper les aubergines en tranches de 1 cm d'épaisseur, dans le sens de la longueur.
Mélanger la moutarde avec la crème dans un bol. Etalez cette préparation sur une face des aubergines.
Faites des lamelles de fromage que vous placez sur les aubergines pour recouvrir toute la surface.
Coupez les tomates en tranches de 5 mm d'épaisseur.
Mettre 1 ou 2 tranches de tomate sur les aubergines.
Saupoudrez d'herbes de Provence.

On obtiendra la liste :

- 1 aubergine
- 1 moutarde
- 10 cl crème liquide
- 2 tomates
- herbes de Provence
- 50 g gruyère

Notez que tous les mots de liaison ont disparu et que le gruyère n'étant pas cité par son nom mais comme du "fromage", celui-ci est tout de même ajouté à la fin de la liste.

Matrice d'affinité

Une fois la base de données récupérée, ainsi que l'ordre des ingrédients d'une recette gérée, nous avons créé une matrice pour représenter l'affinité entre les ingrédients. En effet, nous avons parsé toutes les recettes de notre base de données pour récupérer le nombre de fois où 2 aliments apparaissent dans une même recette. Pour ce faire, nous avons comparé chaque aliment de toutes les recettes deux à deux. A chaque fois que deux aliments étaient dans la même recette nous ajoutons à notre matrice une affinité entre ces deux aliments.

Exemple avec les trois recettes suivantes :

<u>Recette 1</u>	<u>Recette 2</u>	<u>Recette 3</u>
Beurre	Beurre	Farine
Farine	Chocolat	Sucre
Lait	Farine	Beurre

Notre matrice serait alors :

	Beurre	Farine	Lait	Chocolat	Sucre
Beurre	0	2	1	1	0
Farine	1	0	1	0	1
Lait	0	0	0	0	0
Chocolat	0	1	0	0	0
Sucre	1	0	0	0	0

Nous constatons que parmi ces trois recettes, le beurre et la farine sont, par exemple, apparus dans deux recettes. De plus, notre matrice n'est pas symétrique pour pouvoir prendre en compte l'ordre dans lequel sont ajoutés les aliments dans une recette.

La diagonale est donc remplie de 0 puisqu'on ne peut pas trouver dans une même recette deux fois le même ingrédient dans la liste d'ingrédients (ou alors ce serait une erreur de la part de celui qui a écrit la recette). Notre matrice est créée dans un fichier csv. Dans ce fichier csv, les noms des ingrédients ne sont pas présents, nous avons créé en parallèle, une hashmap qui associe un ingrédient à un numéro.

Ce numéro correspond à l'indice où se trouve l'ingrédient dans la matrice. Cette matrice nous a servi pour générer nos recettes. Elle a subi quelques modifications au fil du temps quand nous avons décidé d'ajouter les quantités. Maintenant, chaque case de la matrice possède l'affinité entre deux ingrédients, la moyenne et l'écart type du rapport de quantités entre ces deux ingrédients.

Génération d'une liste d'ingrédients

1- Fonction de génération :

Une fois la matrice d'associations créée, nous sommes dédiés à l'implémentation d'un algorithme permettant de générer une liste d'ingrédients cohérente. L'utilisateur fournit le nombre d'ingrédients de la recette qu'il veut créer ainsi qu'une liste d'ingrédients. Cette même liste d'ingrédient fournie par l'utilisateur contient les ingrédients qu'il souhaite avoir dans la recette. Il peut cependant fournir une liste vide et ainsi le premier ingrédient sera choisi de manière aléatoire parmi tous les ingrédients appartenant à notre base de données.

Afin de compléter la liste d'ingrédient, des ingrédients ont dû être sélectionnés un par un en suivant le processus suivant :

- Chaque ingrédient de notre base de données passe par une fonction de validité qui vérifie que l'ingrédient ait au moins une affinité avec au moins 80% des ingrédients déjà présent dans la liste, ce qui permet de garder une cohérence au niveau des ingrédients mais laisse les possibilités ouvertes.
- On calcule ensuite le score de ces ingrédients valides en additionnant les affinités qu'à l'ingrédient avec tous les ingrédients de notre liste. Ensuite, nous ne sélectionnons que les n-meilleurs (n étant prédéfini par le programmeur).
- Ensuite, on prend un ingrédient au hasard dans cette liste et l'ajoute à la liste d'ingrédients de la recette.

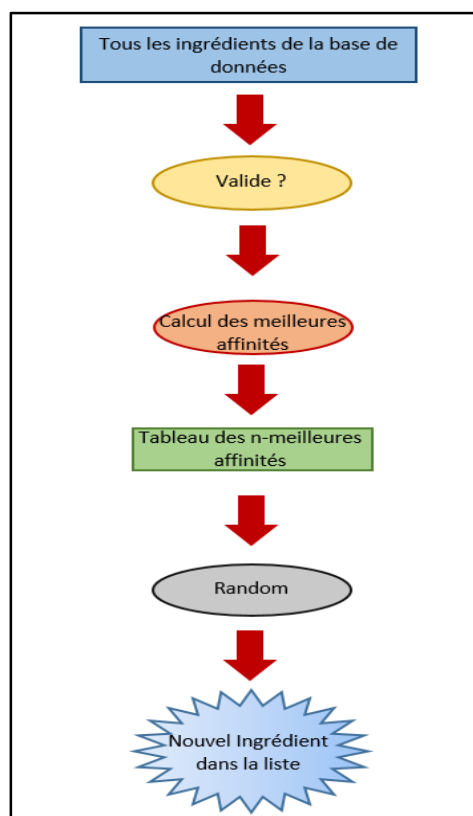


Schéma du fonctionnement de l'algorithme

2- Fonction inverse à la génération :

Afin de vérifier que notre implémentation respecte bien la logique des recettes, nous avons écrit une fonction permettant de retrouver la liste d'ingrédient d'une recette donnée. L'utilisateur fournit les ingrédients de la recette. L'algorithme va prendre le premier ingrédient de la liste et vérifier que l'ingrédient suivant dans la liste fournie apparaît bien dans le tableau des meilleures affinités. Certaines recettes nécessitent une taille de tableau plus élevée que d'autre (cela dépend de la rareté des associations), mais malgré cela on retrouve bien la logique des recettes.

Gestion des quantités

Une fois l'implémentation de notre générateur d'ingrédients terminée, nous avons décidé de l'améliorer afin d'y ajouter les quantités. Pour cela nous nous sommes servis des moyennes et écarts-types présents dans notre matrice d'association. En effet, ces deux paramètres allés nous être très utile pour la détermination de lois normales. Pour le calcul de quantité nous avons trouvé judicieux d'employer les lois normales car elles permettent de récupérer les rapports de quantités appartenant à un certain intervalle et d'éliminer ceux trop éloignés des bornes fixées. Après concertation, nous avons jugé qu'il était intéressant de garder environs 80% des valeurs, car cela permettait de garder une certaine variabilité au niveau des recettes.

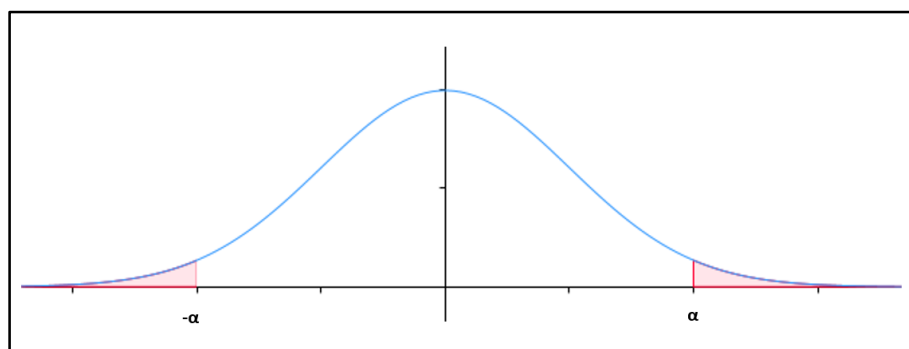


Schéma d'une loi normale et de l'intervalle $[-\sigma ; \sigma]$

Pour déterminer les valeurs de notre intervalle nous avons dû résoudre cette équation :

$$\begin{aligned} & 1 - 2 \cdot P(Z > \alpha) = 80\% \quad (Z \text{ suit une loi Normale } N(0,1)) \\ \Rightarrow & P(Z > \alpha) = 10\% \\ \Rightarrow & \alpha = 1,645 \\ \text{or} & Z = (Y - m) / \sigma \quad (Y \text{ suit une loi Normale } N(m, \sigma)) \\ \Rightarrow & P(Y > (\beta - m) / \sigma) \\ \Rightarrow & (\beta - m) / \sigma = \alpha \\ \Rightarrow & \beta = \alpha \cdot \sigma + m \Rightarrow \beta = 1,645 \cdot \sigma + m \end{aligned}$$

Résolution de l'équation pour trouver σ et β

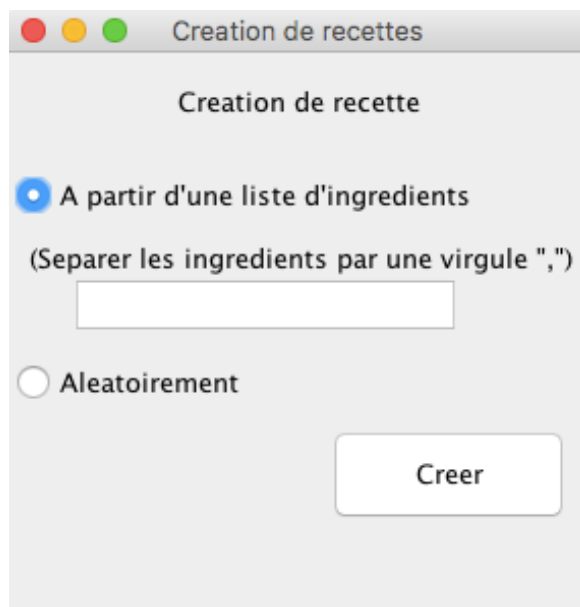
On peut alors prendre un rapport aléatoirement dans l'intervalle $[m - \beta ; m + \beta]$. La quantité de l'ingrédient n est déterminée avec le rapport de quantité entre lui-même et le premier ingrédient de la liste. Dans le cas où leur affinité est de 0 (car l'ingrédient ne peut matcher qu'avec un minimum de 80% des ingrédients de la liste), sa quantité est calculée avec le prochain ingrédient de la liste et ainsi de suite jusqu'à trouver un ingrédient de la liste avec qui il partage des affinités.

Reste alors à se poser la question de comment déterminer la quantité du premier ingrédient de la liste ? En effet, celui-ci étant seul au début, il n'est pas possible de sélectionner un rapport avec un autre ingrédient. Pour parer ce problème, la quantité du premier élément est déterminée de manière aléatoire à partir d'intervalles que nous avons définis de manière arbitraire. Ces mêmes intervalles dépendent aussi du type de la quantité (pièce, litre, gramme).

Mise en place d'une application

Notre idée a été de pouvoir créer de nouvelles recettes de cuisine soit à partir d'un ingrédient que l'on choisit aléatoirement soit à partir d'une liste d'ingrédients que l'utilisateur fournit.

Nous avons donc mis en place une petite interface graphique où l'utilisateur choisit dans un premier temps s'il souhaite fournir une liste d'ingrédients ou s'il souhaite générer une recette à partir de n'importe quel ingrédient.



Fenêtre principale

Dans le cas où l'utilisateur fournit une liste d'ingrédients, l'interface graphique récupère cette liste et l'envoie à la classe qui s'occupe de générer une recette afin que le choix du reste des ingrédients soit cohérent avec les ingrédients demandés par l'utilisateur. Afin que les ingrédients soient bien reconnus par notre application, nous utilisons le normaliseur utilisé lors de la récupération des recettes de cuisine. Grâce à ce passage, même si l'utilisateur se trompe dans l'orthographe de l'ingrédient, il sera reconnu correctement. Par exemple, « tomat » sera compris comme « tomate ».

Si l'utilisateur choisit une génération de recette complètement aléatoire, alors un ingrédient sera choisi au hasard dans notre base de données et trouvera les ingrédients suivants en fonction du premier. Une fois que l'utilisateur appuie sur le bouton "Créer", une nouvelle fenêtre avec une recette apparaît.



Fenêtre des recettes

Comme vous pouvez le voir sur l'image ci-dessus, l'utilisateur peut choisir de générer plus de recettes, enregistrer la recette courante (sous format .txt) ou si la recette ne lui plaît pas, il peut fermer l'onglet.

Pour aller plus loin

1- Les étapes de la recette :

Notre logiciel ne génère finalement qu'une liste d'ingrédients avec la quantité de chaque ingrédient. Or, nous avons vu qu'une recette est aussi composée d'une liste d'étapes afin de la réaliser.

Nous n'avons pas pu gérer ceci dans les temps impartis mais nous avons réfléchi longuement à sa réalisation. Notre logiciel aurait dû analyser les étapes des recettes et comprendre comment son manipuler les ingrédients, leur cuisson ou leur utilisation. Mais la langue française étant très complexe nous n'avons pas trouvé de moyens fiables pour analyser les étapes.

Si nous avions réussi nous aurions enregistré ces informations sous forme de motifs pour ensuite les combiner entre eux et les restituer sous forme de phrase en français. Voici deux exemples de motifs montrant la manipulation d'ingrédients :

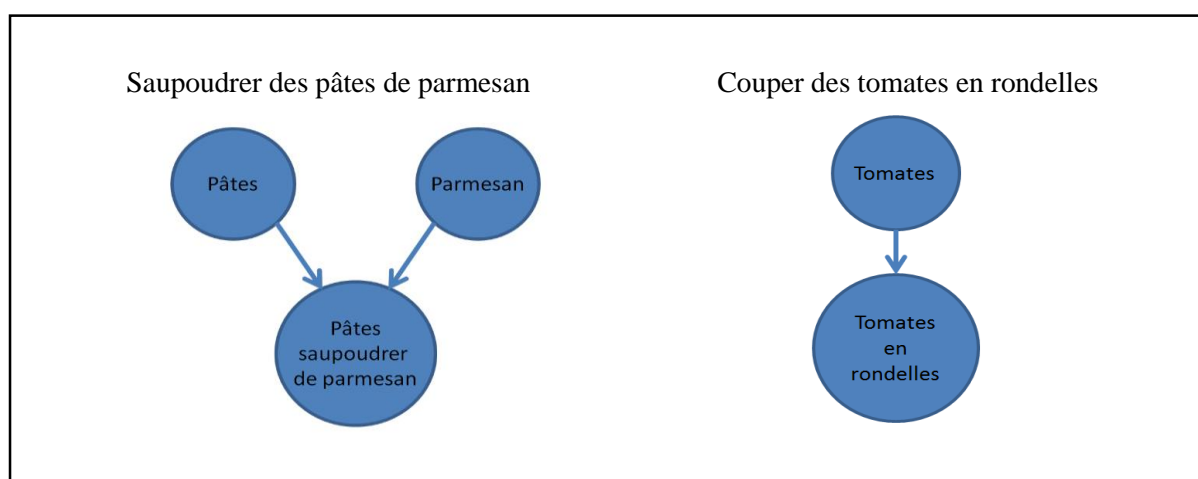


Schéma de la création de motifs

2- Une application avec plus de choix :

Comme dit en introduction, nous avons pensé à créer une application qui pose des questions à l'utilisateur pour orienter son choix de génération de recette. Dans le cas où l'utilisateur a passé une mauvaise journée, il voudrait peut-être quelque chose de doux et sucré pour le réconforter par exemple. Mais également, s'il a fait chaud, un plat rafraichissant serait un met agréable. L'utilisateur pourrait aussi avoir la possibilité de choisir la génération d'une recette en fonction d'une nationalité choisie.

Conclusion

Nous sommes satisfaits de ce que nous avons accompli, même s'il reste de nombreuses améliorations à faire. Nous nous sommes rapidement aperçu qu'il était compliqué de générer un texte explicatif pour réaliser nos recettes. En effet, la création de motifs est quelque chose d'assez complexe et nous avons passé énormément de temps sur la gestion des quantités pour avoir le temps de passer à la génération d'étapes. Ce fut néanmoins un projet réellement enrichissant car il nous a permis d'en apprendre plus sur les méthodes d'apprentissage de l'intelligence artificielle et sur le monde de la cuisine mais également car nous étions libres pour la conception du projet.

Par ailleurs, travailler en équipe a permis à tous les membres du groupe, de s'améliorer en communication et gestion de projet. Au fur et à mesure que le projet avançait, nous sommes passés d'une communication brouillon à une réelle organisation, avec des réunions périodiques et des comptes rendus de séance permettant le suivi des travaux de chaque membre du groupe. En plus, il a fallu apprendre à s'adapter aux comportements et aux méthodes de travail de tout à chacun, savoir formuler des critiques constructives et faire passer des messages importants sans vexer quiconque.

Enfin, nous avons développé une méthode de travail en autonomie, nous efforçant de mettre au point une première version le plus rapidement possible afin d'avoir l'avis de nos tuteurs et de vérifier que notre vision du projet correspondait bien à la leur. Par la suite, à chaque rencontre avec l'un d'entre eux, nous leur présentions une version améliorée du projet, que ce soit avec la correction de bugs ou l'ajout de nouvelles fonctionnalités.