

# **Institut National des Langues et Civilisations Orientales**

Département Textes, Informatique, Multilinguisme

---

## **ProjeTambouille**

---

**MASTER  
TRAITEMENT AUTOMATIQUE DES LANGUES**

*Parcours Ingénierie Multilingue*

**Martin Digard et Anaëlle Pierredon**

Année universitaire 2020/2021

# TABLES DES MATIÈRES

<b>TABLES DES MATIÈRES</b>	<b>2</b>
<b>INTRODUCTION</b>	<b>3</b>
Présentation générale	3
<b>ÉTAT DE L'ART</b>	<b>5</b>
1.1 Introduction	5
1.2 Contenu	5
1.3 Conclusion	7
<b>MÉTHODES</b>	<b>9</b>
2.1 Introduction	9
2.2 Contenu	9
2.3 Conclusion	11
<b>CORPUS</b>	<b>14</b>
2.1 Introduction	14
2.2 Contenu	15
2.3 Conclusion	16
<b>RÉSULTATS</b>	<b>17</b>
2.1 Introduction	17
2.2 Contenu	17
2.3 Conclusion	19
<b>DISCUSSION</b>	<b>20</b>
2.1 Introduction	20
2.2 Contenu	20
2.3 Conclusion	21
<b>CONCLUSION GÉNÉRALE</b>	<b>22</b>
<b>BIBLIOGRAPHIE</b>	<b>23</b>
Analyse automatique de recettes de cuisine	23
Complexité	23
<b>BONUS</b>	<b>24</b>
Extrait de conversation visant à démontrer l'impact positif de la programmation récursive sur le cerveau humain.	24

# INTRODUCTION

## Présentation générale

L'objectif du projet est de s'inspirer du modèle de calcul de complexité en temps et en espace des programmes informatiques et de l'appliquer à des recettes de cuisine. Le programme réalisé utilise un corpus de recettes de cuisine et calcule pour chacune d'elles sa complexité en temps et en espace, en utilisant au maximum le style de programmation récursif.

Nous allons dans un premier temps présenter le contexte général puis dans un second temps les expérimentations réalisées.

# Première partie

## Contexte général

# ÉTAT DE L'ART

## 1.1 Introduction

Afin de mener à bien notre projet, nous nous sommes renseignés sur les travaux antérieurs qui ont été réalisés à propos de l'analyse automatique de recettes.

Le DEFT<sup>1</sup> (Défi Fouille de Textes) est une conférence d'évaluation qui a lieu chaque année depuis 2005 sur différents thèmes. Son intérêt est de permettre de confronter le travail de plusieurs équipes sur un même corpus mais avec des méthodes et des outils différents. Les thèmes relèvent de la fouille de textes et concernent toujours la langue française. Le thème de l'année 2013 était l'analyse automatique de recettes et les articles publiés à cette occasion vont nous servir de base pour nos expériences.

Nous allons donc présenter dans cette partie le DEFT 2013 ainsi que les articles qui nous ont aiguillés dans le traitement de notre corpus.

## 1.2 Contenu

Le DEFT 2013 consistait en quatre tâches :

- l'identification du niveau de difficulté de réalisation d'une recette à partir du titre et du texte de la recette, sur une échelle à quatre niveaux (très facile, facile, moyennement difficile et difficile).
- l'identification du type de plat préparé (entrée, plat principal, dessert) à partir du titre et du texte de la recette.
- l'appariement d'une recette avec son titre.
- l'identification des ingrédients d'une recette à partir de son titre et de son contenu.

Les tâches qui pourraient nous intéresser dans notre travail sont les tâches 1 et 4. Les participants travaillent sur le même corpus que nous et, bien qu'ils utilisent des méthodes différentes (apprentissage automatique), ils peuvent rencontrer les mêmes problématiques que nous liées aux tâches et au corpus. Nous allons donc nous intéresser aux traitements proposés par les différentes équipes dans ces deux tâches.

Pour l'identification des ingrédients d'une recette (tâche 4), les participants s'étaient vus fournir une liste globale contenant tous les ingrédients à trouver dans les recettes.

---

<sup>1</sup> <https://deft.limsi.fr/>

Les principales difficultés rencontrées lors de cette tâche concernent la correspondance entre un ingrédient et la forme sous laquelle il apparaît. En effet, les ingrédients ne sont pas toujours précisés dans les recettes : « mélanger énergiquement tous les ingrédients » (recette 27174) et les ingrédients de l'entête ne sont pas tous utilisés ou alors des ingrédients non listés sont ajoutés dans le corps de la recette (Bost et al, 2013). Certains ingrédients peuvent être implicites (ex : la liste d'ingrédients contient « huile d'olives » et le lecteur doit déduire qu'il doit l'utiliser pour « faire revenir les oignons ») et contenir des fautes d'orthographe (Dini et al, 2013). Les ingrédients peuvent être remplacés par des formes verbales (*saler* au lieu de *sel*) ou nominales (*légume* au lieu de *carotte* ou *viande* au lieu de *escalope de poulet*)(Charton et al, 2013). D'après les analyses de ces participants, un ingrédient à extraire n'est pas mentionné explicitement dans l'entête dans 39,8% des cas.

D'autres difficultés ont été évoquées et elles concernent notamment la catégorisation des éléments. En effet, certains participants ont voulu différencier les ingrédients (« brut ») et les aliments (déjà cuisinés). Cependant, certains aliments comme les pâtes ou les glaces sont aussi des ingrédients (Hamon et al, 2013).

Afin de contourner ces difficultés, les participants ont pu enrichir les ingrédients de leur liste avec des synonymes (ex : l'ingrédient *tarte* est enrichi avec *gâteau*, *pâtisserie*, *tartelette*, *gaufre*, *cake*, *galette*, *crêpe* et *beignet*) mais cela détériore malheureusement leurs résultats (Hamon et al, 2013). Certains ont pris en compte tous les accents possibles (ex : carr[eééèê] frais/ p[aââà]te feuillet[eéèèê]) et ont utilisé les formes non-contractées (ex : *crème chantilly* pour *chantilly*)(Dini et al, 2013). D'autres ont substitué à certaines formes nominales ou verbales les ingrédients correspondants (ex : « beurrée|beurrez|beurrer » sont remplacés par *beurre*) à l'aide d'une trentaine d'expressions régulières (Charton et al, 2013).

Enfin, un groupe s'est interrogé sur la bonne manière de quantifier les ingrédients selon leur unité. Pour pouvoir les traiter de la même manière ils ont converti toutes les quantités en grammes ou en litres à l'aide de convertisseurs disponibles en ligne (ex : *soupçon*, *chouia* et *pincée* deviennent 1 gramme de produit, et *louche*, *poignée* deviennent 100 gramme de produit)(Hamon et al, 2013).

Les principales difficultés rencontrées lors de la tâche d'identification du niveau de difficulté (tâche 1) sont la répartition inégale entre les niveaux : la plupart sont *Très facile* ou *Facile* et la subjectivité du niveau (déterminé par l'auteur).

Pour contourner ces difficultés, certains participants proposent de commencer par classer les recettes en deux catégories : *facile* ou *difficile* puis de classer de nouveaux ces sous-groupes en deux catégories : *Très facile* ou *Facile* et *Moyennement difficile* ou *difficile* (Bost et al, 2013). D'autres proposent d'attribuer l'étiquette *Facile* par défaut (catégorie la plus présente) et de la modifier selon les probabilités qu'elles appartiennent à une autre étiquette. Pour calculer ces probabilités, ils comparent la recette à étiqueter avec les recettes dont la complexité est connue (Lejeune et al, 2013).

Plusieurs groupes se sont basés sur le nombre de verbes pour évaluer la complexité de la recette. Mais il n'est pas correct de simplement considérer que plus il y a de verbes, plus la recette est complexe, car il y a des verbes qui n'ont pas de rapport avec la recette en elle-même (ex : un auteur qui raconte l'origine de la recette...). De plus, certains verbes sont intrinsèquement faciles comme *faire chauffer* ou *faire bouillir* mais d'autres nécessitent plus

d'expérience comme *flamber* (Dini et al, 2013). Une des solutions proposée est de classer les verbes en différentes catégories selon leur niveau de difficulté (Dini et al, 2013 et Charton et al, 2013). On pourrait également considérer que plus un verbe a de modifieurs, plus l'action qu'il dénote est complexe (ex : *mélangez le tout* < *mélangez longuement le tout* < *mélangez longuement le tout avec attention*) et compléter avec un petit lexique de certains mots qui dénotent une polarité difficile ou facile (Dini et al, 2013).

## 1.3 Conclusion

Étant donné que nous ne disposons pas de cette liste regroupant tous les ingrédients et devant la difficulté de repérer tous les ingrédients d'une recette démontrée par les différents articles présentés précédemment, nous avons décidé de nous baser sur la liste des ingrédients de l'entête et d'ignorer les ingrédients n'étant pas présent dans cette liste ou n'étant pas explicitement énoncés (ex: « découpez les légumes » ne sera pas annoté). Nous avons préféré nous concentrer sur la reconnaissance des ingrédients qui sont explicitement précisés.

Avec cette manière de faire, nous espérons éviter le problème des fautes d'orthographe, l'entête et le corps de la recette étant écrit par la même personne.

Concernant la quantification des ingrédients, nous nous sommes interrogés sur la nécessité de convertir chaque unité en gramme ou en litre (Hamon et al, 2013). Nous avons en premier lieu hésité entre considérer les mots tels que *tranches* ou *feuilles* comme des ingrédients ou comme des quantifieurs. Dans le syntagme « tranche de pain », l'ingrédient est-il « pain » ou « tranche de pain » ?

On peut par exemple faire le rapprochement entre 2 *tranches* de pain et 100 *grammes* de farine. En considérant *tranche* comme un quantifieur, l'ingrédient se réduirait à *pain*, au lieu de n'avoir qu'un ingrédient *tranche de pain* sans quantifieur nécessaire pour le décompte.

Nous avons finalement décidé de considérer pour seuls quantifieurs les mots de mesure (*g*, *kilo*, etc.) et tous les mots comme *tranche de* de la même manière que l'on pense *escalope de*, c'est-à-dire comme un ingrédient. Ainsi, dans 3 *tranches de pain*, 3 est le nombre d'ingrédients et dans « 3g de farine », 3 est la quantité de farine. Nous n'avons gardé que les unités qui correspondent à un volume et non à un compteur.

Lorsque l'ingrédient peut être compté (ex : œuf), il représente lui-même sa propre unité de compte, mais lorsqu'il ne peut pas être compté (ex : *farine*), des unités de mesure sont indispensables à moins de ne compter qu'un ingrédient *farine*. La reconnaissance de mots comme *escalope de*, *tranche de*, *filet de* comme ingrédient par l'algorithme nous a semblé plus fiable. De plus, cela permet de bien différencier les ingrédients des unités de mesure.

Nous avons donc adopté la stratégie inverse de celle présentée dans « Efficacité combinée du flou et de l'exact des recettes de cuisine » (Hamon et al, 2013) : au lieu de convertir chaque quantité d'ingrédient en gramme ou en litre, nous avons voulu considérer chaque ingrédient comme une unité. Si l'ingrédient peut être compté alors sa quantité est donnée directement dans la recette (ex : 3 œufs) et s'il nécessite une unité alors sa quantité est égale à 1 (ex: 300g de viande = 1 viande).

Pour l'estimation du niveau de la recette, notre façon de faire sera assez différente de celles présentées par les articles. En effet, nous nous basons sur des calculs de complexité en temps et en espace pour estimer la difficulté d'une recette. Nous pouvons cependant rapprocher les deux méthodes car nous nous basons sur les opérations d'une recette et

celles-ci correspondent plus ou moins à un verbe. Mais au lieu de leur associer un niveau de difficulté (Dini et al, 2013 ou Charton et al, 2013), nous leur avons associé un temps d'exécution et un nombre de récipients nécessaires pour tenter de représenter leur complexité.



## MÉTHODES

### 2.1 Introduction

Nous allons commencer par présenter notre méthode de manière théorique puis nous allons expliciter son application en présentant l'arborescence des fichiers, les modules et l'organisation de notre code.

Étant donné la taille du corpus sur lequel nous travaillons, nous avons commencé par réaliser nos traitements sur un petit nombre de recettes que nous avons progressivement augmenté. Cette façon de faire nous a permis d'observer nos résultats et de régler les problèmes petit à petit. Afin de s'assurer de pouvoir accomplir l'entièreté de la chaîne de traitement, nous avons décidé de privilégier une approche naïve pour les étapes de complexité : temps = somme des temps pour chaque opération ; espace = nombre de récipients.

### 2.2 Contenu

Nous avons commencé par annoter le corpus en ingrédients, opérations et récipients. Ces annotations nous serviront plus tard à réaliser les calculs de complexité pour chaque recette. Les ingrédients de l'entête ont été utilisés pour l'annotation des ingrédients de la recette. Nous avons décidé de lemmatiser la liste des ingrédients pour éviter que *pomme de terre* ne soit pas reconnu quand il est écrit *pommes de terre* dans la recette (seulement *terre* se retrouverait alors annoté...). Cependant, certains ingrédients apparaissent uniquement dans le corps de la recette et ne sont pas précisés dans l'entête. Ces cas ne seront alors pas pris en compte par notre programme.

Pour l'annotation en opération, nous avons annoté en parties du discours avec SpaCy et repéré les opérations à partir des verbes jusqu'à ce que l'on rencontre de nouveau un verbe (sauf pour *faire* et *laisser* pour lesquels un second verbe est autorisé), ou une fin de phrase. Nous avons aussi choisi de ne prendre que les verbes à l'infinitif ou à la deuxième personne du pluriel pour limiter les erreurs. Les recettes de cuisine ne contenant pas uniquement des verbes qui dénotent des opérations, nous avons ignoré certains verbes comme « pouvez », « aurez »... (et d'autres mots comme « frais » qui étaient considérés comme des verbes par SpaCy).

Afin d'annoter en récipients, nous avons récupéré la liste d'ustensiles de Wikipédia<sup>2</sup> (Hamon et al, 2013) et n'avons gardé que ce que nous considérons comme un récipient (adieu *girafe* et *couteau sashimi*).

```

▼<recette id="73386">
  <titre>Gâteau fondant au chocolat</titre>
  <type>Dessert</type>
  <niveau>Très facile</niveau>
  <cout>Bon marché</cout>
  ▼<ingredients>
    <p>200 g de chocolat noir pâtissier</p>
    <p>200 g de beurre + une noisette pour le moule</p>
    <p>200 g de sucre glace</p>
    <p>1 cuillère à soupe de farine + 1 pour le moule</p>
    <p>5 œufs entiers</p>
  </ingredients>
  ▼<preparation>
    <![CDATA[ au <recipient>bain - marie</recipient>, <operation>faire fondre le
    <ingredient>chocolat</ingredient> et le <ingredient>beurre</ingredient></operation>.
    une fois fondus, rajouter un à un les 5 <ingredient>œufs entiers</ingredient> en
    remuant bien au fouet. incorporer ensuite le <ingredient>sucres glace</ingredient> et 1
    <ingredient>cuillère de farine</ingredient>. bien mélanger pour que la pâte soit lisse
    et homogène et <operation>répartir dans un <recipient>plat</recipient> beurré et
    fariné</operation>. enfourner 25 à 30 min à 180 ° c (thermostat 6). ]]>
  </preparation>
</recette>

```

### Exemple du fondant au chocolat à la fin de la première phase d'annotation

Une fois cette première étape d'annotation réalisée, nous avons associé une quantité aux ingrédients ainsi qu'un temps et un espace aux opérations de base en ajoutant des attributs aux balises.

Afin de compléter les annotations en ingrédients nous nous sommes servis des quantités indiquées dans l'entête : s'il n'y a aucune unité précisée (ex : 3 oeufs) nous avons récupéré la quantité directement indiquée par la recette et s'il y a une unité (ex : 20 cl de lait de coco), nous avons considéré que la quantité était 1.

Pour associer un temps aux opérations de base, nous avons annoté manuellement les 85 opérations les plus courantes dans un fichier tsv séparé, puis réalisé une moyenne de tous ces temps. De cette manière, on associe à chaque opération : le temps qu'elle contient s'il est spécifié dans le texte (ex: <operation temps=35min>Faire cuire pendant 35 minutes</operation>) ; le temps annoté manuellement s'il y en a un ; et le temps moyen (6,86 min) dans tous les autres cas.

La recette ayant été annotée en récipients, nous avons compté le nombre de récipients par recette et indiqué cette quantité dans chaque balise *opération*. Nous aurions aimé faire une estimation du nombre de récipients nécessaires par opération plutôt que d'indiquer le même nombre pour chaque opération, mais nous avons manqué de temps.

```

▼<recette id="73386">
  <titre>Gâteau fondant au chocolat</titre>
  <type>Dessert</type>
  <niveau>Très facile</niveau>
  <cout>Bon marché</cout>
  ▼<ingredients>
    <p>200 g de chocolat noir pâtissier</p>
    <p>200 g de beurre + une noisette pour le moule</p>
    <p>200 g de sucre glace</p>
    <p>1 cuillère à soupe de farine + 1 pour le moule</p>
    <p>5 œufs entiers</p>
  </ingredients>
  ▼<preparation>
    <![CDATA[ au <recipient>bain - marie</recipient>, <operation temps=6.86min
    espace=8>faire fondre le <ingredient quantite=1>chocolat</ingredient> et le <ingredient
    quantite=1>beurre</ingredient></operation>. une fois fondus, rajouter un à un les 5
    <ingredient quantite=5>œufs entiers</ingredient> en remuant bien au fouet. incorporer
    ensuite le <ingredient quantite=1>sucres glace</ingredient> et 1 <ingredient
    quantite=1>cuillère de farine</ingredient>. bien mélanger pour que la pâte soit lisse
    et homogène et <operation temps=6.86min espace=8>répartir dans un
    <recipient>plat</recipient> beurré et fariné</operation>. enfourner 25 à 30 min à 180 °
    c (thermostat 6). ]]>
  </preparation>
</recette>

```

<sup>2</sup> Ustensil

### *Exemple du fondant au chocolat à la fin de la deuxième phase d'annotation*

Nous avons ensuite réalisé une évaluation de la fonction de reconnaissance d'ingrédients en comparant les ingrédients annotés aux ingrédients indiqués dans l'entête de chaque recette. Nous avons vérifié que tous les ingrédients annotés se trouvaient dans l'entête et que tous les ingrédients de l'entête étaient annotés dans le texte de la recette. Nous avons ainsi pu connaître le nombre de vrais positifs, faux positifs et faux négatifs, ainsi que calculer la précision, le rappel et la f-mesure.

Nous sommes conscients que nos annotations étant basées sur l'entête, les résultats de l'évaluation sont légèrement biaisés. En effet, il ne devrait pas y avoir d'ingrédient annoté qui ne soit pas déjà présent dans l'entête. Avec cette méthode, nos précision, rappel et f-mesure auraient dû avoir 100%. Ce procédé n'était donc pas inintéressant car il nous a montré que notre code pouvait encore être amélioré avant de passer à une étape plus exigeante. De plus, il y a certaines recettes dont les ingrédients utilisés ne sont pas précisés dans l'entête et d'autres pour lesquelles certains ingrédients de l'entête ne sont finalement pas utilisés dans la recette. Est-il vraiment correct de considérer ces cas comme des faux positifs ou des faux négatifs ?

Nous avons également calculé le coefficient de corrélation entre le niveau (très facile, facile, moyennement difficile ou difficile) et le temps, ainsi qu'entre le niveau et l'espace. Le but de ces deux calculs est de savoir s'il est possible de prédire le niveau de difficulté d'une recette à partir de ses complexités en temps et en espace (recette (x)  $\Rightarrow$  niveau (y)).

Enfin, nous avons analysé la complexité en temps de notre programme d'annotation.

## 2.3 Conclusion

Notre code s'organise de la manière suivante :

```
Projetambouille/
├── 00_enonce.pdf
├── 0_corpus
│   ├── 0_corpus
│   ├── 1_corpus_annotate
│   └── 2_corpus_final
├── 1_annotations
│   ├── a_cuisine.py
│   ├── b_autocuisseur.py
│   ├── c_ingredients.py
│   ├── d_operations.py
│   ├── e_recipients.py
│   └── z_lexique_recipients
├── 2_complexite
│   ├── a_complexites.py
│   ├── b_espace_temps.py
│   ├── c_quantites.py
│   ├── z_infos_ingredients.tsv
│   └── z_temps_operations.tsv
├── 3_resultats
│   ├── 0_rapport_analyse_complexite
│   ├── a_calculs.py
│   ├── b_evaluation.py
│   ├── c_analyse_complexite.py
│   └── graphiques
├── cornflake
├── z_laboratoire
│   ├── calculs_moyenne_temps
│   ├── expe_iter.py
│   ├── expe_recuratif.py
│   └── factorielle.py
└── 10 directories, 20 files
```

Tous les codes ont été vérifiés avec **cornflake** qui est un petit **script bash**. Ce script prend en argument un fichier python (*fichier.py*) et lance pylint et flake8. Il peut être lancé dans le répertoire *ProjeTambouille* avec la commande **./cornflake \*.py**

Tous les répertoires contiennent un fichier python *a\_fichier.py* qui appelle tous les modules du même répertoire. Le fichier **c\_analyse\_complexite.py** du répertoire **3\_resultats/** doit néanmoins être lancé indépendamment du fichier **a\_calculs.py**. Les fichiers dont le préfixe est *z\_fichier* sont des fichiers utilisés par le programme. Certains sont générés par le code (*z\_infos\_ingredients*) et d'autres l'ont été manuellement (*z\_temps\_operations*, *z\_lexique\_recipients*).

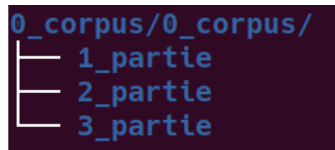
*0\_rapport\_analyse\_complexite* contient les temps d'exécution du programme d'annotation sur différentes tailles de corpus. Nous avons converti tous les temps en secondes à partir de ce fichier afin de créer les graphiques de complexité de notre programme d'annotation.

Les fonctions ou indications **reloutise(un, deux ou trois)**<sup>3</sup> sont utilisées pour debugger certaines sorties partiellement erronées.

Le répertoire **z\_laboratoire/**, contient les fichiers python que nous avons utilisés pour faire des tests de comparaisons entre récursif et itératif. Il contient aussi les scripts de calcul de la moyenne des temps de chaque opération pour le programme de complexité.

### Lancement des scripts :

#### Dans le répertoire **1\_annotations** :



```
0_corpus/0_corpus/  
├── 1_partie  
├── 2_partie  
└── 3_partie
```

```
python3 a_cuisine.py ../0_corpus/0_corpus/1_partie/ ../0_corpus/1_corpus_annotate/1_partie/  
python3 a_cuisine.py ../0_corpus/0_corpus/2_partie/ ../0_corpus/1_corpus_annotate/2_partie/  
python3 a_cuisine.py ../0_corpus/0_corpus/3_partie/ ../0_corpus/1_corpus_annotate/3_partie/
```

#### Dans le répertoire **2\_complexite** :

```
python3 a_complexites.py ../0_corpus/1_corpus_annotate/ ../0_corpus/2_corpus_final/
```

#### Dans le répertoire **3\_resultats** :

```
python3 a_calculs.py ../0_corpus/2_corpus_final/  
python3 c_analyse_complexite.py
```

---

<sup>3</sup> **Analyse morphologique du mot reloutise** pour les plus anciens  
reloutise : néologisme ⇒ lourd (très ennuyeux) ⇒ relou (en verlan)  
⇒ une relou + suffixe *tise* : un truc relou

# Deuxième partie

## Expérimentations

## CORPUS

## 2.1 Introduction

Nous disposons pour ce travail d'un corpus de 23 071 recettes provenant du site Marmiton<sup>4</sup>, et converties au format XML. Les recettes sont constituées d'une entête et du corps de la recette.

```
▼<recette id="1439">
  <titre>Un amour de cheese au Carré Frais</titre>
  <type>Dessert</type>
  <niveau>Facile</niveau>
  <cout>Bon marché</cout>
  ▼<ingredients>
    <p>1 barquette de Carré Frais au yahourt</p>
    <p>1 citron</p>
    <p>10 cl de crème fraîche entière épaisse</p>
    <p>50 g de sucre en poudre</p>
    <p>100 g de biscuits (type Petit beurre)</p>
    <p>2 oeufs</p>
    <p>30 g de beurre doux mou</p>
    <p>1 pincée de sel</p>
  </ingredients>
  ▼<preparation>
    <![CDATA[ Mélanger les Carrés Frais avec la crème fraîche, ajouter le zeste du citron
    (lavé), puis incorporer les oeufs entiers, le sucre et la pincée de sel. Faire chauffer
    le four à 200°C (thermostat 6-7). Pendant ce temps, écraser les biscuits, et les
    mélanger avec le beurre. Tapisser le fond d'un plat carré, pas trop grand, de ce
    mélange, et cuire environ 30 min... Vérifier la cuisson avec votre couteau. Se déguste
    bien frais ! ]]>
  </preparation>
</recette>
```

*Exemple d'une recette du corpus*

Dans l'entête sont spécifiés les critères de la recette et la liste des ingrédients nécessaires. Les différents critères renseignés sont le titre de la recette, son type (*Entrée, Plat principal, Dessert*), son niveau (*Très facile, Facile, Moyennement difficile* ou *Difficile*), son coût (*Assez Cher, Bon marché, Moyen*) et la liste des ingrédients nécessaires à sa réalisation.

Nous allons vous présenter les caractéristiques du corpus utilisé et les différents problèmes qu'elles ont pu causer.

<sup>4</sup> <https://www.marmiton.org/>

## 2.2 Contenu

Marmiton étant un site participatif, toutes les recettes ne sont pas forcément uniformisées. On a notamment pu le constater au niveau de la liste des ingrédients. En effet, certains auteurs ajoutent des signes pour créer des jolies listes (ex : « <p>- 4 tranches de pain de mie </p> », « <p>x une pincée de sel </p> »), ou ajoutent beaucoup d'informations dans la liste d'ingrédient (ex : « <p>pâte feuilletée prête à dérouler ou faite maison pour les plus courageux</p> » , « <p>-tomates cerises pour la déco...et le goût</p> »), d'autres décident de mettre plusieurs ingrédients similaires dans le même paragraphe (ex : « <p> sel, poivre, herbe de provence, basilic </p> »), ou pire encore, mettent tous les ingrédients dans le même paragraphe (« <p>-une pâte à tarte (maison de préférence) -Une bûche de chèvre-6 "Chavroux"-1 cuillère à soupe de moutarde-2 à 3 tomates-Miel-Gruyère râpé-Sel, poivre, basilic</p> »).

De plus, chaque auteur a décidé lui-même de la difficulté de sa recette. On peut donc imaginer que la plupart des recettes seront classées comme *très faciles* ou *faciles*, de sorte qu'elles soient plus consultées. En effet, le tableau ci-dessous montre que les recettes *Très faciles* sont sur-représentées. On observe que c'est également le cas pour les recettes *Bon marché*.

Difficulté des recettes	Coût des recettes	Type des recettes
126 Difficile	700 Assez Cher	6938 Dessert
9581 Facile	6851 Moyen	5400 Entrée
1770 Moyennement difficile	15520 Bon marché	10733 Plat principal
11594 Très facile		

*Tableau présentant quelques statistiques sur le corpus*

Le corpus n'étant pas uniformisé, nous avons procédé à une normalisation des apostrophes afin de n'avoir que des apostrophes typographiques. Nous avons également eu besoin de faire beaucoup de nettoyage, notamment pour l'annotation de la quantité des ingrédients. En effet, certaines quantités étaient bizarrement construites : ½ cuillère à café de sucre, 1½ pot de yaourt , 2x4 œufs, 3à4 œufs, 3-4 œufs...

### Parcours du corpus

Nous avons voulu utiliser le style de programmation récursif au maximum, comme précisé dans la consigne, mais le corpus étant composé de 23 071 recettes nous avons atteint le maximum de récursions autorisées au moment d'appliquer nos traitements sur l'ensemble du corpus.

```

File "annotation.py", line 145, in annotation_ingr
    ingr_annotate.append(lire_ingr(ingredients, tokens_recette[0],
File "annotation.py", line 128, in lire_ingr
    return lire_ingr(ingredients[1:], token, ancien_token, futur_token)
File "annotation.py", line 128, in lire_ingr
    return lire_ingr(ingredients[1:], token, ancien_token, futur_token)
File "annotation.py", line 128, in lire_ingr
    return lire_ingr(ingredients[1:], token, ancien_token, futur_token)
[Previous line repeated 21 more times]
File "annotation.py", line 129, in lire_ingr
    return str(token)
File "spacy/tokens/token.pyx", line 119, in spacy.tokens.token.Token.__str__
File "spacy/tokens/token.pyx", line 113, in spacy.tokens.token.Token.__unicode__
File "spacy/tokens/token.pyx", line 263, in spacy.tokens.token.Token.text.__get__
File "spacy/tokens/token.pyx", line 806, in spacy.tokens.token.Token.orth.__get__
File "spacy/strings.pyx", line 126, in spacy.strings.StringStore.__getitem__
RecursionError: maximum recursion depth exceeded in comparison

```

*Erreur - Maximum de récursions atteint*

Pour contourner ce problème nous avons décidé d'augmenter le nombre de récursions autorisées avec :

```

import sys
sys.setrecursionlimit(10**6)

```

Nous avons donc séparé le corpus en trois parties, et nous lançons trois fois le script pour la partie annotation.

Pour les parties *complexité*(corpus annoté) et *résultats*(corpus final), nous avons considéré qu'une lecture itérative était plus appropriée pour la boucle de lecture des fichiers.

## 2.3 Conclusion

Les principales difficultés de notre corpus sont donc son manque d'uniformité ainsi que la répartition inégale des recettes dans les différentes catégories.



## RÉSULTATS

### 2.1 Introduction

Dans cette partie nous allons présenter quelques annotations réalisées par notre programme et évaluer celles-ci avec des calculs de précision, rappel et f-mesure.

Nous allons également proposer un coefficient de corrélation entre le niveau et le temps ou l'espace et réaliser une analyse de complexité de notre fonction d'annotation.

### 2.2 Contenu

```
▼<recette id="230091">
  <titre>Velouté potimarron, tomates, lentilles corail</titre>
  <type>Entrée</type>
  <niveau>Facile</niveau>
  <cout>Bon marché</cout>
  ▼<ingrédients>
    <p>1 potimarron</p>
    <p>3 tomates (environ 500 g) ou 1 boîte de 400 g de chair de tomate ou de tomates pelées</p>
    <p>100 g de lentilles corail</p>
    <p>1 gros oignon</p>
    <p>sel, poivre</p>
    <p>20 cl de crème fraîche épaisse</p>
    <p>persil et/ou ciboulette et/ou cerfeuil</p>
  </ingrédients>
  ▼<preparation>
    <![CDATA[ éplucher et couper l' <ingredient quantite=1>oignon</ingredient> en fines tranches. bien laver le <ingredient quantite=1>potimarron</ingredient>, enlever les pépins, et le couper en petits morceaux avec la peau. <operation temps=6.86min espace=9>faire chauffer l' huile et légèrement</operation> rissoler l' <ingredient quantite=1>oignon</ingredient> dans une <recipient>marmite</recipient> pendant 5 min. ajouter les morceaux de <ingredient quantite=1>potimarron</ingredient> et <operation temps=6.86min espace=9>couvrir largement d' eau</operation> ; porter à ébullition et <operation temps=20.0min espace=9>laisser cuire 20 min</operation>. rincer les <ingredient quantite=1>lentilles</ingredient> et les ajouter dans la <recipient>marmite</recipient> ainsi que les <ingredient quantite=1>tomates</ingredient> préalablement coupées en morceaux. <operation temps=10.0min espace=9>laisser cuire encore 10 min</operation>. passer au moulin à légumes et servir très chaud, avec de la <ingredient quantite=1>crème</ingredient> fraîche et herbes ciselées. ]]>
  </preparation>
</recette>
```

*Annotations complètes sur une recette de velouté potimarron, tomates et lentilles corail*

On peut observer que notre corpus final n'est pas parfaitement annoté. Un certain nombre d'opérations ne sont pas annotées (« éplucher et couper », « enlever les pépins » ...). Nous

supposons que certains verbes n'ont pas été correctement reconnus par SpaCy. En revanche, les ingrédients et les récipients semblent bien annotés, mis à part l'*huile* ou l'*eau* qui ne font pas partie des ingrédients de l'entête.

Afin d'évaluer notre annotation en ingrédients, nous avons récupéré les vrais positifs, faux positifs et faux négatifs et calculé les valeurs de précision, rappel et f-mesure :

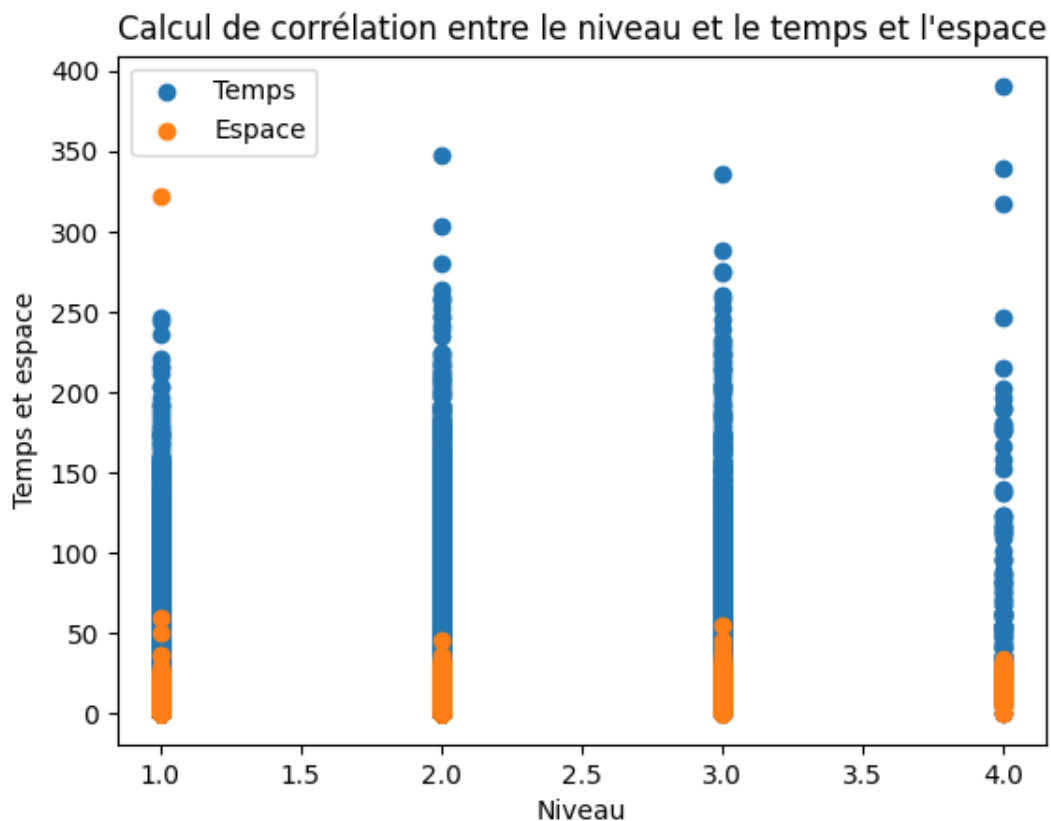
```
Vrais positifs : 144446
Faux positifs : 39378
Faux négatifs : 45567

Évaluation :
Précision : 0.786
Rappel : 0.76
F-mesure : 0.773
```

#### *Évaluation de notre fonction de reconnaissance des ingrédients*

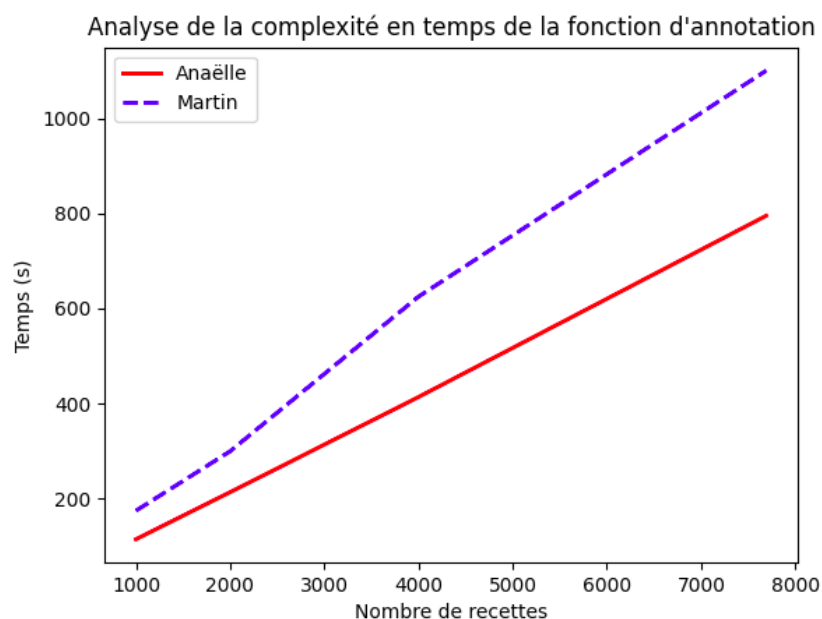
La f-mesure est donc de 0,77 ce qui est un score plutôt satisfaisant.

Nous avons ensuite calculé les différents coefficients de corrélation. Celui entre le niveau et le temps est de 0,25 et celui entre le niveau et l'espace est de 0,30. Le coefficient de corrélation est compris entre -1 et 1. Lorsque le coefficient est proche de 0, la relation linéaire entre les variables est faible.



Les plus grandes valeurs de temps se trouvent dans les recettes les plus difficiles et l'espace utilisé ne semble pas avoir beaucoup d'impact sur la difficulté de la recette. Enfin, nous avons réalisé une analyse de complexité de notre fonction d'annotation en relevant les différents temps de traitement selon le nombre de recettes à annoter pour nos ordinateurs respectifs :

Temps Martin	Temps Anaëlle
1000 recettes ==> 2min55	1000 recettes ==> 1min55
2000 recettes ==> 5min	2000 recettes ==> 3min34
4000 recettes ==> 10min25	4000 recettes ==> 6min54
7691 recettes ==> 18min20	7691 recettes ==> 13min15



Sans surprise, la complexité en temps de notre fonction d'annotation est linéaire.

## 2.3 Conclusion

On peut en conclure que notre programme reconnaît plutôt correctement les ingrédients et qu'il n'y a pas de corrélation flagrante entre le niveau de la recette et le temps ou l'espace.

## DISCUSSION

### 2.1 Introduction

Dans cette partie nous allons présenter quelques choix que l'on a fait lors du traitement de notre corpus et discuter de leur impact sur nos résultats.

### 2.2 Contenu

Afin de ne pas utiliser l'entête pour annoter les ingrédients de chaque recette, nous avons pensé à créer une liste d'ingrédients à partir des entêtes de toutes les recettes et créer ainsi un lexique des ingrédients. Cependant, nous nous sommes rendus compte qu'il existe de nombreuses manières d'écrire un même ingrédient et cette liste donnait beaucoup de bruit (« fines tranches de », « tranches fines mais pas trop épaisses », « cuiller|cuillère|cuillère|cuillerées etc... à café ou à soupe... »), sans compter les fautes d'orthographe. Malgré les nombreux traitements appliqués en bash pour tenter de normaliser le texte, il restait toujours beaucoup de bruit et cela aurait demandé trop de travail. De plus la liste était énorme : environ 42000 lignes après les tentatives de traitement. Nous avons donc décidé de changer de méthode. Une telle liste aurait peut-être pu aider à améliorer la reconnaissance en ingrédients de notre programme.

```
cat z_prepa_lexique_5 | sed 's/x [0-9] //g' | tr -s ' ' | sed -r 's/^un (petit )?peu (de
[d']moins d'1 de )//g' | sed -r 's/^(U|u)n ((petit )?(pot|verre(s)?
```

```
cat z_prepa_lexique_8 | sed -r s"/^(u|U)n(e)?
(rouelle|rouleau|rase|rasade|râpée|noix|noisette|moitié|marmite|demi-botte|(demi
cuillère|cuillerée) à (café|soupe)) (de [d'])(cuilléré|cuillère|cuiller) (a |à |de )//g" | sed -r
s"/^(u|U)n(e)?
```

Nous avons également écrit un script python pour prendre uniquement des unigrammes, bigrammes ou trigrammes afin de réduire la liste et essayer de faire en sorte que dans « pâte feuilletée prête à dérouler ou faite maison pour les plus courageux » ne soit pris en compte que « pâte feuilletée ». En ne prenant que les résultats ayant un nombre élevé d'occurrences nous espérions avoir une liste plus petite et relativement propre. Cependant cette idée n'a pas été concluante.

Une autre partie qui aurait pu être améliorée est l'annotation en opérations. Il reste de nombreux verbes qui n'ont pas été reconnus correctement et ces problèmes auraient pu être réglés avec des traitements plus fins.

Nous avons expliqué précédemment avoir privilégié une approche naïve pour la complexité, le but était de faire l'entièreté de la chaîne de traitement. Une fois cet objectif atteint, nous avons prévu d'affiner les calculs :

Pour la complexité en espace, toutes les balises d'une recette contiennent le nombre total de récipients de la recette. Une approche un peu moins naïve aurait été de marquer dans l'attribut *espace* de chaque balise *opération* le nombre de récipients utilisés par l'opération culinaire en question. Les démarches finales seraient de donner les noms des récipients utilisés dans l'attribut de la balise, pour ensuite spécifier pour chaque récipient, le nombre d'ingrédients qu'il contient durant l'opération culinaire afin de pouvoir déduire par les ingrédients d'une opération, le nombre de récipients nécessaires pour cette opération.

## 2.3 Conclusion

Pour conclure, nos annotations auraient pu être améliorées si nous avions eu à disposition une liste d'ingrédients à reconnaître dans les recettes et si nous avions utilisé des annotations en parties du discours plus fiables que celles de SpaCy.

Nous aurions pu améliorer nos estimations de complexité par recette en affinant les calculs pour le temps et l'espace.

# CONCLUSION GÉNÉRALE

Pour conclure, la phase la plus longue et la plus déterminante pour nos résultats a été celle de l'annotation. Si l'annotation des ingrédients et des récipients est plutôt satisfaisante, l'annotation des opérations laisse à désirer. Certaines recettes ne contiennent aucune opération annotée, et nous pensons qu'une annotation en parties du discours plus fiable nous permettrait d'améliorer grandement le nombre d'opérations reconnues.

Le calcul de la complexité en temps et en espace pour chacune des recettes a été réalisé de manière naïve et aurait pu être amélioré en annotant le nombre de récipients nécessaires à chaque opération plutôt que d'indiquer le nombre total de récipients de la recette. Nos estimations nous ont néanmoins permis d'observer que la relation linéaire entre la difficulté d'une recette et son temps ou son espace est faible.

Ce travail mériterait donc d'être continué avec de meilleures annotations et en approfondissant les calculs de complexité afin de permettre une interprétation plus fiable des coefficients de corrélations.

Durant la réalisation de ce projet, nous avons observé une progression dans l'écriture de notre code en python, notamment au niveau de la répartition des fonctions et des modules mais aussi de l'optimisation du code en récursif. Par exemple, notre première fonction récursive était très longue et peu lisible, puis nous avons appris à faire deux fonctions qui se répondent pour finalement imbriquer les fonctions récursives entre elles. Nous avons aussi, au fil des étapes du projet, mieux perçu quel style de programmation était pertinent pour une tâche donnée.

Comme vous l'avez peut-être remarqué, on s'est bien amusés.

Merci pour ce sujet.

# BIBLIOGRAPHIE

## *Analyse automatique de recettes de cuisine*

[Grouin et al, 2013] Grouin, C., Zweigenbaum, P., Paroubek, P. (2013). DEFT2013 se met à table : présentation du défi et résultats. *Actes du neuvième défi fouille de texte, DEFT2013*, Les Sables-d'Olonne, France. Pages 3-16. [[lien](#)]

[Hamon et al, 2013] Hamon T., Périnet A., Grabar N. (2013). Efficacité combinée du flou et de l'exact des recettes de cuisine. *Actes du neuvième défi fouille de texte, DEFT2013*, Les Sables-d'Olonne, France. Pages 19-32. [[lien](#)]

[Lejeune et al, 2013] Lejeune G., Lecluze C., Brixte R. (2013). DEFT2013, une cuisine de caractères. *Actes du neuvième défi fouille de texte, DEFT2013*, Les Sables-d'Olonne, France. Pages 33-40. [[lien](#)]

[Bost et al, 2013] Bost X., Brunetti I., Cabrera-Diego L.A., Cossu J-V., Linhares A., Morchid M., Torres-Moreno J-M., El-Bèze M., Dufour R. (2013). Systèmes du LIA à DEFT'13. *Actes du neuvième défi fouille de texte, DEFT2013*, Les Sables-d'Olonne, France. Pages 41-51. [[lien](#)]

[Dini et al, 2013] Dini, L., Bittar A., Ruhlmann M. (2013). Approches hybrides pour l'analyse de recettes de cuisine DEFT, TALN-RECITAL 2013. *Actes du neuvième défi fouille de texte, DEFT2013*, Les Sables-d'Olonne, France. Pages 53-65. [[lien](#)]

[Collin et al, 2013] Collin, O., Guerraz A., Hiou Y., Voisine N. (2013). Participation de Orange Labs à DEFT 2013. *Actes du neuvième défi fouille de texte, DEFT2013*, Les Sables-d'Olonne, France. Pages 67-79. [[lien](#)]

[Charton et al, 2013] Charton, E., Ludovic, J-L., Meurs, M-J., Gagnon M. (2013). Trois recettes d'apprentissage automatique pour un système d'extraction d'information et de classification de recettes de cuisine. *Actes du neuvième défi fouille de texte, DEFT2013*, Les Sables-d'Olonne, France. Pages 81-93. [[lien](#)]

## *Complexité*

[Paroubek, 2021] Paroubek P. (2021). Complexité en temps et en espace, théorique/empirique, style récursif. [[lien](#)]

## BONUS

**Extrait de conversation visant à démontrer l'impact positif de la programmation récursive sur le cerveau humain.**



Anaëlle Aujourd'hui à 03:29

```
cat z_prepa_lexique_8 | sed -r s"/^(u|U)n(e)?
(rouelle|rouleau|rase|rasade|râpée|quinzaine|noix|noisette|moitié|m
armite|demi-botte|(demi cuillère|cuillerée) à (café|soupe)) (de |d')|
(cuilléré|cuillère|cuiller) (a |à |de )//g" | sed -r s"/^(u|U)n(e)?
(brique|brousse|briquette|bûche|c a c|branche|botte(s)?
|boîte|barquette|assiette|assiette à soupe) (de |d')//g" | sed -r
s"/(u|U)n(e)? (petite |grosse |grande |belle |pincée |de |d')?//g" |
sort | uniq > z_prepa_lexique_9
```



Martin-Ramtin Aujourd'hui à 03:30

```
re(s)?|sachet|rouleau|reste|paquet|bol|bocal) (de |d'))?//g' | sed
's/^verres (de //g' | egrep verre
1783 cat z_prepa_lexique_5 | sed 's/x [0-9] //g' | tr -s ' ' | sed -r
's/^un (petit )?peu (de |d')moins d'1 de )//g' | sed -r 's/^(U|u)n
((petit )?(pot|verre(s)?|sachet|rouleau|reste|paquet|bol|bocal) (de
|d'))?//g' | sed -r 's/^verres (de |d')//g' | egrep verre
1784 cat z_prepa_lexique_5 | sed 's/x [0-9] //g' | tr -s ' ' | sed -r
's/^un (petit )?peu (de |d')moins d'1 de )//g' | sed -r 's/^(U|u)n
((petit )?(pot|verre(s)?|sachet|rouleau|reste|paquet|bol|bocal) (de
|d'))?//g' | sed -r 's/^verres (de |d')//g' | sed -r 's/^verres et demi
(de |d')//g' | egrep verre
1785 cat z_prepa_lexique_5 | sed 's/x [0-9] //g' | tr -s ' ' | sed -r
's/^un (petit )?peu (de |d')moins d'1 de )//g' | sed -r 's/^(U|u)n
((petit )?(pot|verre(s)?|sachet|rouleau|reste|paquet|bol|bocal) (de
|d'))?//g' | sed -r 's/^verres (de |d')//g' | sed -r 's/^verre(s)? et
demi (de |d')//g' | egrep verre
1786 cat z_prepa_lexique_5 | sed 's/x [0-9] //g' | tr -s ' ' | sed -r
's/^un (petit )?peu (de |d')moins d'1 de )//g' | sed -r 's/^(U|u)n
((petit )?(pot|verre(s)?|sachet|rouleau|reste|paquet|bol|bocal) (de
|d'))?//g' | sed -r 's/^verres (de |d')//g' | sed -r 's/^verre(s)? et
demi (de |d')//g' | sed 's/verre de //g' | egrep verre
```

