HAL
open science

DEFT2013, une cuisine de caractères

Gaël Lejeune, Charlotte Lecluze, Romain Brixtel

To cite this version:

Gaël Lejeune, Charlotte Lecluze, Romain Brixtel. DEFT2013, une cuisine de caractères. 20ème conférence du Traitement Automatique du Langage Naturel 2013 (TALN 2013), Jun 2013, Sables d'Olonne, France. hal-01074649

HAL Id: hal-01074649

https://hal.science/hal-01074649

Submitted on 15 Oct 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DEFT2013, une cuisine de caractères

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

RÉSUMÉ

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

ABSTRACT

DEFT2013, a distinctive character-based cuisine

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

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

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

1 Introduction

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

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

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

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

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

2 Description des corpus d'apprentissage et de test

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

2.1 Corpus d'apprentissage

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

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

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

0. http://www.marmition.org

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

FIGURE 1 – Un exemple de recette

2.2 Corpus de test

Un corpus de test différent était fourni pour chaque tâche.

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

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

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

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

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

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

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

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

3 Description de la méthode et résultats

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

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

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

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

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

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

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

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

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

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

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

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

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

3.1 Tâche 1 : identification du niveau de difficulté

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

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

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

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

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

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

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

3.2 Tâche 2 : identification du type de plat préparé

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

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

2. Sinon on conserve l'étiquette par défaut

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

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

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

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

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

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

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

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

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

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

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

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

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

4 Discussion

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

Remerciements

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

Références

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

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