HAL
open science

# Du bon usage d'ingrédients linguistiques spéciaux pour classer des recettes exceptionnelles

Elham Mohammadi, Louis Marceau, Eric Charton, Leila Kosseim, Luka Nerima, Marie-Jean Meurs

## To cite this version:

Elham Mohammadi, Louis Marceau, Eric Charton, Leila Kosseim, Luka Nerima, et al.. Du bon usage d'ingrédients linguistiques spéciaux pour classer des recettes exceptionnelles. 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 2 : Traitement Automatique des Langues Naturelles, 2020, Nancy, France. pp.81-94. hal-02784757v3

**HAL Id: hal-02784757**

**https://hal.science/hal-02784757v3**

Submitted on 23 Jun 2020

**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.

*Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition),*
*Traitement Automatique des Langues Naturelles (TALN, 27e édition),*
*Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)*
*Nancy, France, 08-19 juin 2020*

# Du bon usage d'ingrédients linguistiques spéciaux pour classer des recettes exceptionnelles

Elham Mohammadi [1, 2, 4] Louis Marceau [2] Eric Charton [2] Leila Kosseim [1]
Luka Nerima [3] Marie-Jean Meurs [4]
(1) Université Concordia, Montréal, Québec Canada
(2) Banque Nationale du Canada (BNC), Montréal, Québec Canada
(3) Université de Genève (UNIGE), Genève, Suisse
(4) Université du Québec à Montréal (UQAM), Montréal, Québec Canada
{elham.mohammadi,leila.kosseim}@concordia.ca, louis.marceau@bnc.ca, eric.charton@bnc.ca, luka.nerima@unige.ch, meurs.marie-jean@uqam.ca

## Résumé

Nous présentons un modèle d'apprentissage automatique qui combine modèles neuronaux et linguistiques pour traiter les tâches de classification dans lesquelles la distribution des étiquettes des instances est déséquilibrée. Les performances de ce modèle sont mesurées à l'aide d'expériences menées sur les tâches de classification de recettes de cuisine de la campagne DEFT 2013 (Grouin *et al.*, 2013). Nous montrons que les plongements lexicaux (*word embeddings*) associés à des méthodes d'apprentissage profond obtiennent de meilleures performances que tous les algorithmes déployés lors de la campagne DEFT. Nous montrons aussi que ces mêmes classifieurs avec plongements lexicaux peuvent gagner en performance lorsqu'un modèle linguistique est ajouté au modèle neuronal. Nous observons que l'ajout d'un modèle linguistique au modèle neuronal améliore les performances de classification sur les classes rares.

## Abstract

**Using Special Linguistic Ingredients to Classify Exceptional Recipes**

We propose a joint model composed of neural and linguistic sub-models, to address classification tasks in which the distribution of labels over samples is imbalanced. Different experiments are performed on tasks 1 and 2 of the DEFT 2013 shared task (Grouin *et al.*, 2013), which focused on classification of cooking recipes based on difficulty or meal type. In one set of experiments, the joint model is used for both classification tasks, whereas the second set of experiments involves using the neural sub-model, independently. This allows us to measure the impact of using linguistic features in the joint model. The results for both tasks show that adding a linguistic model to the neural model improves classification performance on the rare classes.

Mots-clés : Classification de textes, apprentissage profond, caractéristiques linguistiques.

Keywords: Text classification, deep learning, linguistic features, cooking recipes.

Volume 2 : Traitement Automatique des Langues Naturelles, pages 81–94.
HAL : hal-02784757.
Cette œuvre est mise à disposition sous licence Attribution 4.0 International.

# 1 Introduction

Différentes techniques issues du traitement automatique des langues (TAL) ont été utilisées au fil des ans pour traiter la tâche de classification de textes. Avec l'émergence de corpus de taille plus importante et l'avènement de l'apprentissage profond, les architectures de réseaux neuronaux sont devenues de plus en plus populaires dans l'exécution de différentes tâches de TAL (Amini *et al.*, 2019). Toutefois, malgré les progrès réalisés par l'apprentissage profond et l'obtention de résultats de pointe dans de nombreuses tâches, peu d'études ont abordé le défi que représentent les tâches de classification sur des ensembles de données déséquilibrés. Ces tâches de classification correspondent pourtant à de nombreux scénarios et applications de la vie réelle (Johnson & Khoshgoftaar, 2019), et représentent toujours un défi majeur.

Dans cet article, nous nous intéressons à deux tâches de classification multi-classes avec une distribution très déséquilibrée des étiquettes sur les données et nous mesurons l'efficacité de différentes méthodes pour relever un tel défi. Les tâches considérées sont la classification des recettes de cuisine selon 4 niveaux de difficulté (*Très facile*, *Facile*, *Assez difficile*, *Difficile* – Tâche 1) et la classification en fonction du type de repas (*Entrée*, *Plat principal*, *Dessert* – Tâche 2). Ces tâches et les ensembles de données, tous deux très déséquilibrés, sont issus de la campagne d'évaluation DEFT (DÉfi Fouille de Textes) 2013 (Grouin *et al.*, 2013). Nous expérimentons un modèle neuronal qui utilise des plongements lexicaux pré-entraînés comme caractéristiques d'entrée puis un modèle hybride composé de sous-modèles neuronaux et linguistiques et mesurons leur efficacité pour gérer une distribution de classe déséquilibrée. Par cette double expérimentation, nous cherchons à évaluer dans quelle mesure les plongements lexicaux pré-entraînés sont réellement en mesure de supprimer, dans une tâche de classification, tout recours au prétraitement linguistique. Nous souhaitons en particulier évaluer si l'influence du pré-traitement linguistique des données textuelles sur le caractère discriminant du classifieur demeure, en particulier lorsque ce classifieur est déjà renforcé par des plongements lexicaux.

Cet article est organisée comme suit : dans la section 2 nous passons brièvement en revue les travaux antérieurs connexes. La section 3 présente les ensembles de données utilisés et détaille les particularités de la campagne d'évaluation DEFT 2013. L'architecture globale du modèle, les sous-modèles et les différentes configurations utilisées sont décrits dans la section 5. La section 6 analyse les résultats obtenus et la section 8 conclut ce travail.

# 2 État de l'art

Selon Johnson & Khoshgoftaar (2019), en apprentissage automatique, trois types d'approches pour traiter les données déséquilibrées ont été proposées : (1) les approches au niveau des données qui consistent à modifier la distribution des classes par un sous- ou sur-échantillonnage des données limitant le déséquilibre ; (2) les approches algorithmiques qui adaptent l'apprentissage pour prendre en compte le déséquilibre en utilisant par exemple des poids sur les classes pour attribuer une pénalité plus importante à une erreur commise dans une classe rare par rapport à une classe fréquente ; (3) les approches hybrides qui combinent les approches (1) et (2) pour mieux gérer une distribution déséquilibrée.

| Tâche 1 - Niveau de difficulté | Entraînement | | Développement | | Test | |
|---|---|---|---|---|---|---|
| | Nb instances | Proportion | Nb instances | Proportion | Nb instances | Proportion |
| Très facile | 5569 | 50,2% | 1393 | 50,2% | 1132 | 49,0% |
| Facile | 4601 | 41,5% | 1151 | 41,5% | 968 | 41,9% |
| Assez difficile | 855 | 7,7% | 213 | 7,7% | 189 | 8,2% |
| Difficile | 64 | 0,6% | 16 | 0,6% | 20 | 0,9% |
| Total | 11089 | 100% | 2773 | 100% | 2309 | 100% |
| **Tâche 2 - Type de plat** | **Entraînement** | | **Développement** | | **Test** | |
| | **Nb instances** | **Proportion** | **Nb instances** | **Proportion** | **Nb instances** | **Proportion** |
| Entrée | 2599 | 23,4% | 647 | 23,3% | 562 | 24,4% |
| Plat | 5167 | 46,6% | 1280 | 46,1% | 1084 | 47,0% |
| Dessert | 3323 | 30,0% | 846 | 30,5% | 661 | 28,6% |
| Total | 11089 | 100% | 2773 | 100% | 2307 | 100% |

TABLE 1 – Composition des ensembles de données pour les tâches 1 et 2

Cependant, ces méthodes peuvent être d'une efficacité limitée dans le cas de classes extrêmement déséquilibrées. Krawczyk (2016) montre qu'il est alors intéressant d'extraire les caractéristiques discriminantes, en tenant compte du déséquilibre entre classes. Utiliser ces approches parallèlement aux représentations distribuées dans une architecture profonde améliore les résultats dans plusieurs types de tâches (Bogdanova *et al.*, 2017). De même, le modèle d'étiquetage morpho-syntaxique (POS) proposé par Bach *et al.* (2019) – un réseau bidirectionnel à mémoire (BiLSTM) suivi d'une couche de champs aléatoires conditionnels (CRF) prédisant les étiquettes – obtient des performances améliorées quand il est enrichi avec des caractéristiques conçues manuellement. Pour la classification des textes courts, les travaux de Wang *et al.* (2017) utilisent un modèle faisant appel à des représentations implicites (plongements lexicaux et plongements de caractères pré-entrainés) et explicites (concepts, extraits d'une base de connaissances). En alimentant un modèle convolutif à branches avec ces représentations, les auteurs obtiennent des résultats à l'état de l'art. Cela suggère que l'enrichissement des caractéristiques par des éléments issus d'une base de connaissances améliore les capacités de classification. Dans ce travail, nous ajoutons des caractéristiques linguistiques aux modèles neuronaux et mesurons le gain de performance obtenu, en mettant l'accent sur les classes minoritaires.

# 3 Ensembles de données

Nous avons choisi de retenir pour nos expériences le corpus de données proposé dans le cadre de la campagne d'évaluation Défi Fouille de Texte 2013 (DEFT 2013) [1]. DEFT est l'une des rares séries de campagnes d'évaluation annuelles à proposer à la communauté scientifique du TALN des tâches de classification en français, sur des corpus volumineux, avec des annotations originales. Elle offre également l'avantage de susciter l'intérêt de nombreux laboratoires et donc d'impliquer la contribution de plusieurs équipes qui appliquent, pour répondre à la tâche, des méthodes de classification très différenciées.

La tâche 2013 n'a pas échappé à cette tradition et a éveillé l'intérêt de 7 équipes qui représentent leurs laboratoires en France et au Canada. L'intérêt particulier de la campagne 2013 pour l'expérimentation que nous souhaitions mener est précisément sa date. Les méthodes proposées par ces 7 équipes établissent un état de l'art sur une tâche de classification de texte dé-balancée précisément quelques mois avant la généralisation, dans la littérature, de l'usage de l'apprentissage profond, puis des

1. https ://deft.limsi.fr/2013/

plongements lexicaux. Elle nous permet donc de comparer avec précision les performances de ces méthodes classiques avec les approches récentes, en bénéficiant d'un protocole expérimental rigoureux et, comme il est d'usage dans une campagne d'évaluation, fiable puisque géré par une tierce partie tant pour ce qui est de l'annotation des corpus que de la mesure de performance des systèmes proposés.

Le corpus utilisé pour le défi 2013 est composé de recettes de cuisine extraites du site Marmiton. Marmiton.org est l'un des sites culinaire francophone de large audience [2]. Les recettes rassemblées dans la base de données de Marmiton.org depuis 1999 sont proposées par les internautes via un formulaire validé pour publication par l'équipe de Marmiton. Lors de la soumission d'une recette, les internautes doivent indiquer le type de plat, le niveau de difficulté, le coût et le type de cuisson en sélectionnant des valeurs parmi des listes de choix pré-établies. Les paramètres numériques de la recette tels les temps de préparation et de cuisson, et le nombre de convives, sont à renseigner dans des champs contraints. Les ingrédients, les consignes de préparation et la boisson conseillée sont recueillis dans des champs en texte libre.

Le corpus d'entraînement contient 13.864 recettes pour un volume de données de 19,2 MB. Les corpus de test pour les tâches 1, 2 et 4 sont composés respectivement de 2309, 2307 et 2306 recettes pour des volumes de données de 3MB, 2,9MB et 2,2MB. Les recettes sont fournies au format XML. Chaque fichier du corpus d'entraînement contient le titre de la recette, son type, son niveau, son coût, la liste non normalisée de ses ingrédients et leur quantité d'usage, ainsi que les indications de préparation en texte libre.

Les données utilisées pour nos expériences sont celles des deux premières tâches de la campagne d'évaluation DEFT 2013 (Grouin *et al.*, 2013). La composition de ces ensembles de données est détaillée dans le tableau 1. Les données de la tâche 1 sont des recettes de cuisine en français, étiquetées avec leur niveau de difficulté respectifs sur une échelle de 1 à 4 (1 pour *Très facile*, 4 pour *Difficile*). La répartition des étiquettes dans cet ensemble de données est très déséquilibrée, avec plus de 90% des échantillons portant une étiquette *Très facile* ou *Facile* et un nombre nettement inférieur d'échantillons portant une étiquette *Assez difficile* ou *Difficile*. Les données de la tâche 2 sont des recettes de cuisine en français, étiquetées avec le type de plat de la recette, soit *Entrée*, *Plat principal* ou *Dessert*. Bien que la distribution des étiquettes ne soit pas aussi déséquilibrée que celle de la tâche 1, près de la moitié des échantillons appartiennent à la classe *Plat principal*, ce qui pose le problème d'un ensemble de données déséquilibré dans la tâche 2 également.

Lors de DEFT 2013, les données ont été publiées en deux parties, pour l'entraînement et le test. Pour les expériences rapportées dans ce document, 20% des données d'entraînement ont été utilisées pour la mise au point du modèle (développement) et le corpus de test original de la campagne a été utilisé pour comparer les résultats finaux. Les proportions des différentes parties du corpus sont présentées dans le tableau 1.

# 4 Méthodes de classification utilisées pour la comparaison

Dans la perspective de comparer efficacement les performances des algorithmes de classification avant et après l'apparition des méthodes à base d'apprentissage profond renforcé par des plongements lexicaux, il est important que les familles d'algorithmes utilisées pour les deux tâches principales de DEFT 2013 puissent être considérées comme représentatives de l'état de l'art de cette période.

2. Avec plus de 300.000 visiteurs par jour (chiffres Smart AdServer mars 2010, source : http ://www.marmiton.org/)

Dans nos analyses, nous utiliserons en tant que référence de comparaison les résultats obtenus par les systèmes proposés par les trois premières équipes, tant pour la tâche 1 que pour la tâche 2 de DEFT 2013. Ces trois équipes ont utilisées des méthodes de pré-traitement linguistique plus ou moins sophistiquées.

Pour la tâche 1, (Collin *et al.*, 2013) applique diverses méthodes de pré-traitement lexical et de sélection de variables. La classification est ensuite conduite avec l'algorithme d'entropie maximale. En ce qui concerne (Bost *et al.*, 2013), un algorithme de boosting est d'abord utilisé pour la tâche 1, associé à une préparation statistique des paramètres (polygrammes et données numériques issues d'observations sur le corpus). Des algorithmes tels que le SVM ainsi qu'une méthode issue de la de recherche d'information (similarité cosinus) sont également testés. Un algorithme de fusion de résultat de classifieur par combinaison linéaire est également utilisé. L'équipe (Charton *et al.*, 2013) qui obtient les meilleurs résultats sur la tâche 1, utilise un modèle d'arbre de décision dont les feuilles sont des fonctions de régression logistique (LMT). Pour la tâche 2, (Collin *et al.*, 2013) applique à nouveau diverses méthodes de pré-traitement lexical et de sélection de variables, accompagnées de variables numériques construites par calcul. La classification est ensuite conduite avec un algorithme propriétaire non décrit. Pour ce qui est de (Bost *et al.*, 2013) la tâche 2 est conduite avec des modèles classiques (boosting, SVM, méthode Electre), dont les résultats sont combinés par fusion. L'équipe (Charton *et al.*, 2013) obtient ses meilleurs résultats avec un classifieur SVM.

De manière générale, et au delà du classement final, on observe que ces trois équipes obtiennent avec leurs systèmes, appliqués sur le corpus de DEFT 2013, des résultats très proches avec les algorithmes et les techniques de pré-processing à l'état de l'art de l'époque : SVM, boosting, combinés avec une sélection fine des paramètres retenus pour l'apprentissage. On notera que des études subséquentes de (Charton *et al.*, 2014) sur cette campagne d'évaluation, entreprennent une comparaison systématique de classifieurs (Régression Logistique, Réseau Bayésien, SVM, arbres, Naîves Bayes) sur le corpus DEFT 2013, sans remettre en cause les résultats de la campagne.

Dans notre cadre expérimental, on peut donc considérer que les résultats de systèmes de classification de texte décrits dans la littérature et appliqués sur le corpus DEFT 2013 offrent une bonne représentation de l'état de l'art en matière de classification supervisée de documents textuels avant 2014. Et qu'il est en conséquence possible d'utiliser le corpus de recherche concerné comme base de comparaison avec des techniques plus récentes.

Par commodité, dans la suite de cette communication, nous désignerons par l'expression *modèle classique* les classifieurs non neuronaux utilisés sur les corpus DEFT 2013. Nous désignerons par l'expression *modèle linguistique* l'ensemble de pré-traitements appliqués aux corpus d'apprentissage textuels.

# 5 Conception des modèles

Notre but est de conduire deux séries de comparaisons. En premier lieu, nous voulons donc déterminer si un modèle neuronal associé à des plongements lexicaux est capable de performances supérieures à un *modèle classique* associé à un *modèle linguistique*. Nous désignerons dans nos résultats cette série d'expérience par le qualificatif de *modèle neuronal*. En second lieu, nous voulons déterminer si ce même modèle neuronal associé à des plongements lexicaux et complété du *modèle linguistique* améliore ses performances. Nous désignerons, dans nos résultats, cette série d'expérience par le

qualificatif de *modèle combiné*.

Dans cette section, nous présentons tout d'abord l'architecture des modèles neuronaux. En utilisant trois modèles de plongements (BERT, FLAUBERT, CAMEMBERT) que nous exploitons avec une architecture neuronale récurrente ou convolutive, nous obtenons 6 modèles (voir Table 2). Nous présentons ensuite le *modèle linguistique* et les détails de sa conception.

## 5.1 Modèle neuronal

**Plongement lexical.** Un plongement lexical est utilisé pour transformer la concaténation du titre d'une recette et du texte de préparation en vecteurs denses. Dans ce travail, trois plongements lexicaux pré-entraînés à base de transformeurs sont utilisés : la version multilingue de BERT (Devlin *et al.*, 2019) ainsi que CamemBERT (Martin *et al.*, 2019) et FlauBERT (Le *et al.*, 2020), construits sur des données françaises en utilisant un modèle BERT. Seules les caractéristiques de la dernière couche des modèles sont extraites, donnant une représentation dense de taille 768 pour chaque token.

**Architecture récurrente.** Pour la couche cachée de l'architecture récurrente, des réseaux de neurones récurrents à portes (GRUs) (Cho *et al.*, 2014) ont été utilisés, car ayant moins de paramètres, ils sont moins sujets au sur-apprentissage (Chung *et al.*, 2014) que les LSTM (Hochreiter & Schmidhuber, 1997). Un GRU bidirectionnel traite les plongements lexicaux consécutivement vers l'avant et vers l'arrière. La sortie de la couche GRU est ensuite transmise à une couche d'attention qui calcule sa moyenne pondérée à l'aide de l'équation $Attention = \sum_{t=1}^{n} y_t \omega_t$ où $y_t$ représente la sortie de la couche GRU au pas de temps $t$ et où $\omega_t$ est le poids attribué à $y_t$ par le mécanisme d'attention.

**Architecture convolutive.** Par soucis de comparaison, nous avons également testé une architecture convolutive. Un réseau de neurones convolutif (CNN) (LeCun *et al.*, 1999) traite des $n$-grammes d'entrée ($n$ représentations consécutives de tokens) en utilisant des filtres de convolution de taille $n$. La couche cachée est suivie d'une couche de regroupement moyen, maximum ou combinant les deux.

## 5.2 Modèle linguistique

Pour chacune des tâches de DEFT 2013 (tâche 1, identification d'une classe de difficulté de recette, et tâche 2, identification du type de plat préparé), un extracteur de caractéristiques est conçu. Les corpus d'apprentissage de la campagne DEFT 2013 sont composés de titres de recettes et de textes libres décrivant ces recettes. Le principe des extracteurs de caractéristiques est d'identifier des paramètres discriminants qui pourraient être dérivés de ces contenus textuels. Par une analyse quantitative systématique, on peut par exemple découvrir que le nombre de mots dans le titre d'une recette, ou encore le nombre d'ingrédients qu'elle contient, sont plus caractéristiques de sa difficulté que les mots (en tant qu'entités lexicales) qui composent ce titre. On espère ainsi, en utilisant ces paramètres dérivés du corpus textuel, plutôt que des sacs de mots, maximiser les performances du classifieur.

Aidé des observations faites sur les corpus, on bâtit des extracteurs dont le rôle est de construire des vecteurs avec les caractéristiques identifiées. Ce sont ces vecteurs qui seront utilisés pour entraîner et faire fonctionner le classifieur. Cette démarche de construction de vecteurs d'apprentissage constitués

de paramètres finement sélectionnés, voir fabriqués de toutes pièces (comme dans le cas des comptages de mots), est classique des systèmes de classification tels qu'on les observe dans la littérature avant la généralisation de l'utilisation des réseaux de neurones et l'avènement des plongements. C'est cette même méthode d'ingénierie des paramètres d'apprentissage que la littérature récentes sur les plongements considère comme moins utile voire superflue (Liu *et al.* (2015); Tymoshenko *et al.* (2016)). L'un des buts de nos expériences est de déterminer si le caractère discriminant des plongements est effectivement suffisant pour rendre l'ingénierie de paramètres inutiles.

Ces extracteurs de caractéristiques sont détaillés dans Charton *et al.* (2013), et leurs grands lignes sont présentées ci-après.

**Caractéristiques pour la tâche 1 (niveau de difficulté).** Sont utilisés le nombre de tokens dans le titre de la recette et dans la partie préparation, le nombre d'ingrédients, le coût du repas sur une échelle de 3 points, la présence de 22 mots et de 48 trigrammes discriminants, et enfin, le nombre de verbes dans la recette qui appartiennent à 3 familles de verbes discriminants. L'extraction de ces caractéristiques donne un vecteur de taille 77 pour chaque recette.

**Caractéristiques pour la tâche 2 (type de plats).** Comme pour la tâche 1, le nombre de tokens dans le titre de la recette et la partie préparation, le nombre d'ingrédients et le coût associé au repas sur une échelle de 3 points sont les quatre premières caractéristiques. S'y ajoutent 1231 noms d'ingrédients, 48 trigrammes discriminants et le nombre de verbes dans la recette appartenant à chacune des trois familles prédéfinies. Un vecteur de caractéristiques de taille 1286 est extrait pour chaque recette.

Pour être utilisé par le classifieur dans les expériences menées avec les *modèles combinés*, les vecteurs de caractéristiques ainsi extraits sont ensuite transmis à un réseau neuronal à simple couche à action anticipée, en faisant correspondre chaque vecteur de caractéristiques à un vecteur de même taille pour obtenir les représentations de sortie du modèle linguistique.

## 5.3 Composante de fusion

La composante de fusion concatène la sortie des deux modèles puis applique une couche entièrement connectée sur le vecteur résultant, lui faisant correspondre un vecteur de taille 4 dans le cas de la tâche 1, et de taille 3 dans le cas de la tâche 2. Cette couche est suivie d'une fonction d'activation softmax qui produit les probabilités des classes. Afin de mesurer l'impact du modèle linguistique, certaines expériences n'utilisent que le modèle neuronal. La fusion est alors remplacée par une couche entièrement connectée faisant correspondre la sortie de la couche d'attention avec le nombre de classes, suivie d'une fonction d'activation produisant la distribution des probabilités sur les classes.

## 5.4 Entraînement

Les modèles ont été entraînés avec des lots de taille 32 et 20 passes (*epochs*). Les paramètres ont été choisis lors de la passe qui a obtenu le meilleur micro score sur les données de développement. Le tableau 2 présente les hyperparamètres des modèles et des détails sont expliqués ci-après.

**Optimiseur.** AdamW (Loshchilov & Hutter, 2019) a été utilisé pour optimiser l'apprentissage. Pour tous les modèles, le taux d'apprentissage initial est de $10^{-3}$. Il a été adapté pour les modèles CNN à $10^{-4}$ après deux passes dans la tâche 1, et après cinq passes dans la tâche 2.

| | Modèle | Tâche 1 | | | Tâche 2 | | |
|---|---|---|---|---|---|---|---|
| | | #HL / #KH | #HN / #K | Regroupement | #HL / #KH | #HN / #K | Regroupement |
| Neuronal | CNN-BERT | 1, 2, 3, 4 | 300, 200, 100, 100 | max | 2 | 200 | max |
| | GRU-BERT | 1 | 64 | attention | 2 | 32 | attention |
| | CNN-FlauBERT | 2 | 200 | max, moyen | 1, 2 | 400, 200 | max |
| | GRU-FlauBERT | 1 | 64 | attention | 2 | 32 | attention |
| | CNN-CamemBERT | 2, 3 | 250, 50 | max | 2 | 200 | max, moyen |
| | GRU-CamemBERT | 2 | 32 | attention | 2 | 64 | attention |
| Combiné | CNN-BERT | 2 | 250 | max | 2 | 200 | max, moyen |
| | GRU-BERT | 1 | 64 | attention | 2 | 32 | attention |
| | CNN-FlauBERT | 1, 2 | 300, 200 | max, moyen | 1, 2 | 300, 200 | max, moyen |
| | GRU-FlauBERT | 1 | 64 | attention | 2 | 32 | attention |
| | CNN-CamemBERT | 1 | 400 | max, moyen | 2 | 400 | max, moyen |
| | GRU-CamemBERT | 2 | 32 | attention | 2 | 32 | attention |

TABLE 2 – Hyperparamètres pour chaque modèle. *#HL / #KH* : Nombre de couches cachées dans les modèles récurrents ou hauteur du noyau dans les CNN. *#HN / #K* : Nombre de noeuds cachés dans chaque couche récurrente ou nombre de noyaux dans les CNN.

**Poids des classes.** Des poids de classe ont été ajoutés dans la fonction de perte d'entropie croisée. Pour les expériences n'utilisant que le modèle neuronal, les poids ont été calculés automatiquement, en tenant compte de la proportion des échantillons de chaque classe. Dans les expériences utilisant le modèle combiné, les poids ont été manuellement réglés à 0,1, 0,1, 0,2 et 0,6 (correspondant respectivement aux classes *Très facile*, *Facile*, *Assez difficile* et *Difficile* pour la tâche 1, et à 0,6, 0,3 et 0,1 (correspondant aux classes *Entrée*, *Plat principal* et *Dessert*, respectivement) pour la tâche 2.

**Régularisation.** Afin de régulariser le réseau, l'optimiseur a été utilisé avec un taux de décroissance du poids de 0,02 . De plus, une couche de décroissance (*dropout*) avec une probabilité de 0,2 a été appliquée sur la concaténation de la sortie des deux modèles dans la composante de fusion et sur la sortie de la couche attention/regroupement dans les modèles neuronaux.

**Réglage fin des modèles BERT et CamemBERT.** Comme dans les expériences qui ont fait appel à des modèles combinés et neuronaux, la couche de plongement lexical a été figée, les modèles BERT et CamemBERT ont été affinés sur les deux tâches en tant qu'expériences supplémentaires.

# 6 Résultats

Les résultats des différentes expériences pour la tâche 1, ainsi que ceux des équipes DEFT 2013 ayant obtenu les meilleurs micro scores sont présentés dans le tableau 3. Dans tous les cas, le modèle combiné a permis d'obtenir des performances (souvent très) supérieures en termes de micro et macro F1 par rapport à un modèle uniquement neuronal. L'amélioration des résultats peut être également observée en termes de macro précision et de macro rappel.

Les scores micro et macro les plus élevés (sauf pour la macro précision (Charton *et al.*, 2014)) sont obtenus par des modèles combinés qui utilisent les plongements CamemBERT, illustrant leur efficacité sur des données en français. En outre, le tableau 3 montre que le modèle combiné CNN-CamemBERT dépasse largement tous les autres en termes de micro score, de macro F1 et de rappel dans la tâche 1. Dans 3 des 4 classes, le meilleur score F1 est également obtenu par des modèles combinés, en particulier ceux qui utilisent CamemBERT. Les résultats des modèles qui utilisent CamemBERT montrent que l'ajout de caractéristiques linguistiques améliore les performances par classe et que ces caractéristiques ont complété plus efficacement les plongements CamemBERT que BERT.

| | Modèle | Développement | | | | Test | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Micro Score | Macro F1 | Macro P | Macro R | Micro Score | Macro F1 | Macro P | Macro R |
| Neuronal | CNN-BERT | 61,5 | 39,3 | 41,1 | 37,6 | 58,8 | 37,7 | 39,4 | 36,1 |
| | GRU-BERT | 59,9 | 38,5 | 38,5 | 38,4 | 58,0 | 36,8 | 37,0 | 36,7 |
| | BERT affiné | 56,9 | 39,3 | 42,9 | 36,2 | 55,9 | 36,0 | 36,8 | 35,3 |
| | CNN-FlauBERT | 59,4 | 36,2 | 41,9 | 31,8 | 58,1 | 30,0 | 28,9 | 31,3 |
| | GRU-FlauBERT | 56,2 | 37,4 | 39,1 | 35,8 | 54,4 | 33,6 | 33,9 | 33,3 |
| | CNN-CamemBERT | 60,9 | 42,7 | 43,4 | 42,0 | 59,3 | 38,6 | 39,9 | 37,3 |
| | GRU-CamemBERT | 62,4 | 36,1 | 38,1 | 34,3 | 60,1 | 36,9 | 40,1 | 34,1 |
| | CamemBERT affiné | 61,2 | 37,3 | 38,5 | 36,2 | 59,3 | 37,6 | 38,9 | 36,4 |
| Combiné | CNN-BERT | 64,5 | 49,1 | 60,0 | 41,6 | 62,0 | 47,3 | 59,3 | 39,3 |
| | GRU-BERT | 65,8 | 41,7 | 45,5 | 38,5 | 63,1 | 39,3 | 42,1 | 36,8 |
| | CNN-FlauBERT | 63,7 | 45,1 | 56,7 | 37,4 | 60,6 | 43,3 | 55,1 | 35,7 |
| | GRU-FlauBERT | 62,6 | 49,0 | 49,8 | **48,3** | 61,2 | 44,3 | 45,2 | **43,4** |
| | CNN-CamemBERT | **66,4** | 50,3 | 58,5 | 44,2 | **63,8** | **50,0** | 62,0 | 42,0 |
| | GRU-CamemBERT | 65,3 | **51,1** | **68,5** | 40,8 | 63,1 | 40,5 | 42,5 | 38,7 |
| DEFT 2013 | Première équipe (Charton *et al.*, 2014) | - | - | - | - | 62,5 | 48,4 | **68,2** | 37,5 |
| | Seconde équipe (Collin *et al.*, 2013) | - | - | - | - | 61,2 | 45,1 | 52,4 | 39,5 |
| | Troisième équipe (Bost *et al.*, 2013) | - | - | - | - | 59,2 | 45,3 | 63,3 | 35,3 |

| | Modèle | Développement | | | | Test | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Très facile | Facile | Assez difficile | Difficile | Très facile | Facile | Assez difficile | Difficile |
| Neuronal | CNN-BERT | 66,3 | 61,6 | 25,1 | 0,0 | 63,1 | 59,7 | 23,5 | 0,0 |
| | GRU-BERT | 68,0 | 56,0 | **29,9** | 0,0 | 65,8 | 55,2 | 26,1 | 0,0 |
| | CNN-FlauBERT | 68,1 | 53,0 | 0,9 | 0,0 | 67,8 | 50,9 | 0,0 | 0,0 |
| | GRU-FlauBERT | 66,7 | 47,5 | 22,7 | 9,1 | 65,9 | 44,4 | 22,0 | 0,0 |
| | CNN-CamemBERT | 71,0 | 51,9 | 22,6 | 19,5 | 69,9 | 51,2 | 19,2 | 9,8 |
| | GRU-CamemBERT | 71,1 | 56,7 | 7,3 | 0,0 | 68,7 | 55,0 | 12,6 | 0,0 |
| Combiné | CNN-BERT | 72,1 | 60,8 | 24,9 | 21,1 | 70,0 | 58,1 | 22,5 | 17,4 |
| | GRU-BERT | 73,9 | 61,1 | 22,6 | 0,0 | 72,0 | 58,1 | 18,9 | 0,0 |
| | CNN-FlauBERT | 73,6 | 54,9 | 4,5 | 20,0 | 71,5 | 51,1 | 5,8 | 17,4 |
| | GRU-FlauBERT | 67,8 | **62,2** | 22,8 | **33,3** | 67,1 | **61,5** | 16,9 | 22,6 |
| | CNN-CamemBERT | 74,0 | 61,8 | 27,0 | 27,3 | 72,2 | 59,0 | 25,2 | **25,0** |
| | GRU-CamemBERT | **74,4** | 58,3 | 27,9 | 11,8 | **72,5** | 56,3 | **29,4** | 0,0 |
| DEFT 2013 | Première équipe (Charton *et al.*, 2014) | - | - | - | - | 71,7 | 56,2 | 18,8 | 9,5 |
| | Seconde équipe (Collin *et al.*, 2013) | - | - | - | - | 69,2 | 57,0 | 26,1 | 16,0 |
| | Troisième équipe (Bost *et al.*, 2013) | - | - | - | - | 68,6 | 52,5 | 15,6 | 9,5 |

TABLE 3 – Résultats généraux et par classe (score F1) pour la tâche 1.

Sur l'ensemble des tests, le modèle combiné CNN-CamemBERT obtient des scores F1 supérieurs au meilleur modèle de référence. Ce modèle combiné obtient également le meilleur score F1, soit 25%, dans la classe *Difficile*, qui est la plus rare. Seuls 3 des 8 modèles obtiennent un score F1 non nul dans la classe *Difficile* et deux sont des modèles combinés. Les résultats par classe montrent l'efficacité des caractéristiques linguistiques lorsque la tâche implique un ensemble de données très déséquilibré. Le tableau 4 montre les résultats obtenus pour la tâche 2. Le modèle CamemBERT affiné (miam !) a obtenu la meilleure performance globale. Toutefois, le modèle combiné CNN-CamemBERT, déjà le meilleur dans la tâche 1, fait mieux dans la phase de test. Cela montre que le modèle combiné CNN-CamemBERT semble mieux généraliser en présence de nouvelles instances.

D'après le tableau 4, tous les modèles combinés surpassent leurs homologues neuronaux en termes de micro et macro scores. Toutefois, contrairement à la tâche 1, cette amélioration n'est pas assez importante pour faire la différence en score micro moyen. Cette performance peut s'expliquer par les caractéristiques linguistiques utilisées pour la tâche 2, qui sont peut-être moins représentatives des classes que dans la tâche 1.

En examinant les résultats de la tâche 1 dans le tableau 3, on peut observer qu'après ajout du modèle linguistique, lorsqu'il y a une amélioration, celle-ci est significativement plus importante pour les classes rares. On peut donc émettre l'hypothèse que le modèle combiné gère mieux une distribution déséquilibrée des étiquettes. Sachant que cette distribution est nettement plus déséquilibrée dans la tâche 1 que dans la tâche 2, le modèle combiné est donc plus efficace dans le premier cas que dans le second. Enfin, le tableau 4 montre également les scores F1 par classe obtenus par les trois équipes les plus performantes de la campagne DEFT 2013.

| | Modèle | Développement | | | | Test | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Micro Score | Macro F1 | Macro P | Macro R | Micro Score | Macro F1 | Macro P | Macro R |
| Neuronal | CNN-BERT | 86,4 | 84,9 | 85,7 | 84,2 | 85,9 | 84,9 | 85,6 | 84,2 |
| | GRU-BERT | 84,4 | 83,2 | 83,2 | 83,2 | 84,8 | 84,0 | 84,0 | 83,9 |
| | BERT affiné | 86,3 | 85,8 | 85,1 | 86,5 | 86,4 | 86,2 | 85,6 | 86,8 |
| | CNN-FlauBERT | 86,4 | 85,2 | 85,3 | 85,1 | 86,7 | 85,8 | 86,2 | 85,5 |
| | GRU-FlauBERT | 84,2 | 83,3 | 82,6 | 84,0 | 85,1 | 84,6 | 83,8 | 85,4 |
| | CNN-CamemBERT | 87,6 | 86,8 | 86,5 | **87,2** | 88,1 | 87,6 | 87,6 | 87,7 |
| | GRU-CamemBERT | 86,5 | 85,6 | 85,5 | 85,6 | 87,1 | 86,5 | 86,6 | 86,4 |
| | CamemBERT affiné | **88,2** | **87,1** | **87,3** | 86,9 | 88,1 | 87,4 | 87,5 | 87,3 |
| Combiné | CNN-BERT | 86,0 | 85,2 | 84,9 | 85,4 | 87,0 | 86,5 | 86,3 | 86,7 |
| | GRU-BERT | 85,0 | 84,2 | 83,9 | 84,6 | 85,5 | 85,0 | 84,8 | 85,2 |
| | CNN-FlauBERT | 86,6 | 85,3 | 85,8 | 84,8 | 87,6 | 86,8 | 87,5 | 86,1 |
| | GRU-FlauBERT | 85,3 | 83,5 | 84,3 | 82,8 | 86,1 | 85,0 | 86,1 | 84,0 |
| | CNN-CamemBERT | 87,5 | 86,8 | 86,4 | 87,1 | 88,6 | **88,2** | 88,0 | **88,3** |
| | GRU-CamemBERT | 86,9 | 86,1 | 85,8 | 86,5 | 87,8 | 87,3 | 87,2 | 87,4 |
| DEFT 2013 | Première équipe (Bost *et al.*, 2013) | - | - | - | - | **88,9** | **88,2** | **88,4** | 88,1 |
| | Seconde équipe (Charton *et al.*, 2014) | - | - | - | - | 85,6 | 84,7 | 85,0 | 84,3 |
| | Troisième équipe (Hamon *et al.*, 2013) | - | - | - | - | 84,9 | 84,1 | 84,2 | 84,1 |

| | Modèle | Développement | | | Test | | |
|---|---|---|---|---|---|---|---|
| | | Entrée | Plat | Dessert | Entrée | Plat | Dessert |
| Neuronal | CNN-BERT | 70,2 | 86,5 | 97,6 | 71,0 | 86,4 | 96,8 |
| | GRU-BERT | 67,9 | 83,8 | 97,8 | 70,5 | 85,0 | 96,4 |
| | CNN-FlauBERT | 71,4 | 85,9 | 98,1 | 73,0 | 87,0 | 97,2 |
| | GRU-FlauBERT | 69,3 | 83,4 | 96,9 | 72,6 | 85,1 | 95,7 |
| | CNN-CamemBERT | **75,1** | **87,1** | 98,2 | 77,1 | 88,0 | 97,7 |
| | GRU-CamemBERT | 72,8 | 86,4 | 97,4 | 75,4 | 87,8 | 96,4 |
| Combiné | CNN-BERT | 72,0 | 85,6 | 97,7 | 75,0 | 87,2 | 97,3 |
| | GRU-BERT | 70,7 | 84,5 | 97,2 | 72,6 | 85,8 | 96,5 |
| | CNN-FlauBERT | 70,9 | 86,7 | 98,0 | 74,4 | 87,8 | 97,9 |
| | GRU-FlauBERT | 67,0 | 85,7 | 97,4 | 69,9 | 86,8 | 97,5 |
| | CNN-CamemBERT | 74,7 | 86,9 | **98,6** | **78,1** | 88,5 | 98,0 |
| | GRU-CamemBERT | 73,4 | 86,3 | 98,5 | 76,7 | 87,8 | 97,5 |
| DEFT 2013 | Première équipe (Bost *et al.*, 2013) | - | - | - | 77,3 | **88,8** | **98,6** |
| | Seconde équipe (Charton *et al.*, 2014) | - | - | - | 70,3 | 85,6 | 97,9 |
| | Troisième équipe (Hamon *et al.*, 2013) | - | - | - | 69,4 | 84,8 | 98,2 |

TABLE 4 – Résultats généraux et par classe (score F1) pour la tâche 2.

Parmi les 8 modèles que nous avons développés, les résultats montrent que le modèle conjoint CNN-CamemBERT obtient les scores F les plus élevés pour les trois classes de l'ensemble des tests. Il obtient également les meilleurs résultats sur la classe *Entrée* qui est la classe la plus rare. Ceci est en accord avec l'hypothèse selon laquelle la force du modèle combiné réside dans le traitement des classes rares.

# 7 Discussion

Il est intéressant d'observer que sur les corpus de tests, aucun des modèles neuronaux dont les plongements sont non personnalisés, n'obtient de meilleurs performances que les *modèles classiques* associés aux modèles linguistiques déployés lors de la campagne DEFT 2013. Il apparaît également que les modèles neuronaux dont les plongements sont affinés sur les corpus d'entraînement, produisent eux aussi des résultats inférieurs à ceux des *modèles classiques* associés aux modèles linguistiques.

On peut envisager ces résultats comme une indication que les modèles neuronaux à apprentissage profond, y compris lorsqu'ils sont finement entraînés ne produisent pas automatiquement de performances supérieures à celles des *modèles classiques* quand ces derniers sont finement paramétrés. Ceci semble particulièrement vrai sur des corpus textuels dé-balancés : les modèles neuronaux sont

incapables, dans toutes les configurations de nos expériences qui ne font pas appel aux modèles linguistiques, de modéliser la classe *difficile* de la tâche 1 alors que les *modèles classiques* y parviennent. Cette difficulté qu'ont les modèles neuronaux à modéliser les classes sous représentées apparaît dans d'autres expériences (Marceau *et al.*, 2019; Johnson & Khoshgoftaar, 2019). Il conviendrait de mener d'autres études dans d'autres contextes pour confirmer ce point. Si l'incapacité des modèles à apprentissage profond à modéliser une classe fortement dé-balancée venait à être confirmée plus largement, elle les disqualifierait pour de nombreuses applications, au profit d'autres classifieurs beaucoup plus performants dans ce contexte, tels que XGBoost (Chen & Guestrin, 2016; Nielsen, 2016).

On remarque néanmoins que les modèles neuronaux, avec très peu d'efforts de préparation, atteignent des scores très proches (moins de 1 point d'écart pour Camembert) de ceux obtenus par les *modèles classiques*, lorsque les corpus sont mieux balancés. Nous observons pour finir que tous les modèles neuronaux, lorsqu'ils voient leur données d'apprentissage enrichies par les vecteurs de paramètres issus du modèle linguistique, obtiennent les meilleures performances, y compris sur les classes dé-balancés. On peut déduire de cette observation qu'un réseau de neurone à apprentissage profond — même associé à des plongements censés lui fournir des caractères linguistiques discriminants — gagne à voir ses données d'entraînement complétées par des paramètres complémentaires et sélectionnés pour leurs propriétés discriminantes, obtenues via une phase d'ingénierie des paramètres.

# 8 Conclusion

Ces travaux dédiés à la classification de textes présentent un modèle combiné, composé d'un modèle neuronal et d'un modèle linguistique. Ce modèle combiné est évalué sur les tâches 1 et 2 de la campagne DEFT 2013 (Grouin *et al.*, 2013), qui consistent à classer les recettes de cuisine en français en fonction du niveau de difficulté ou du type de repas. Les résultats de ces expériences montrent que, dans les deux tâches, les modèles combinés sont plus performants que leurs homologues uniquement neuronaux. Dans la tâche 1, le modèle combiné a pu obtenir les meilleurs résultats en termes de micro et macro F1 moyens, ce qui montre son efficacité dans les contextes de classes très déséquilibrées.

Dans notre contexte expérimental, la conception rapide d'un système performant est possible avec les modèles neuronaux combinés avec des plongements, si les classes ne sont pas exagérément débalancées. Cependant, le système le plus performant et robuste est obtenu en soumettant aux modèles neuronaux des caractères discriminants patiemment sélectionnés.

# Reproductibilité

Pour faciliter la reproduction de nos travaux et permettre les comparaisons, nos systèmes sont disponibles en code source libre dans le dépôt suivant :
`https://github.com/cooking-classification/TALN2020`.
Les données peuvent être obtenues en contactant le comité d'organisation de la campagne DEFT 2013 (voir `https://deft.limsi.fr/2013/index.php`)

# Références


AMINI H., FARAHNAK F. & KOSSEIM L. (2019). Natural Language Processing : An Overview. In M. BLOM, N. NOBILE & C. Y. SUEN, Éds., *Frontiers in Pattern Recognition and Artificial Intelligence*, volume 5, chapitre 3, p. 35–55. World Scientific. DOI : 10.1142/9789811203527_0003.

BACH N. X., DUY T. K. & PHUONG T. M. (2019). A POS Tagging Model for Vietnamese Social Media Text Using BiLSTM-CRF with Rich Features. In *Pacific Rim International Conference on Artificial Intelligence*, p. 206–219 : Springer.

BOGDANOVA D., FOSTER J., DZENDZIK D. & LIU Q. (2017). If You Can't Beat them Join them : Handcrafted Features Complement Neural Nets for Non-factoid Answer Reranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 1, Long Papers*, p. 121–131.

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. In *Actes du neuvième DÉfi Fouille de Textes (DEFT2013), atelier de la 20e conférence de la conférence sur le Traitement Automatique du Langage Naturel 2013 (TALN 2013) et de la 15ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL 2013)*, Les Sables-d'Olonne, France. HAL : hal-01313065.

CHARTON E., JEAN-LOUIS L., MEURS M.-J. & GAGNON M. (2013). Trois recettes d'apprentissage automatique pour un systeme d'extraction d'information et de classification de recettes de cuisines. In *Actes du neuvième DÉfi Fouille de Textes (DEFT2013), atelier de la 20e conférence de la conférence sur le Traitement Automatique du Langage Naturel 2013 (TALN 2013) et de la 15ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL 2013)*, p. 17–21, Les Sables-d'Olonne, France. https://deft.limsi.fr/actes/2013/pdf/deft13_submission_6.pdf.

CHARTON E., MEURS M.-J., JEAN-LOUIS L. & GAGNON M. (2014). Using Collaborative Tagging for Text Classification : From Text Classification to Opinion Mining. *Informatics*, **1**(1), 32–51. DOI : 10.3390/informatics1010032.

CHEN T. & GUESTRIN C. (2016). XGBoost : A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, p. 785–794.

CHO K., VAN MERRIENBOER B., GULCEHRE C., BAHDANAU D., BOUGARES F., SCHWENK H. & BENGIO Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, p. 1724–1734, Doha, Qatar.

CHUNG J., GULCEHRE C., CHO K. & BENGIO Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *NIPS 2014 Deep Learning and Representation Learning Workshop*, Montreal, Canada.

COLLIN O., GUERRAZ A., HIOU Y. & VOISINE N. (2013). Participation de Orange Labs à DEFT 2013. In *Actes du neuvième DÉfi Fouille de Textes (DEFT2013), atelier de la 20e conférence de la conférence sur le Traitement Automatique du Langage Naturel 2013 (TALN 2013) et de la 15ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL 2013)*, p. 67–79, Les Sables-d'Olonne, France. https://deft.limsi.fr/actes/2013/pdf/deft13_submission_5.pdf.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of*

*the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (ACL/HLT 2019)*, p. 4171–4186, Minneapolis, Minnesota.

GROUIN C., PAROUBEK P. & ZWEIGENBAUM P. (2013). DEFT2013 se met à table : présentation du défi et résultats. In *Actes du neuvième DÉfi Fouille de Textes (DEFT2013), atelier de la 20e conférence de la conférence sur le Traitement Automatique du Langage Naturel 2013 (TALN 2013) et de la 15ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL 2013)*, Les Sables-d'Olonne, France. https://deft.limsi.fr/actes/2013/pdf/deft13_submission_0.pdf.

HAMON T., PÉRINET A. & GRABAR N. (2013). Efficacité combinée du flou et de l'exact des recettes de cuisine. In *Actes du neuvième DÉfi Fouille de Textes (DEFT2013), atelier de la 20e conférence de la conférence sur le Traitement Automatique du Langage Naturel 2013 (TALN 2013) et de la 15ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL 2013)*, Les Sables-d'Olonne, France. https://deft.limsi.fr/actes/2013/pdf/deft13_submission_1.pdf.

HOCHREITER S. & SCHMIDHUBER J. (1997). Long Short-Term Memory. *Neural Computation*, **9**(8), 1735–1780.

JOHNSON J. M. & KHOSHGOFTAAR T. M. (2019). Survey on Deep Learning with Class Imbalance. *Journal of Big Data*, **6**(1), 27.

KRAWCZYK B. (2016). Learning from Imbalanced Data : Open Challenges and Future Directions. *Progress in Artificial Intelligence*, **5**(4), 221–232.

LE H., VIAL L., FREJ J., SEGONNE V., COAVOUX M., LECOUTEUX B., ALLAUZEN A., CRABBÉ B., BESACIER L. & SCHWAB D. (2020). FlauBERT : Unsupervised Language Model Pre-training for French. In *Proceedings of the 12th Edition of the Language Resources and Evaluation Conference (LREC 2020)*. arXiv : 1912.05372.

LECUN Y., HAFFNER P., BOTTOU L. & BENGIO Y. (1999). Object Recognition with Gradient-based Learning. In D. A. FORSYTH, J. L. MUNDY, V. DI GESÚ & R. CIPOLLA, Éds., *Shape, contour and grouping in computer vision*, volume 1681 de *Lecture Notes in Computer Science*, p. 319–345. Springer. DOI : 10.1007/3-540-46805-6_19.

LIU P., JOTY S. & MENG H. (2015). Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 1433–1443.

LOSHCHILOV I. & HUTTER F. (2019). Decoupled Weight Decay Regularization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2019)*, New Orleans, Louisiana, USA.

MARCEAU L., QIU L., VANDEWIELE N. & CHARTON E. (2019). A comparison of deep learning performances with other machine learning algorithms on credit scoring unbalanced data. arXiv preprint : 1907.12363.

MARTIN L., MULLER B., SUÁREZ P. J. O., DUPONT Y., ROMARY L., DE LA CLERGERIE É. V., SEDDAH D. & SAGOT B. (2019). CamemBERT : A Tasty French Language Model. arXiv preprint : 1911.03894.

NIELSEN D. (2016). Tree boosting with XGBoost- Why does XGBoost win "every" machine learning competition ? Mémoire de master, NTNU.

TYMOSHENKO K., BONADIMAN D. & MOSCHITTI A. (2016). Convolutional neural networks vs. convolution kernels : Feature engineering for answer sentence reranking. In *Proceedings of the*

*2016 conference of the North American chapter of the association for computational linguistics : human language technologies*, p. 1268–1278.

WANG J., WANG Z., ZHANG D. & YAN J. (2017). Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification. In *International Joint Conference on Artificial Intelligence (IJCAI)*, p. 2915–2921.