

UNIVERSITÉ PARIS 10 NANTERRE
MASTER PLURITAL



MÉMOIRE

Analyse d'opinion dans les dialogues d'agents
conversationnels

Opinion analysis in chatbots talks

par

Guillaume Le Noé-Bienvenu

Session : Juin 2019

Directeur de mémoire

Damien Nouvel

Encadrant Orange Bank

Djamel Mostefa

Attestation de non-plagiat

Déclaration sur l'honneur

Je soussigné Guillaume Le Noé-Bienvenu, déclare avoir rédigé ce travail sans aides extérieures ni sources autres que celles qui sont citées. Toutes les utilisations de textes préexistants, publiés ou non, y compris en version électronique, sont signalées comme telles. Ce travail n'a été soumis à aucun autre jury d'examen sous une forme identique ou similaire, que ce soit en France ou à l'étranger, à l'université ou dans une autre institution, par moi-même ou par autrui.

Date : 15 juin 2019

Signature :

A handwritten signature in black ink, appearing to read 'Guillaume Le Noé-Bienvenu', written in a cursive style.

Remerciements

Je tiens à remercier les enseignants du master pluriTAL pour leur disponibilité et leur accompagnement pendant ces deux années d'études, qui ont été riches en découvertes.

Je remercie également mon directeur de mémoire, Damien Nouvel, qui m'a accompagné pendant l'élaboration de ce mémoire.

Je souhaite remercier mon encadrant de stage Djamel Mostefa et toute l'équipe du pôle Intelligence Artificielle d'Orange Bank pour mon intégration dans l'équipe.

Enfin, je souhaite particulièrement remercier mes parents pour leur soutien lors de cette reprise d'études.

Résumé

L'analyse d'opinion est utilisée par les entreprises afin de mieux connaître le ressenti des clients sur un produit. Le traitement automatique des langues aide à répondre à cette problématique, en faisant ressortir des tendances au sein d'un corpus. Dans ce mémoire, nous proposons plusieurs applications de l'analyse d'opinion sur les conversations d'un agent conversationnel. Ces applications, qui peuvent produire des visualisations, permettent de déterminer la satisfaction du client pour des thématiques déterminées et de regarder de plus près celles sur lesquelles les clients ont une mauvaise opinion. Ce mémoire présente la mise en place d'un tel outil au sein de l'entreprise Orange Bank, en se focalisant sur une analyse temporelle de ces indicateurs et la comparaison de l'interaction entre un agent conversationnel (chatbot) et un agent humain.

Abstract

Opinion mining is used by companies to better understand how customers feel about a product. The natural language processing tools help to address this issue by highlighting trends through data. In this thesis, we propose several uses of opinion analysis on conversational agent conversations. These applications, which can produce visualizations, make it possible to determine customer satisfaction for specific topics and to take a closer look at those on which customers have a poor opinion. This thesis presents the implementation of such tool within Orange Bank, focusing on a temporal analysis of these indicators and the comparison of the interaction between a conversational agent (chatbot) and a human agent.

Chapitre 1

Introduction

1.1 Motivations

De nos jours, de plus en plus d'entreprises ont recourt à des assistants virtuels. Ces assistants, que nous appellerons *chatbots*, sont utilisés dans de nombreux secteurs (vente à particulier, support logiciel, réservation de services, etc.) et peuvent être dirigés pour les clients d'une entreprise (B2C pour Business To Client) ou ses collaborateurs (B2E pour Business To Employees). L'analyse d'opinion sur les conversations chatbot permet de répondre à plusieurs besoins, par exemple savoir si les utilisateurs du chatbot sont satisfaits de l'interaction, ou avoir un aperçu "à chaud" de la satisfaction des clients.

1.2 Présentation de l'entreprise Orange Bank

Orange Bank est une banque 100% mobile dont le lancement a eu lieu en novembre 2017. Ses services sont uniquement délivrés via des applications web ou mobiles et ne possèdent pas, contrairement aux banques traditionnelles, d'agences physiques. C'est la filiale bancaire de l'opérateur téléphonique Orange. Afin de pouvoir répondre aux nombreuses demandes de ses clients, Orange Bank a mis en place sur les applications web et mobiles un chatbot nommé Djingo. Ce conseiller virtuel se base sur la technologie Watson d'IBM.

1.3 Présentation de l'agent conversationnel Djingo

Djingo est l'agent conversationnel d'Orange Bank, disponible 24h/24h 7 jours sur 7 pour ses 3 000 utilisateurs quotidiens. Il est capable de comprendre 390 intentions et dispose de 1 000 réponses adaptées aux besoins de l'utilisateur. Djingo est à la fois utilisé comme une Foire aux Questions (FAQ) portant sur les produits commercialisés par Orange Bank (exemple : ai-je des frais de retraits à l'étranger, combien de temps prend la livraison d'un chéquier, etc.) et comme assistant pour réaliser des actions liées au compte client (par exemple : commander un chéquier, bloquer la carte bancaire, etc.). Les réponses orientées FAQ sont généralement les mêmes pour tous les clients, mais les réponses réalisant une action déclenchent une action qui dépend du compte client. Par exemple, si un utilisateur souhaite commander un chéquier, Djingo va vérifier si l'utilisateur est identifié, s'il n'y a pas de commande de chéquier en cours, si l'utilisateur peut commander un nouveau moyen de paiement, etc. À chaque étape, en fonction des éléments reçus par l'intermédiaire d'une interface programmatique (API ou Application Programming Interface), Djingo apporte à l'utilisateur une réponse appropriée. Lors de la conversation, des thèmes sont identifiés

en lien avec les intentions. Les thèmes, comme les intentions, sont détectés par la brique Watson. À ce jour, il existe environ 60 thèmes : Orange-Bank, app-site-info, app-site-probleme, assurance-info, assurance-resiliation, etc. Les conversations peuvent contenir plusieurs thèmes. Si l'utilisateur pose une question dont Djingo n'a pas la réponse ou qu'il détecte que l'utilisateur ne parvient pas à se faire comprendre, il propose à l'utilisateur d'être redirigé vers un conseiller. Lors des interactions, l'assistant conversationnel Djingo peut suggérer des boîtes de dialogues dans lesquelles les utilisateurs n'ont pas d'autre choix que de sélectionner une proposition.

C'est le cas par exemple du premier message de Djingo sur le site d'Orange Bank 1.1.

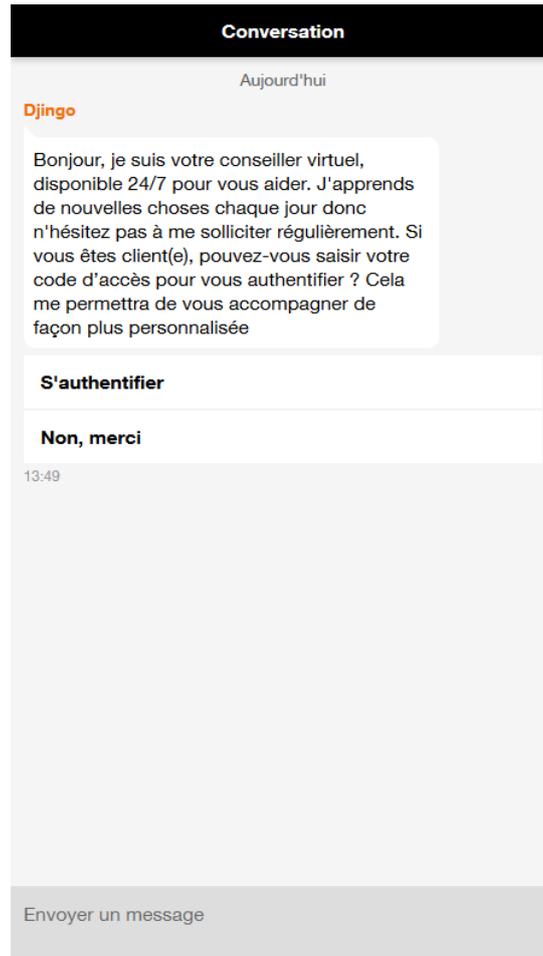


FIGURE 1.1 – Lorsque la fenêtre de Chat s'ouvre, Djingo demande si l'utilisateur souhaite s'identifier. L'utilisateur n'a d'autre choix que de sélectionner une des réponses proposées.

Chapitre 2

État de l'art

Comme nous l'avons indiqué, il est important pour Orange Bank de savoir comment les utilisateurs interagissent avec Djingo, s'ils en sont satisfaits et quels sont les points à améliorer. Voilà pourquoi nous avons établi un état de l'art sur les opinions, la polarité et la classification de textes.

2.1 Opinion, polarité, émotions

L'analyse d'opinion (ou opinion mining, sentiment analysis) est une tâche de TAL visant à quantifier ou qualifier les opinions telles qu'elles sont exprimées dans un texte. Une opinion, si elle peut être mesurée, se rapporte dans la grande majorité des cas à une entité. Ce domaine a été fortement développé par l'essor des réseaux sociaux et des blogs, sur lesquels les internautes émettent des avis sur des produits. Beaucoup d'entreprises cherchent donc à connaître l'opinion des utilisateurs sur les produits commercialisés ou leurs services.

Une opinion est un avis porté sur une entité (une personne, un lieu, un évènement, etc.). Liu [1] définit l'opinion comme un quintuplet $(o_j, f_{jk}, oo_{ijkl}, h_i, t_l)$ où

- o_j est l'objet sur lequel l'opinion est portée
- f_{jk} correspond à une propriété de l'objet o_j
- oo_{ijkl} est associé au sentiment ou polarité de l'énonciateur de l'opinion h_i , sur la propriété f_{jk} de l'objet o_j . Cet élément peut être positif, négatif, ou neutre, ou disposer de plus de granularité
- h_i représente la personne énonciatrice de l'opinion
- t_l se rapporte au temps à laquelle l'opinion est exprimée

Il définit ensuite le but de l'analyse d'opinion comme déterminer tous les quintuples d'opinions dans un document. Dans notre cas, l'objet de l'opinion seront les thématiques identifiées dans les messages. Notons que l'analyse des sentiments se porte sur des objets ou leurs propriétés, mais peu sur des conversations humain / chatbot.

Les travaux de Zhang et Ferrari [2] définissent la polarité comme une propriété d'un texte pouvant être positif, négatif ou neutre. Elle fait référence à des valeurs esthétiques, cognitives, éthiques ou pratiques. Par exemple, les termes « beau », « intelligent » et « dynamique » disposent d'une connotation positive, tandis que « laid », « stupide » et « apathique » sont négatifs. Dans les exemples de la table 2.1, l'exemple négatif contient un terme disposant d'une forte connotation négative (« gâchis »), alors que l'exemple positif ne contient pas de termes discriminants. La polarité est souvent utilisée en TAL

afin d’avoir un ressenti global sur un texte. Les guides d’annotation des DEFT (Défis Fouilles de Textes) 2015 [3] et 2018 [4] disposent de respectivement trois et quatre classes de polarité : positif, négatif, neutre ou mixte pour le premier, positif, négatif, neutre, mixte positif-négatif pour le second.

Tweet	Polarité
Appareil connecté : coût planétaire, gâchis environnemental	négatif
Les services rendus par les écosystèmes, si on les chiffrerait, reviendrait... au PIB mondial!	neutre
Des scientifiques élaborent une cellule #solaire qui stocke sa propre énergie	positif

TABLE 2.1 – Exemple d’annotation de polarité du DEFT 2015

Enfin, les émotions sont définies comme des réactions affectives transitoires, habituellement provoquées par une stimulation venue de l’environnement (dictionnaire Larousse). Parrott [5], définit six émotions basiques : bonheur, tristesse, peur, colère, dégoût et surprise, qui peuvent-elles mêmes être subdivisées en plusieurs catégories. Liu nous indique aussi que l’intensité des opinions ou sentiments peut être liée à certaines émotions. Par exemple une opinion très négative pourra être liée à la colère.

Des outils permettant de travailler sur la polarité et les émotions fonctionnant à base de règles comme VADER [6] ou SentiWordNet [7] sont disponibles en ligne, mais restent appliqués à la langue anglaise. Pour le français, il existe aussi des ressources, comme le Corpus Cénaphore [8], mais restent spécifiques au tweets.

Les opinions peuvent être détectées sur des empan de texte, sur des messages, ou encore sur des documents. Dans le corpus, il est rare que plusieurs opinions soient exprimées dans un message, ce qui nous a poussé à nous diriger vers la classification de texte afin de pouvoir analyser la polarité des messages.

2.2 Classification de textes

La classification de texte est une tâche de fouille de texte qui vise à assigner une ou plusieurs classes à un document. Elle s’appuie généralement sur le contenu textuel du document en utilisant des modèles à base de sacs de mots. Dans notre cas, un message constitue un document et les classes peuvent être : positif, négatif ou neutre. Un message ne peut appartenir qu’à une seule catégorie. L’apprentissage automatique consiste à faire en sorte que la machine puisse étiqueter ou catégoriser des documents. En tenant compte de suffisamment de documents appartenant à chacune des classes, le système est en mesure d’établir des prédictions de documents qu’il n’a jamais traités. Dans notre cas, cet apprentissage est supervisé, c’est à dire que les classes du corpus d’entraînement sont attribuées par manuellement (corpus annoté). L’algorithme d’apprentissage se sert de la catégorie de chaque document pour apprendre à partir d’exemples. Une fois la partie d’apprentissage terminée, le système est testé sur un autre jeu de données. On peut alors quantifier la qualité de l’annotation produite par l’algorithme à l’aide de métriques d’évaluation (f-mesure, rappel, précision...).

L’algorithme d’apprentissage automatique des SVMs a été utilisé dès les années 1990 pour catégoriser des textes [9]. Ceux-ci obtiennent généralement des scores corrects,

comme par exemple plus de 80% de f-mesure par Pang and Lee [10]. Le fonctionnement de l’algorithme est décrit dans l’annexe 9.

Il est aussi possible d’utiliser des algorithmes ayant recours à l’apprentissage profond (deep learning). Les réseaux de neurones convolutionnels obtiennent de très bons scores pour la classification de texte [11], mais demandent plus de temps et d’exemples pour l’entraînement. Aussi, les vainqueurs de nombreux défis en TAL (DEFT 2015, Hackatal 2018) ont principalement utilisé les SVM. Nous avons donc choisi d’implémenter cet algorithme de classification pour nos expériences.

2.3 Travaux précédents

Les expériences réalisées dans le cadre de ce mémoire font suite au stage d’Audrey CORNU, qui a eu lieu en 2018. Les travaux étaient centrés sur l’analyse d’opinion, et en particulier, l’analyse de la polarité sur des données issues de commentaires venant de réseaux sociaux, de formulaires de satisfaction, ou de commentaires de l’application OrangeBank. L’approche utilisée était de classer ces données suivant trois classes (positif, négatif, neutre). Pour ce faire, 3000 documents avaient été annotés manuellement, et des algorithmes d’apprentissage supervisés avaient été entraînés sur ces données. Les résultats de ces algorithmes de classification sont reportés dans la table 2.2.

Algorithme utilisé	f-mesure micro	f-mesure macro
SVM	0.82	0.82
Naïve Bayes	0.73	0.70
Régression logistique	0.82	0.82
Gradient stochastique	0.81	0.81

TABLE 2.2 – Tableau récapitulatif des résultats précédents

Chapitre 3

Méthodologie

Les messages des conversations étant courts et généralement peu chargés en sentiments, nous avons favorisé une analyse de la polarité plutôt qu'une analyse des sentiments. Ce mémoire explore donc l'analyse d'opinion dans les interactions humains/chatbots, et en particulier la polarité. Cette évolution a été étudiée à plusieurs échelles. Nous avons tout d'abord travaillé au niveau de la polarité des messages et conversations sur une échelle de temps, croisée ou non avec les thèmes des conversations (par exemple comment évolue la polarité des conversations sur le mois). Puis dans un deuxième temps, nous nous sommes attardés sur l'évolution de la polarité au sein de la conversation (comment évolue la polarité des messages au cours de l'interaction avec l'agent artificiel, puis si tel est le cas avec l'agent humain.).

Comme énoncé dans la partie précédente, nous nous sommes d'abord orientés vers la classification de textes, en construisant un modèle permettant de catégoriser les messages en fonction de leur polarité (positif, négatif ou neutre). Pour construire ce modèle, des données ont été labelisées manuellement, avant d'entraîner un classifieur sur celles-ci. Ensuite, ce modèle a été appliqué sur l'intégralité du corpus, afin de pouvoir visualiser l'évolution de la polarité. Enfin, ce modèle a été exploité pour analyser l'évolution de la polarité à l'intérieur de la conversation.

Chapitre 4

Données

4.1 Description qualitative

Les données utilisées dans ce mémoire sont constituées de logs de conversation écrites des utilisateurs de Djingo. Sachant que Djingo converse quotidiennement jusqu'à maintenant, nous nous avons extrait la période Novembre 2017 - Décembre 2018.

Les données sur l'année 2019 ont été utilisées dans certains cas afin de visualiser les tendances sur une plus grande période de temps.

4.2 Description quantitative

Les données issues des logs contiennent une vingtaine de méta-données. Toutes n'ont pas été utiles lors des expérimentations. Les éléments les plus pertinents figurent dans la table 4.1. Les méta-données nous ont permis d'ordonner les messages pour analyser les conversations. La table 4.1 résume les quantités de données. Le corpus est volumineux (6 millions de messages) et adapté aux approches par apprentissages. Enfin, la table 4.3 présente les messages les plus fréquents dans le corpus, qui sont généralement des messages courts (« Non », « bonjour », etc.) ou préformulés (« Je veux être mis en relation avec un expert ? », « Offre de bienvenue à l'ouverture d'un compte », etc.).

Conversation_id	L'identifiant de la conversation
CS_Req_Timestamp	L'heure à laquelle le message a été reçu
CS_Resp_Timestamp	L'heure à laquelle la réponse au message a été envoyée
Question	La question de l'utilisateur de Djingo
SL_Answer	La réponse de Djingo
CS_Intent1	La première intention détectée par Djingo
CS_I1_conf	La probabilité que cette intention soit contenue dans le message
CS_Intent2	La deuxième intention détectée par Djingo
CS_I2_conf	La probabilité que cette intention soit contenue dans le message
CS_Intent3	La troisième intention détectée par Djingo
CS_I3_conf	La probabilité que cette intention soit contenue dans le message

TABLE 4.1 – Méta-données des conversations

Objet	Quantité
Nombre de conversations	1 231 519
Nombre de messages	6 296 631
Nombre de messages uniques	1 620 902
Nombre de tokens	29 222 593
Nombre de formes distinctes	132 851
Nombre moyen de messages de l'utilisateur par conversation	3.0
Nombre moyen de mots par message	4.6
Taille du corpus (texte uniquement)	164 MB
Taille du corpus compressé (méta données incluses)	1.76 GB
Taille du corpus décompressé (méta données incluses)	12.4 GB

TABLE 4.2 – Récapitulatif des données entre Novembre 2017 et Décembre 2018 - Les messages sont uniquement les messages clients

Nombre d'occurrences	Message les plus fréquents
113147	Non
85759	bonjour
81503	Compte bancaire
79595	Oui
57256	Bonjour
53921	Je l'ai perdu
41752	Je ne l'ai jamais reçu
33240	Je veux être mis en relation avec un expert ?
32460	Expert
28943	Merci
26797	Offre de bienvenue à l'ouverture d'un compte
21673	non
19550	Carte bancaire
16553	Code d'accès à l'espace client
16535	Autre
16486	Avantage Client Orange ou Sosh
13687	oui
13439	Quand vais-je recevoir mon avantage client Orange ou Sosh ?
13276	Paielement mobile
13128	Plafond de carte bancaire
12525	Ok

TABLE 4.3 – Messages les plus fréquents

Chapitre 5

Annotation

Ce chapitre résume les travaux effectués dans le but de catégoriser des données. Nous nous attarderons en particulier sur le guide d’annotation ainsi que sur la répartition des annotations dans le corpus.

5.1 Constitution de la liste des messages

Environ 3000 messages ont été pris au hasard dans le corpus. Les messages ont été annotés en positif/négatif/neutre par nos soins suivant le guide d’annotation détaillé plus bas.

5.2 Guide d’annotation

Ce guide d’annotation a été inspiré de celui de la tâche 1 du Défi Fouille de Texte (DEFT) 2015¹ et de celui du DEFT 2018². Les messages ont été annotés en trois catégories : positif (+), négatif (-), neutre (0). Chaque message devant appartenir à une unique catégorie. Notons que dans le guide d’annotation DEFT 2015, la catégorie ‘neutre ou mixte’ est attribuée aussi bien aux messages présentant une polarité neutre (ni positif, ni négatif), qu’à ceux présentant les deux polarités en même temps. Dans notre corpus, nous n’avons pas rencontré suffisamment d’occurrences de message contenant une polarité positive et une négative à la fois. La catégorie neutre ne réfère donc dans la très large majorité des cas qu’à des messages ayant une polarité neutre.

5.2.1 Positif (+)

Un message appartient à la classe **positif** s’il est globalement positif. Par exemple

— un signe de politesse s’il apparaît seul

exemple : « bonjour »

— des remerciements

exemple : « merci »

— discussions informelles adressés à Djingo, blagues ou plaisanteries

exemple : « quel est ton nom »

— un signe de satisfaction à l’égard d’un des services d’Orange

exemple : « Merci orange pour les 80 euros »

1. <https://hal.archives-ouvertes.fr/hal-01617180/document>

2. https://perso.limsi.fr/pap/DEFT2018/annotation_guidelines/index.html

5.2.2 Négatif (-)

- Un message appartient à la classe **négatif** s'il est globalement négatif. Par exemple :
- problèmes liés à un des services d'OrangeBank que ce soit l'application ou autre exemple : « je n'arrive pas à me connecter » - « Je n'ai pas reçu mon identifiant pour application » - « Je ne l'ai jamais reçu » - « J'ai été débité deux fois du même montant »
 - l'utilisateur ne s'est pas fait comprendre par Djingo exemple : « cela ne repond pas a la question »
 - présence d'insulte ou de juron dans le message exemple : « JE VEUX PARLER A UN EXPERT CONNARD » - « robot de merde »

5.2.3 Neutre (0)

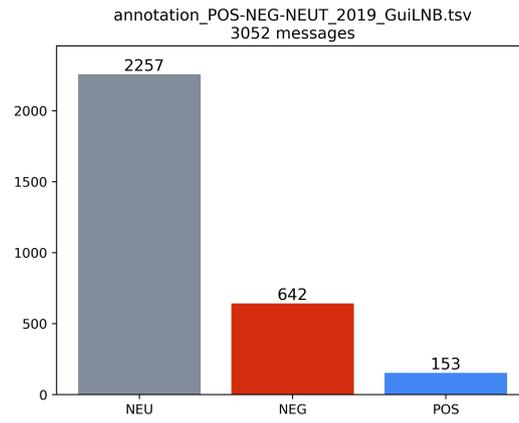
- Un message est considéré **neutre** s'il remplit l'un des critères suivants :
- mots vides ou trop courts exemple : « ok » - « oui »
 - l'utilisateur exprime une demande sur un des services d'OrangeBank eg : « Je souhaiterai ouvrir un compte » - « que faire en cas de problème avec ma carte »

5.3 Exemples d'annotation

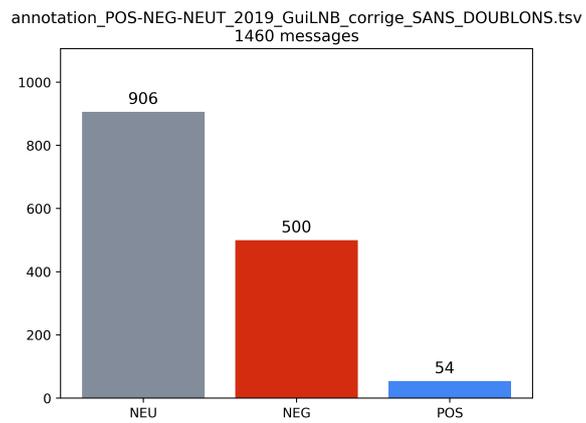
Message	Annotation
Virement non reçu	-
Bonjour	+
100 % d'échec sur mes 2 dernières tentatives. . . cela peut dissuader clairement d'utiliser la carte orange bank. . .	-
Ca va Djingo ?	+
Quels sont les frais de modification de plafonds ?	0
Pourtant je m en suis servi de la carte en février	0
VOS REPONSES SONT AUTOMATISEES ET NE CORRESPONDENT PAS DU TOUT A MA DEMANDE	-

5.4 Répartition des annotations dans le corpus

Le nombre d'annotations est réparti comme montré dans les graphiques 5.1a 5.1b. On remarque un grand nombre de messages neutres (plus de 2/3 du corpus), et une faible proportion de messages positifs. De plus, ces messages étant pris au hasard, plus de la moitié sont des doublons. On remarque en revanche que la tendance reste sensiblement la même entre le corpus avec et sans doublons. Ces doublons viennent d'une part du fait qu'un grand nombre de réponses courtes sont les mêmes pour tous les utilisateurs ('ok', 'merci', 'non', ...), et d'autre part que, dans la conversation, Djingo peut proposer des choix à des questions pré-remplies (noté ClicText dans le corpus) pour lesquelles l'utilisateur ne peut renvoyer rien d'autre qu'une des réponses proposées par Djingo ('Comment commander un chéquier?', 'Autre', 'Identifiant erroné', etc.).



(a) Répartition de l'annotation avec doublons



(b) Répartition de l'annotation sans doublons

Chapitre 6

Classification de texte par apprentissage automatique supervisé

Dans cette partie, nous traiterons la tâche de classification de texte, et la manière dont nous l'avons appliquée sur ce corpus. Nous allons nous porter tout d'abord sur une partie d'explication des représentations vectorielles, expliquerons notre choix d'algorithme de classification et nous discuterons des résultats obtenus. Nous passerons ensuite à la sélection du modèle et son amélioration, puis à l'évaluation du modèle sur d'autres jeux de données.

6.1 Représentations vectorielles

La première étape pour la classification consiste à représenter le texte des documents dans un format compréhensible pour la machine. Cela consiste à *vectoriser* le texte. C'est à dire, transformer le texte en une liste de nombres. Cette liste, appelé *vecteur* permet de rendre les manipulations plus aisées pour les algorithmes de classification.

Les vecteurs des texte disposent d'une taille, appelée *dimension* qui peut être la taille du vocabulaire (nombre de mots ou de formes du corpus), ou nombre arbitraire (par exemple 50, 100, 300) en fonction de l'algorithme de représentation utilisé. Il existe plusieurs manières de représenter des documents sous un format vectoriel. Les représentations ci-dessous ont été utilisées au cours des expériences.

6.1.1 One-hot

La représentation en *one-hot* est une représentation basique de document en fonction de la présence ou non de termes dans les textes. Chaque vecteur de document a comme dimension la taille du vocabulaire du corpus (nombre de mots ou de formes). Chaque coordonnée du vecteur correspond à la présence (1 pour un élément, 2 pour deux éléments) ou non (0) d'un terme du vocabulaire.

Exemple pour le corpus de phrases :

1. Le chat mange la souris.
2. La souris aime le fromage
3. Le chat n'aime pas le fromage.

Il y a 9 formes dans le corpus (le corpus étant passé en minuscule) :

['aime', 'chat', 'fromage', 'la', 'le', 'mange', 'n', 'pas', 'souris']

. Chaque terme dispose de son indice propre : 'aime' \rightarrow 0, 'chat' \rightarrow 1, 'fromage' \rightarrow 2, etc... On peut ainsi associer un vecteur à chaque document :

1. le chat mange la souris

[0 1 0 1 1 0 0 0 1]

2. la souris aime le fromage

[0 0 1 1 1 0 0 0 1]

3. le chat n'aime pas le fromage

[1 1 1 0 2 0 1 1 0]

Cette représentation est dite en sac de mots (Bag Of Words : BOW). Ce qui signifie que l'ordre des mots du document n'est pas pris en compte dans la représentation. Elle est généralement combinée avec la représentation $tf \times idf$ décrite dans la prochaine section. Notons tout d'abord que ces vecteurs sont généralement vides (remplis de 0) si la taille du vocabulaire est grande. Ensuite, cette représentation tient uniquement compte des mots qui sont présents dans le vocabulaire. Si le système rencontre un mot inconnu, aucun indice ne pourra être utilisé. Nous verrons plus loin une technique (le hashing trick) pouvant pallier à ce désavantage.

6.1.2 Tf-Idf

La représentation en $tf \times idf$ (la fréquence du terme (*term frequency* multiplié par la fréquence inverse du document (*inverse document frequency*)) est une technique de pondération qui permet de faire ressortir les termes discriminants d'un document. Elle fut proposée par Salton et Buckley [12] et est composée du produit de deux éléments :

TF : Terme Frequency, la fréquence du terme dans le document (qui peut être elle-même pondérée selon le nombre de terme dans le document)

IDF : Inverse Document Frequency, l'inverse du nombre de documents qui contiennent le terme.

Cette méthode permet de moins tenir compte de termes qui apparaissent dans de nombreux documents, comme par exemple les déterminants.

6.1.3 Prolongements - Word2vec

Les prolongements (*embeddings* en anglais) est une technique de représentation vectorielle qui vise à représenter les mots selon leurs contextes d'apparition. Cette technique a été décrite en 2013 par Mikolov et al. [13] et a été appelée Word2vec. À la différence d'une représentation one-hot comme vu précédemment, les vecteurs produits par word2vec permettent de tenir compte d'une certaine similarité sémantique entre des termes. Par exemple, le vecteur de « chat » devrait être proche de celui de « chien ». Ces vecteurs sont aussi denses, car ils disposent d'une dimension prédéfinie par l'algorithme. Généralement cette dimension est comprise entre 100 et 300.

Avec cette technique, il est possible de représenter une séquence de mots, comme une phrase, en s'appuyant sur leur représentation vectorielle, en sommant chaque composante des vecteurs et en le divisant par le nombre de termes additionnés. Cette technique peut être utilisée seule ou être combinée à d'autres comme nous verrons par la suite.

6.1.4 Doc2Vec

Doc2vec est une technique de représentation vectorielle de documents décrite par Quoc V. Le [14]. Cette technique est basée sur les représentations Word2Vec, mais avec pour sortie des vecteurs correspondant à des documents. Dans notre cas, les documents sont des messages. Doc2vec offre la possibilité de varier plusieurs paramètres de calcul et de sortie, comme la taille du vecteur final, ou le nombre d'itérations sur les données d'apprentissage (noté *epochs*). Environ une quarantaine de configurations ont été testées. Nous avons choisi de faire figurer trois configurations représentatives (la meilleure ainsi que deux autres) dans le tableau des résultats. Le reste des résultats est disponible dans l'annexe 9.2.

De même qu'avec word2vec, cette technique peut être utilisée comme seule représentation ou employée avec d'autres.

6.1.5 Hashing Trick

Le hashing trick est une technique qui consiste à utiliser une fonction de hachage. Cette fonction, pour une chaîne de caractères donnée, va lui associer un identifiant unique. Cet identifiant est ensuite converti en vecteur qui correspond à une représentation vectorielle de la chaîne de caractères. La dimension des vecteurs de sortie est prédéterminée à l'avance. Elle peut être variée comme hyperparamètre.

Cette technique a l'avantage de ne pas dépendre au préalable d'un vocabulaire, comme la représentation en one-hot et est moins coûteuse en terme de mémoire. Traditionnellement, le hashing trick a été utilisé en TAL pour la classification d'email en légitime/illégitime. En 2009, une équipe de Yahoo a publié une étude sur la classification en spam/ham en utilisant cette technique [15].

6.1.6 Concaténation de plusieurs représentations

Ces représentations peuvent être concaténées entre elles (mis bout à bout). Plusieurs articles utilisent cette technique de concaténation pour la classification de texte, comme par exemple dans [16], où les auteurs utilisent les vecteurs produits par word2vec en les combinant avec les vecteurs $tf \times idf$, ou dans [17] où les auteurs concatènent $tf \times idf$, hashing trick, word2vec et une représentation fasttext [18].

6.2 Choix du classifieur et Pré-traitement des données

Une fois les documents vectorisés un classifieur peut être utilisé pour catégoriser les données. Comme vu dans la section état de l'art, les SVM sont des algorithmes de classification performants en TAL. Ils ont pour avantages d'être rapides en temps de calcul, d'être efficaces pour classifier des données de grandes dimensions, et ne nécessitent qu'un faible nombre d'hyperparamètres. Une description simplifiée du fonctionnement de l'algorithme se trouve en annexe 9. Pour ces raisons, nous avons choisi d'utiliser les SVM comme classifieur. Les pré-traitements classiques suivants ont été employés : décapitalisation, dédiacritalisation (suppression des accents sur les caractères diacrités) et délétion des mots vides (les articles, les formes du verbe être, ...). Tous les algorithmes ont été testés avec et sans doublons. Le corpus annoté (avec et sans doublons) a été séparé en 2 parties : train (pour l'apprentissage) - 80% des messages du corpus, et test (pour la phase de test) - 20% des messages du corpus.

6.3 Résultats

Le tableau 6.1 résume les résultats obtenus. La représentation en $tf \times idf$ seul fournit dans trois cas les meilleurs résultats. De plus c'est une représentation peu coûteuse en terme de calcul et qui est donc rapide à mettre en place. Nous avons donc choisi d'approfondir avec cette représentation vectorielle.

Représentation utilisée	Sans doublons		Avec doublons	
	f-mesure macro	f-mesure micro	f-mesure macro	f-mesure micro
$tf \times idf$ seul	0.80	0.88	0.89	0.93
CBOW Word2Vec seul	0.64	0.78	0.80	0.89
$tf \times idf$ + CBOW Word2Vec	0.75	0.83	0.84	0.91
Doc2Vec seul $vs=50$, $w=7$, $mincount=2$, $epochs=200$	0.56	0.84	0.56	0.86
$tf \times idf$ + Doc2Vec	0.81	0.87	0.56	0.86
Hashing Trick Seul $n=100$	0.69	0.79	0.80	0.88
Hashing Trick Seul $n=1000$	0.79	0.85	0.88	0.92
Hashing Trick Seul $n=10000$	0.80	0.86	0.88	0.92
$tf \times idf$ + Hashing Trick $n=1000$	0.83	0.87	0.88	0.93
$tf \times idf$ + Doc2Vec + Hashing Trick $n=1000$	0.81	0.88	0.88	0.93
$tf \times idf$ + Doc2Vec + CBOW Word2Vec + Hashing Trick $n=1000$	0.71	0.83	0.85	0.91

TABLE 6.1 – Résultats obtenus en fonction des représentations vectorielles

'+' signifie la concaténation des représentations vectorielles des messages

vs =taille du vecteur, w =la fenêtre des mots telle que décrite dans la documentation de `doc2vec`, $epochs$ =le nombre d'itérations (epochs), n =la dimension du vecteur produit par le hashing trick

6.4 Sélection du modèle

6.4.1 Grid search

Afin de déterminer la meilleure combinaison d'hyper paramètres, nous avons effectué un grid search sur le fichier contenant les doublons. Cette technique consiste à faire varier les paramètres un à un afin de trouver la meilleure combinaison possible. Cette méthode est coûteuse en ressources, mais permet d'obtenir des valeurs d'hyperparamètres donnant la meilleure performance sur le jeu de données concerné.

Liste des pré-traitements classiques

- La casse : deux configurations sont testées : mise en casse minuscule de l'ensemble du texte ou non
- Dédiacritisation : dédiacritisation (suppression des accents) sur le texte ou non
- Délétion des mots vides : suppression des mots vides (articles, déterminants, ...) ou non¹

1. Obtenus via NLTK

- Tokenization et lemmatization : plusieurs tokenizers/lemmatizers ont été testés : blob² et NLTK³

Liste des pré-traitements liés aux données

Nous avons ensuite pensé que, dans certaines conversations, la présence de certains éléments (émoji, insultes, ...) devait être un élément discriminant pour classer le message correctement. Ainsi, plusieurs balises ont été ajoutées à l'intérieur des messages afin de faciliter la classification.

- Présence de points d'exclamations (plus de 5 côte à côte) ou non : dans les conversations, la présence d'un grand nombre de points d'exclamation traduit très souvent une irritation de l'utilisateur, donc une opinion négative. Nous avons donc ajouté une balise dans le texte correspondant à ces marqueurs dans le texte. *exemples : 'Je ne parviens pas à signer mon contrat apres plusieurs essais je suis interdit d'ouverture!!!!!!!!' - 'pas moyen d'ouvrir un compte!!!!!!!!!!!!!! c'est une vraie galère!!!!!!!!!!!!!!!!!!!!', 'Il est impossible de s'inscrire depuis 5 jours!!!!!!!!'*
- Présence d'insultes ou de jurons ou non : de même, la présence d'insulte dans un message est un critère discriminant. Comme le nombre d'insultes de la langue est assez important, plutôt que d'annoter tous les messages qui contiennent une insulte, nous avons préféré constituer une liste d'insultes (via wikipedia⁴ et d'autres ressources⁵) et remplacer les insultes du messages par un marqueur commun aux insultes/jurons. Cette méthode fait indéniablement perdre en compréhension, vu que des mots différents sont remplacés par le même marqueur, mais elle présente l'avantage de traiter de la même manière des mots qui n'auraient pas été vus.
- Casse : de même, nous avons observé que les messages écrits entièrement en majuscule disposaient tous d'une polarité négative. Nous avons donc inséré une balise relative à la casse si tous les mots du message étaient en majuscule.
- Émoji/émoticon/kamoji : Enfin, sachant que les messages contenant un symbole positif ont tous été annotés de manière positive (de même pour les négatifs), nous avons procédé de la même manière en remplaçant tous les émojis/émoticons/kamojis positifs par un marqueur lié à ces symboles, de même que pour les négatifs.

Ces trois configurations ont été testée ensemble, afin de diminuer le temps de calcul.

Liste des hyperparamètres de la représentation vectorielle et du SVM

La librairie python sklearn [19] a été utilisée pour ces expériences. Nous avons fait varier les paramètres suivants :

- max df : un nombre en 0 et 1. Permet d'ignorer les mots dont la fréquence est supérieure à ce seuil. (valeur testées : 0.5, 0.75, 0.90, 0.95, 1.0)
- min df : de même. Permet d'ignorer les mots dont la fréquence est inférieure à ce seuil (valeur testées : 0.0, 0.05, 0.1, 0.25)
- type de noyau du SVM : linéaire ou sigmoïde

Au total, 8 paramètres ont été variés, pour un total de 1280 combinaisons possibles.

2. <https://textblob.readthedocs.io/en/dev/>

3. <https://www.nltk.org/api/nltk.tokenize.html>

4. https://fr.wiktionary.org/wiki/Cat%C3%A9gorie:Insultes_en_fran%C3%A7ais

5. <https://www.lovefrance.info/french-swear-words/>

6.4.2 Résultats

Nous présentons les meilleurs résultats dans la table 6.2. Ceux ci ont été obtenus avec les hyper-paramètres suivants : Tokenization et lemmatization : NLTK, noyau : linéaire, casse : gardée, dédiacritisation : non, suppression des mots vides : non, max df : 1.0, min df : 0.0.

Les meilleurs résultats ont été obtenus pour la classe neutre (0.96 de f-mesure), ce qui peut s'expliquer par le fait que cette classe dispose du plus grand nombre de données. La classe positive obtient aussi un bon score (0.88 en f-mesure), et la classe négative, une f-mesure de 0.86. Notons aussi une différence sensible entre les moyennes micro et macro.

Nous avons remarqué que, si moins de traitements étaient appliqués au texte, les résultats étaient meilleurs. Nous avons donc choisi de n'appliquer aucun traitement au texte.

	F-mesure	Précision	Rappel	Support
Positif	0.88	1.00	0.79	28
Négatif	0.86	0.90	0.82	124
Neutre	0.96	0.94	0.98	459
Moyenne micro	0.94	0.94	0.94	611
Moyenne macro	0.90	0.95	0.86	611
Moyenne pondérée	0.93	0.94	0.94	611

TABLE 6.2 – Résultats du grid search

Matrices de confusion

En nous intéressant aux matrices de confusion, nous remarquons que pour ces données d'entraînement, le modèle ne commet pas de 'grosses' erreurs. Concernant les messages positifs et négatifs, s'ils sont mal prédits, ils sont prédits comme étant neutres.

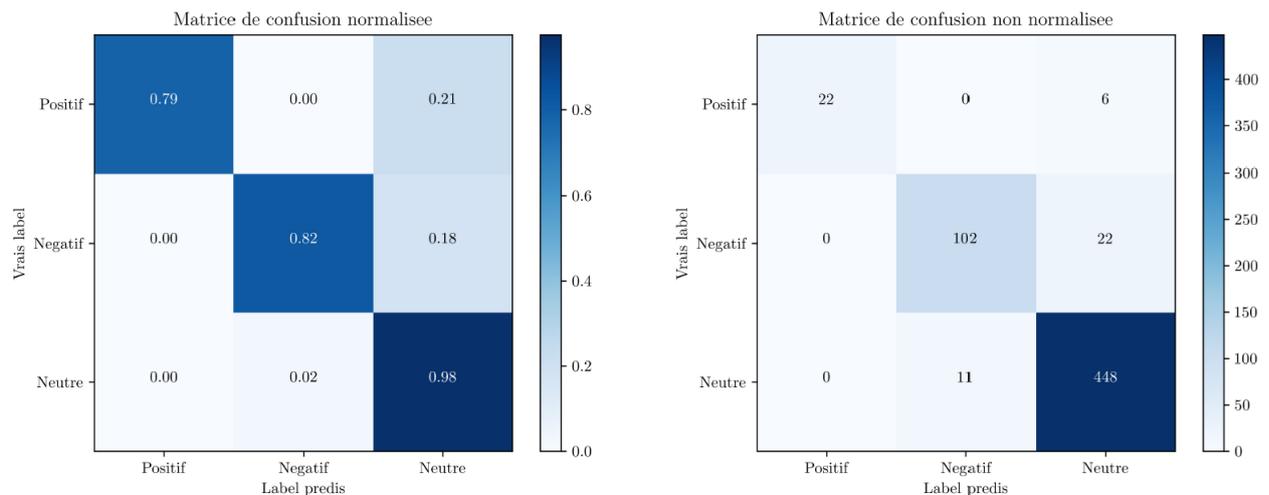


FIGURE 6.1 – Matrices de confusion du modèle

6.4.3 Compréhension du modèle

Courbes d'apprentissage

Nous nous sommes aussi intéressés aux courbes d'apprentissages du modèle. Sur le graphique ci-dessous, nous remarquons que l'algorithme du SVM converge rapidement. Environ 500 messages sont nécessaires pour obtenir des scores corrects, puis les scores de l'algorithme augmentent doucement. Nous notons une augmentation mineure du rappel à partir de 1600 messages. Avec ces courbes, nous pouvons conclure qu'apporter plus de données à ce modèle augmenterait potentiellement ces scores, mais ne permettrait pas une amélioration substantielle des scores déjà corrects.

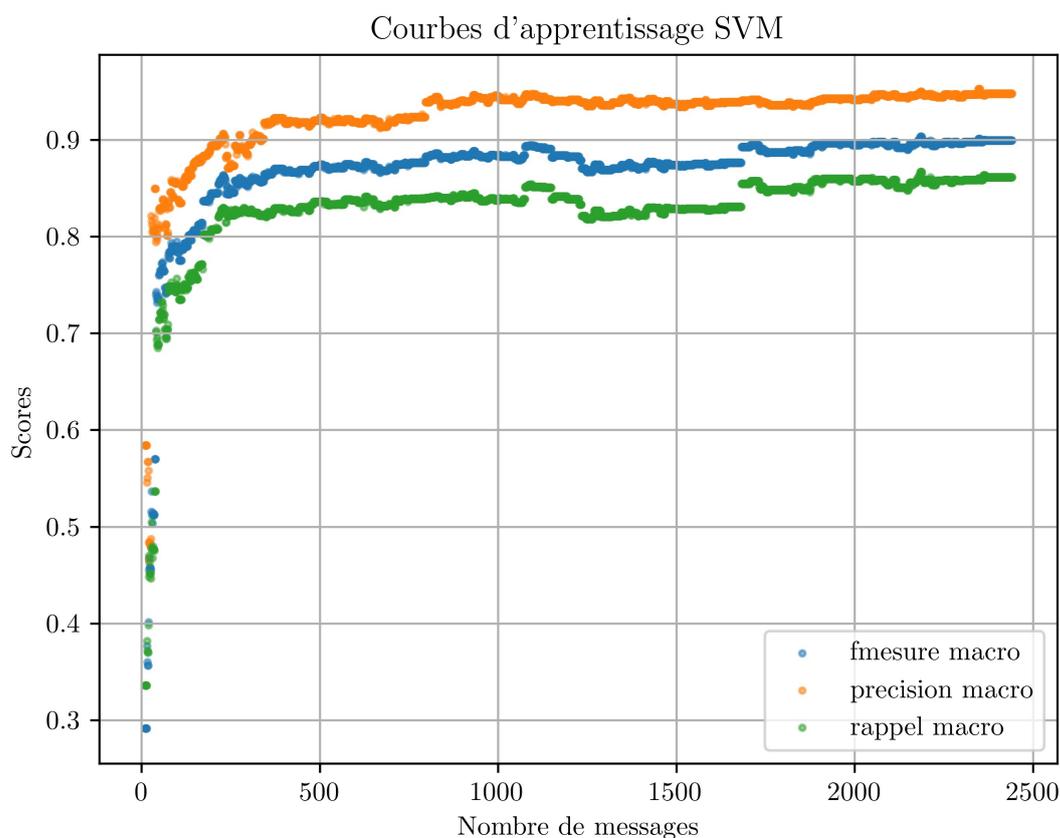


FIGURE 6.2 – Courbes d'apprentissage du modèle $tf \times idf$ seul

Explication des prédictions

Nous avons utilisé Lime [20] afin de mieux comprendre les critères discriminants du classifieur. Nous allons nous concentrer sur quatre exemples, deux erreurs et deux réussites. Ces exemples sont disponibles en annexe 9. Comme vu dans la partie grid search 6.4.1, le modèle fonctionne mieux sans aucun pré-traitement. Par conséquent, aucun traitement sur le texte n'a été appliqué.

Dans le premier exemple, un utilisateur de Djingo lui pose la question : "Bonjour je tente désespérément d'ouvrir un compte". Ce message dispose d'une polarité négative, mais a été labelisé comme neutre. Le mot 'Bonjour' oriente le message vers une polarité

positive, car dans les exemples d'entraînement, la plupart des messages contenant 'Bonjour' étaient labélisés comme ayant une polarité positive. Ensuite, nous pouvons penser que les mots 'tente' et 'désespérément' sont des critères discriminants importants. Néanmoins, le classifieur n'a pas tenu compte de ces mots, absents dans le vocabulaire du corpus d'entraînement. Par conséquent, en tenant compte des autres mots du message, notre modèle a commis une erreur dans sa catégorisation.

C'est le même type d'erreur que nous retrouvons dans le deuxième exemple. L'utilisateur a posé à Djingo la question : "Je suis mécontent". Ce message a été labélisé comme neutre par notre modèle, alors qu'il dispose d'une polarité négative. Le mot 'mécontent' pourrait être un mot discriminant, néanmoins comme celui-ci étant absent du vocabulaire d'entraînement, notre modèle l'a ignoré. Le modèle a ensuite utilisé les mots référencés comme "je" et "suis" car ils sont souvent présents dans des messages labélisés positifs. Notre modèle a donc classé ce message comme positif.

Dans le troisième exemple, notre modèle a prédit avec succès la classe du message "Tu connais des blagues?" (polarité positive)⁶. De la même manière que dans l'exemple précédent, notre modèle ne connaissait pas tous les mots du message et s'est donc rattaché aux mots connus. Nous pouvons voir que le mot 'Tu' est très discriminant (orienté très positif). Nous en expliquons la cause de la même manière que précédemment, à savoir que les messages contenant "Tu" sont la plupart labélisés comme positifs.

Enfin, dans le quatrième exemple, le modèle classifie correctement le message "Justement, je n'y arrive pas. . ." comme négatif. Pour ce faire, il s'est basé majoritairement sur le mot 'pas' (orienté vers le négatif). Ceci peut sembler cohérent d'une certaine mesure : un utilisateur employant la négation veut certainement dire qu'il n'arrive pas à réaliser une action ou communique sur un dysfonctionnement. Cet exemple illustre pourquoi les résultats du modèle étaient meilleurs sans déletion des mots vides (pas, ne, tu, ...) : ces mots disposent d'un poids important dans la polarité.

6.4.4 Test sur d'autres jeux de données

Données DEFT 2015

Afin de vérifier la validité de ce modèle, nous avons choisi de le tester sur les données de la tâche 1 du Défi Fouille de Textes (DEFT) 2015. La tâche 1 consistait à détecter la polarité des tweets parmi trois valeurs possibles : positif, neutre ou mixte, et négatif. Pour ce faire, un corpus d'entraînement et de test ont été donnés (7512 tweets en entraînement contre 3284 tweet en test). Le meilleur score obtenu pour cette tâche était de 0.736 macro précision (équipe LIF).

Nous avons donc élaboré deux approches pour valider notre méthode.

La première est de réutiliser le modèle tel quel sur le corpus de tweet, et regarder si les scores obtenus via les prédictions du modèle sont satisfaisants.

La deuxième approche consiste à ré-entraîner un modèle sur les données DEFT avec les mêmes paramètres que nous avons choisis (tf × idf + SVM), et de comparer nos résultats avec les scores des équipes de DEFT. Les résultats figurent dans les tableaux

6. Djingo connaît effectivement trois plaisanteries.

6.3 et 6.4.

Nous remarquons que les scores du modèle utilisé directement sont assez faibles (0.50 en macro précision, 0.30 en f-mesure macro). Cette perte en performance comparée au score du modèle sur nos données peut s’expliquer par le fait que la nature de notre corpus reste très différente de celle du corpus DEFT (recueil de questions d’utilisateur de chabot du milieu bancaire vs tweets).

Les scores obtenus en ré-entraînant le modèle sur les données DEFT sont passables (0.67 en macro précision 0.64 en f-mesure), mais se rapprochent des meilleurs résultats du DEFT (0.74 en macro précision).

	F-mesure	Précision	Rappel	Support
Positif	0.02	0.75	0.01	1015
Négatif	0.28	0.29	0.27	762
Neutre	0.59	0.47	0.80	1463
Moyenne micro	0.43	0.43	0.43	3240
Moyenne macro	0.30	0.50	0.36	3240
Moyenne pondérée	0.34	0.51	0.43	3240

TABLE 6.3 – Scores complets du modèle $tf \times idf$ en réutilisant le modèle sur les données DEFT 2015

	F-mesure	Précision	Rappel	Support
Positif	0.61	0.63	0.59	1015
Négatif	0.59	0.72	0.49	762
Neutre	0.72	0.66	0.79	1463
Moyenne micro	0.66	0.66	0.66	3240
Moyenne macro	0.64	0.67	0.63	3240
Moyenne pondérée	0.65	0.66	0.66	3240

TABLE 6.4 – Scores complets du modèle $tf \times idf$ en ré-entraînant le modèle sur les données DEFT

Données Hackatal 2018

Enfin, nous avons voulu tester notre approche sur un dernier jeu de données. La tâche du hackatal 2018⁷ consistait à classer des articles de presse selon trois catégories : fausse (fakeNews), certifiée (trusted) et satire. Pour ce faire, un modèle de type $tf \times idf + SVM$ a été entraîné sur le corpus d’entraînement et testé sur le corpus de test. Les résultats qui suivent nous confirment que ce type de modèle fonctionne correctement pour la tâche de classification de textes.

7. <https://github.com/HackaTAL/2018>

	F-mesure	Précision	Rappel	Support
fakeNews	0.83	0.78	0.89	28
trusted	0.92	0.95	0.89	66
Moyenne micro	0.89	0.89	0.89	94
Moyenne macro	0.88	0.87	0.89	94
Moyenne pondérée	0.90	0.90	0.89	94

TABLE 6.5 – Scores complets du modèle $tf \times idf$ en ré-entraînant le modèle sur les données hackatal 2018

Chapitre 7

Export du modèle

Après ces tests, nous avons jugé le modèle suffisamment performant, et nous l'avons utilisé sur l'intégralité du corpus, à savoir tous les messages contenus dans les conversations entre novembre 2017 et décembre 2018. Dans ce chapitre, nous verrons nos premières approches en matière de visualisation de l'évolution de l'opinion, tout d'abord à la maille des messages, puis à la maille des conversations, le tout sur différentes échelles temporelles.

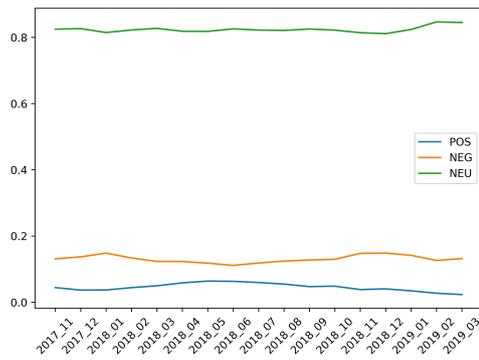
7.1 Évolution de la polarité au cours du temps

Une des premières applications de la connaissance de la polarité des messages est de pouvoir visualiser une évolution au cours du temps de celle-ci. Cette évolution peut être visualisée à l'échelle de l'année (table 7.1a), du mois (table 7.1b), du jour (table 7.1c ou d'une autre échelle de temps. La méthode est restée la même à chaque fois : pour chaque intervalle de temps, nous récupérons le nombre de messages appartenant à chaque classe (positif, négatif et neutre) et nous le normalisons en pourcentage. Nous pouvons ensuite tracer les courbes qui correspondent ¹.

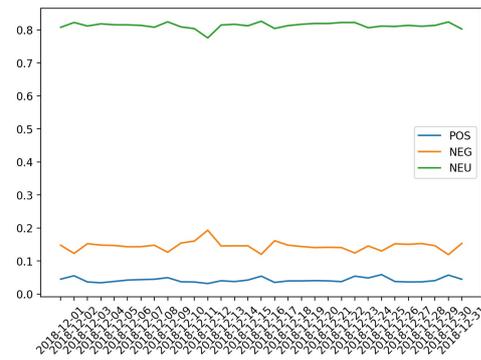
Nous avons ensuite cherché à établir des corrélations entre ces données et d'autres sources. Nous nous sommes tournés vers le taux de re-direction vers le centre de relation client (CRC). Cette mesure correspond au pourcentage des conversations de Djingo qui sont re-dirigées vers le CRC (à la suite d'incompréhension du chatbot ou son manque de qualification pour effectuer certaines tâches) ². Tout d'abord de manière qualitative, on remarque que les courbes à l'échelle du mois ressemblent fortement aux courbes de re-direction (plus bas).

1. Les labels sur les axes des ordonnées correspondent au pourcentage ramené entre 0 et un eg : 0.8 = 80%

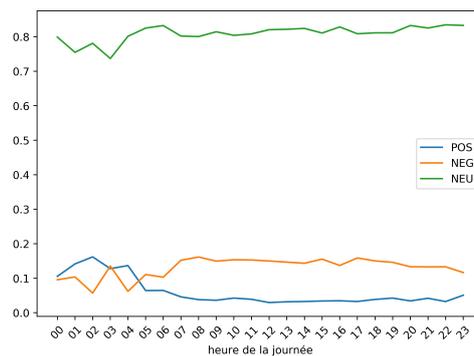
2. Le taux de re-direction vers le CRC oscille autour de la barre des 50% entre janvier 2019 à avril 2019.



(a) Évolution de l'opinion entre septembre 2017 et janvier 2019



(b) Évolution de l'opinion pour le mois de décembre 2018



(c) Évolution de l'opinion pour la journée du 13 décembre 2018

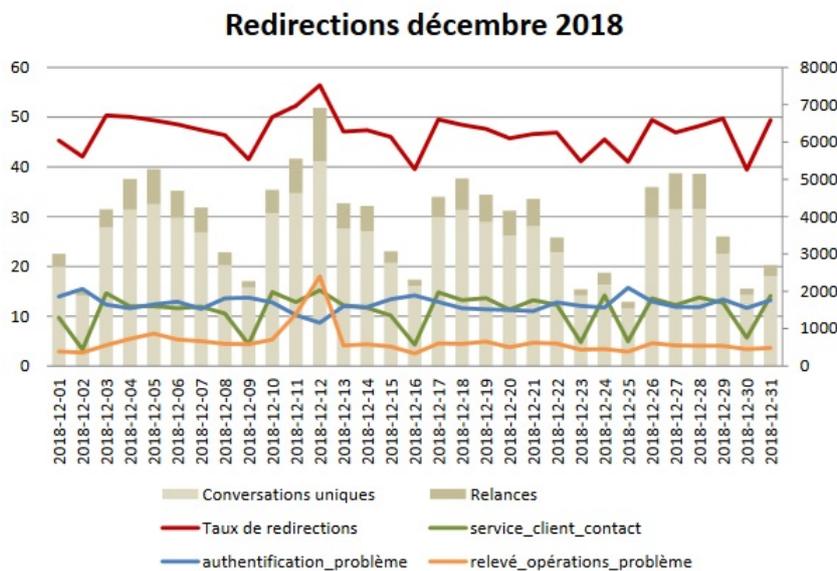


FIGURE 7.2 – Taux de re-direction vers le CRC

Nous avons cherché à prouver cette corrélation à l'échelle de l'année en utilisant des outils statistiques. Tout d'abord, plutôt que de continuer avec des pourcentages de messages appartenant à chaque classe (positif, négatif, neutre), nous avons décidé de les combiner afin d'obtenir un score de polarité avec lequel il serait plus aisé de faire ces tests. Ce *score de polarité* est donné par la somme du nombre de messages neutres et positifs moins le nombre de messages négatifs. Il est ensuite normalisé en pourcentage. Nous avons

ensuite calculé la corrélation entre ce score et le taux de re-direction vers le CRC³ via le coefficient de Pearson⁴, en considérant que, s'il y avait une corrélation, elle serait d'ordre linéaire. Nous avons aussi utilisé la p-value pour déterminer la fiabilité de nos résultats.⁵

En calculant cette corrélation, nous sommes arrivés à un coefficient r de -0.620578, ce qui indique une corrélation linéaire négative. Plus le score d'opinion est élevé (plus d'opinion positive), moins il y aura de redirections. La p-value étant de 0.0078575 % donc inférieure à 5%, nous avons considéré ce résultat comme acceptable. Le graphique 7.3 montre la relation entre ces deux variables. Le détails des données est disponible dans l'annexe 9.4.

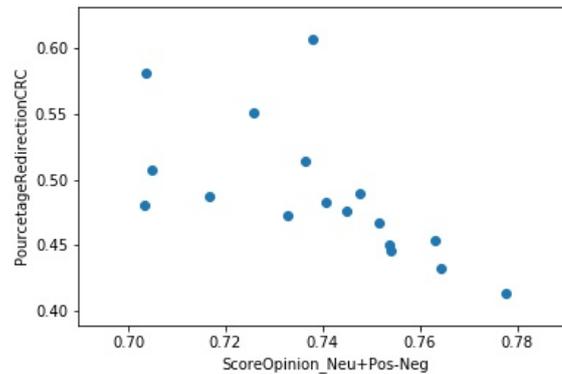


FIGURE 7.3 – Diagramme de dispersion entre le taux de re-direction vers le CRC et le score d'opinion calculé par la somme du nombre de messages (Neutre + Positif - Négatif)

p-value : 0.0078575

r : -0.620578

Sur le graphique, nous remarquons, que plus le score d'opinion est bas (proportion de messages négatifs importante), plus le taux de re-direction va être élevé. Nous pouvons donc conclure qu'à l'échelle de l'année, le score de polarité est corrélé au taux de redirection vers le CRC. Le modèle de détection de la polarité est suffisamment fiable pour pouvoir confirmer l'intuition selon laquelle les redirections sont liées aux polarités négatives.

Enfin pour revenir aux courbes, sur le graphique de la journée du 13 décembre 2018, nous remarquons une proportion plus importante de messages positifs entre minuit et six heures du matin. Nous tentons d'expliquer ce pic par le fait que les messages de type *bavardages*, généralement classés comme positifs par notre modèle sont plus présents en terme de proportions dans ce créneau horaire, d'une part parce que ces messages sont plus nombreux en terme de quantité, mais aussi parce que les messages portants sur les interactions bancaires restent moins nombreux. Exemple de messages identifiés comme positifs envoyés à Djingo entre 0h et 6h le 13 décembre 2018 : "*Robot, robot, quand tu*

3. Les conversations peuvent être redirigées vers un conseiller pour plusieurs raisons : Djingo n'a pas compris la question posée, n'est pas capable de répondre à la question posée ou n'est pas compétent pour répondre à la question (cas d'opérations bancaires demandant une habilitation) ou autre cas.

4. Le coefficient de Pearson (noté r) est une mesure de la corrélation linéaire entre deux variables. Ce coefficient dispose d'une valeur comprise entre +1 et -1, où 1 est la corrélation linéaire positive totale, 0 est l'absence de corrélation linéaire et -1 est la corrélation linéaire négative totale.

5. La p-value est un nombre compris entre 0 et 1, qui indique le niveau de preuves contre l'hypothèse nulle. Dans notre cas, l'hypothèse nulle peut s'énoncer de la sorte : il n'y a pas de corrélation entre le score de polarité et le taux de re-direction. Si la p-value est inférieure à 5%, on rejette alors cette hypothèse.

nous tiens. . .", "Tu as quel age", "tu as un joli coupe de cheveux", "tu connais noemie", "tu es beau", "tu es bête", "tu es bien habillé aujourd'hui", "tu es bien joli", "Je m'ennuie tu connais une blague", "tu es le meilleur", "Tu fais quoi?", "comment t'appelles-tu?", ...

7.2 Évolution de la polarité des conversations en fonction des thèmes

7.2.1 Passage à la maille de la conversation

Nous avons ensuite choisi de croiser les informations sur la polarité des messages avec les thèmes détectés dans les conversations par Djingo afin d'obtenir des visuels plus informatifs. Dans ce cas-ci, nous avons décidé de travailler à l'échelle de la conversation. Pour passer d'une annotation d'un message à une conversation, nous avons suivi la règle suivante : une conversation est positive si elle contient au minimum un message positif et aucun message négatif, inversement pour une conversation négative. Elle peut en revanche contenir des messages neutres. Une conversation est neutre lorsque tous les messages qui la composent sont neutres. Pour les conversations contenant des messages positifs et négatifs, nous avons choisi de la labéliser comme étant mitigée.

En utilisant cette règle, cela nous permet de détecter des conversations très marquées (très positives ou très négatives). Une conversation n'ayant que de messages positifs sera sans doute jugée comme très positive par une personne, alors que si une conversation ne contient qu'un très faible nombre de messages positifs et le reste des messages neutres, elle ne sera jugée que comme étant "un peu" positive. Des exemples de ces conversations sont disponibles dans l'annexe 9.5.

Nous avons ensuite cartographié ces conversations dans les graphiques ci-dessous. Les thèmes simplifiés ont été obtenus en fusionnant des thèmes similaires (par exemple les thèmes 'credit_info', 'credit_resiliation', 'credit_souscription_delai', 'credit_souscription_info', 'credit_souscription_probleme' ont été fusionnés dans le thème 'credit').

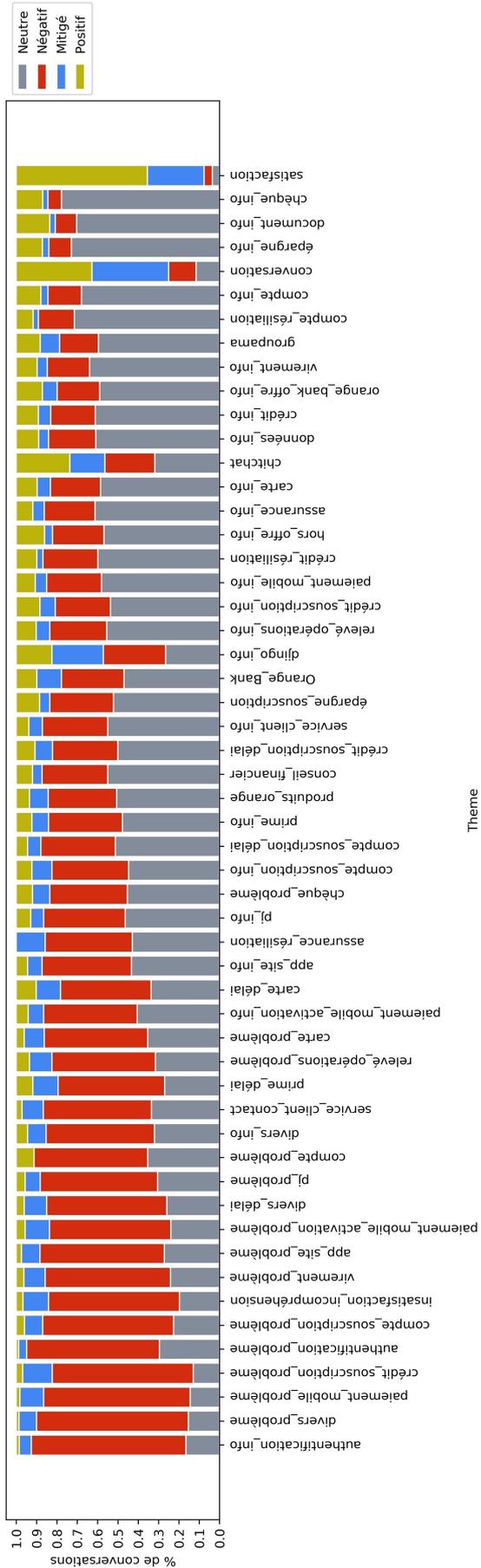


FIGURE 7.4 – Répartition de l'opinion en fonction des thèmes pour le mois de décembre 2018

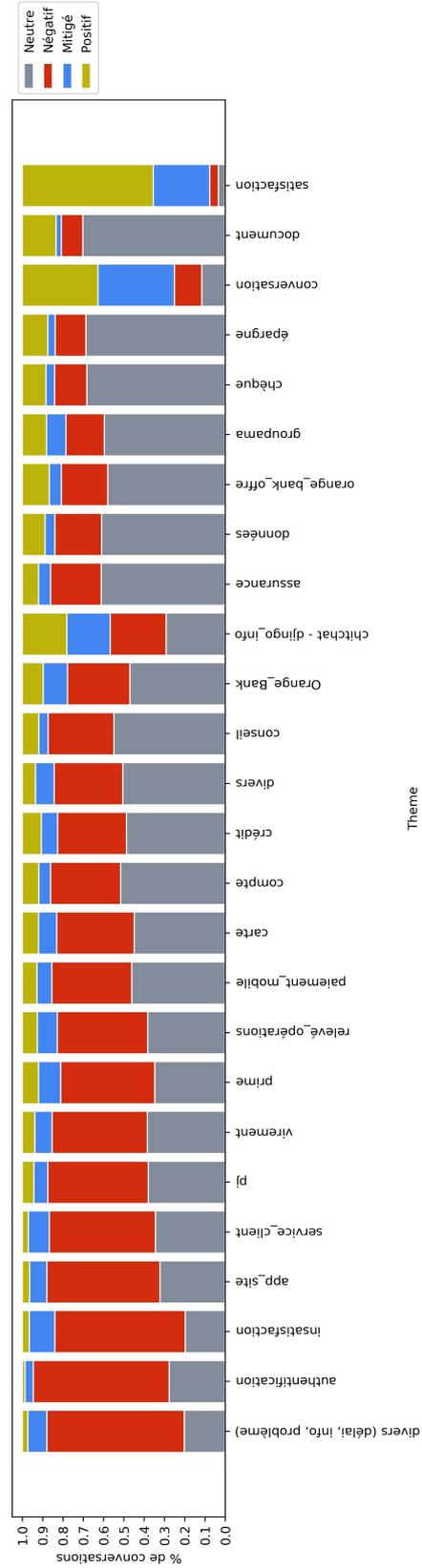


FIGURE 7.5 – Répartition de l'opinion en fonction des thèmes pour le mois de décembre 2018 avec thèmes simplifiés

Cette visualisation nous renseigne sur la proportion de type de conversations (conversation positive, négative, ...) par rapport aux thèmes. Nous pouvons voir par exemple que les conversations liées à la thématique du paiement mobile ('paiement_mobile') disposent d'une proportion de conversations négatives plus importante que celles du chèque. Si l'on fait l'hypothèse que la proportion de conversations négatives est liée à l'insatisfaction client, nous pouvons en déduire que les clients d'Orange Bank sont plus insatisfaits des services liés au paiement mobile que des services liés au chèque.

Pour confirmer cette hypothèse, nous aurions besoin d'une mesure extérieure pour quantifier la satisfaction client par thèmes. Jusqu'à maintenant, nous ne disposons pas d'une telle métrique au sein de l'entreprise.

7.2.2 Visualisation en carte thermique

Ce type de visualisation peut s'extrapoler sur une échelle de temps (mois, année, etc.) et peut être aussi utilisé pour représenter l'évolution de l'opinion par thème en fonction d'une période de temps. Pour ce faire, nous avons réutilisé la technique de calcul du score d'opinion en l'appliquant aux conversations ($Nb\ conversations\ Neutres + Nb\ conversations\ Positives - Nb\ conversations\ Négatives$). Les conversations mitigées ont été ignorées. Nous avons donc obtenu un score d'opinion par thème et par date qui peut être visualisé sous forme de *heatmap* ou carte thermique. Ces visualisations ont été produites avec tous les thèmes et les thèmes simplifiés sur trois différentes échelles (année, mois, jour). Elles sont disponibles dans l'annexe 9.3.

Pour lire ces heatmaps, l'axe temporel se situe en ordonnée, le plus récent en haut. Les thèmes sont répertoriés sur l'axe horizontal. La couleur correspond au score d'opinion. Plus une case tend vers le rouge, plus le taux de conversations négatives sera important, pour cet intervalle de temps et ce thème. Inversement, si une case tend vers le bleu, la proportion de conversations positives sera importante. Nous retrouvons ici le même ordre des thèmes, mais cette visualisation nous indique des variations dans la polarité des conversations liées à certains thèmes. Par exemple, nous pouvons voir que le thème simplifié 'app_site' dispose d'un pic de conversations négatives en avril 2004.

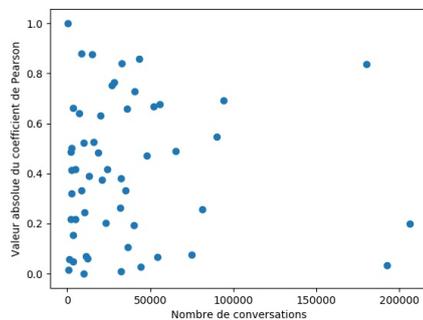
Les heatmaps à l'échelle du jour et du mois ne nous donnent pas pleinement satisfaction, tous les thèmes des conversations ne sont pas abordés chaque jour. Nous avons donc choisi nous concentrer sur les analyses à l'échelle de l'année.

De la même manière que précédemment, nous avons analysé la corrélation entre le score d'opinion avec thèmes et le taux de redirection du CRC en fonction des thèmes. Nous avons donc suivi la même méthode que précédemment en considérant que, s'il y avait une relation, elle serait d'ordre linéaire. Ces résultats sont disponibles dans l'annexe 9.3. Sur les 54 thèmes, une vingtaine dispose d'une p-value inférieure à 0.05 et d'un coefficient de Pearson éloigné de 0. On peut citer entre autre 'virement_problème', 'orange_bank_offre_info', 'divers_délais', 'chèque_info', 'chèque_problème' etc. Ces deux derniers thèmes se situent respectivement au-dessus et en-dessous du nombre moyen de conversations par thèmes. Les thèmes restants ne disposent pas de corrélation entre leur score d'opinion à l'échelle de la conversation et leur taux de redirection vers le CRC.

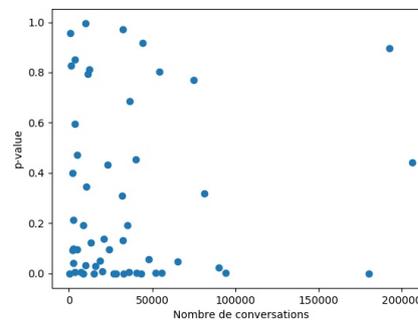
D'autres relations entre les thèmes sont en revanche plus difficiles à interpréter comme par exemple 'authentification_info' qui dispose d'un lien fort entre son taux de redirection et son score d'opinion (p-value < 0.05 et r=-0.84), mais dont un thème proche 'authentification_problème' ne dispose a priori pas de corrélation (p-value > 0.05 et r

proche de zéro). Nous tentons d'expliquer cette divergence par le fait que nous sommes passés à la maille de la conversation et que nous avons volontairement oublié les conversations mitigées. Nous avons tracé certains diagrammes de dispersion entre le taux de re-direction vers le CRC et le score d'opinion pour les thèmes les plus représentatifs. Ces graphiques sont disponibles dans l'annexe 9. Nous avons ensuite cherché à savoir s'il y avait un biais dans nos données, en regardant si le nombre de conversations influait le coefficient de Pearson ou la p-value obtenue précédemment. Les graphiques correspondants se trouvent ci-dessous, et ne montrent pas de corrélation entre le nombre de conversation par thèmes et les résultats obtenus précédemment.

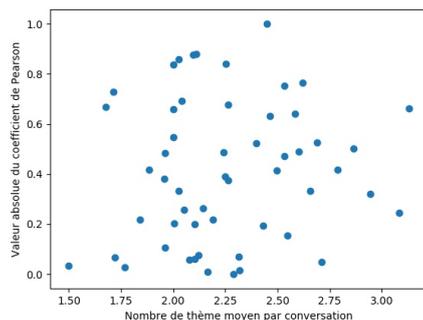
Nous avons aussi analysé l'influence du nombre de thèmes contenus dans les conversations (les conversations peuvent contenir plusieurs thèmes) et les scores liés au thème (r et p-value). De même, nous avons tracé les graphiques correspondants, mais n'avons pas pu conclure à une corrélation entre ces deux mesures.



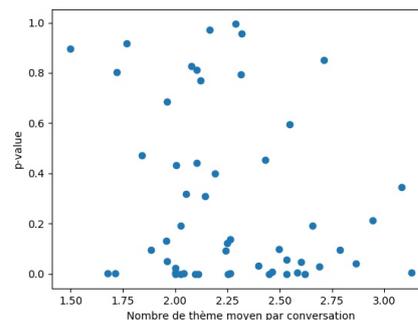
(a) Valeur du coefficient de Pearson par nombre de conversation.
 $r \approx 0.05$, p-value ≈ 0.72



(b) Valeur de la p-value par nombre de conversation.
 $r \approx -0.01$, p-value ≈ 0.97



(a) Valeur de r par nombre de thèmes présents dans la conversation.
 $r \approx 0.12$, p-value ≈ 0.41



(b) Valeur de la p-value par nombre de thèmes présents dans la conversation.
 $r \approx -0.22$, p-value ≈ 0.10

Vis à vis des thèmes, nous pouvons tirer de ces résultats qu'a priori, le score d'opinion et le taux de redirection ne sont pas toujours corrélés.

Chapitre 8

Évolution de la polarité au fil de la conversation

Nous avons ensuite cherché à caractériser l'évolution de la polarité au sein des conversations avec et sans redirection. Pour ce faire, nous avons opté pour une visualisation en graphe représentant le déroulement de la conversation. Sur l'année 2018, nous avons récupéré les conversations ayant été transférées à un agent, et les conversations avec Djingo uniquement. Nous avons ensuite converti chaque message en une liste d'entiers (négatif : 0, neutre : 1, positif : 2), ceci permettant d'obtenir un premier type de graphe. L'exemple ci-dessous représente les étapes précédentes.

Message	Polarité prédite	Score converti
Bonjour je suis nouveau client mais je n'ai pas fait la premier connexion	négatif	0
Aucun	neutre	1
Car je ne pas fait la demande de carte bancaire car je ne pas fait la demande de carte bancaire	négatif	0
Oui mais comment je fait pour me connecter	neutre	1
l'application Orange Bank	neutre	1
Oui	neutre	1
D'accord merci	positif	2

TABLE 8.1 – Exemple de conversation convertie en graphe

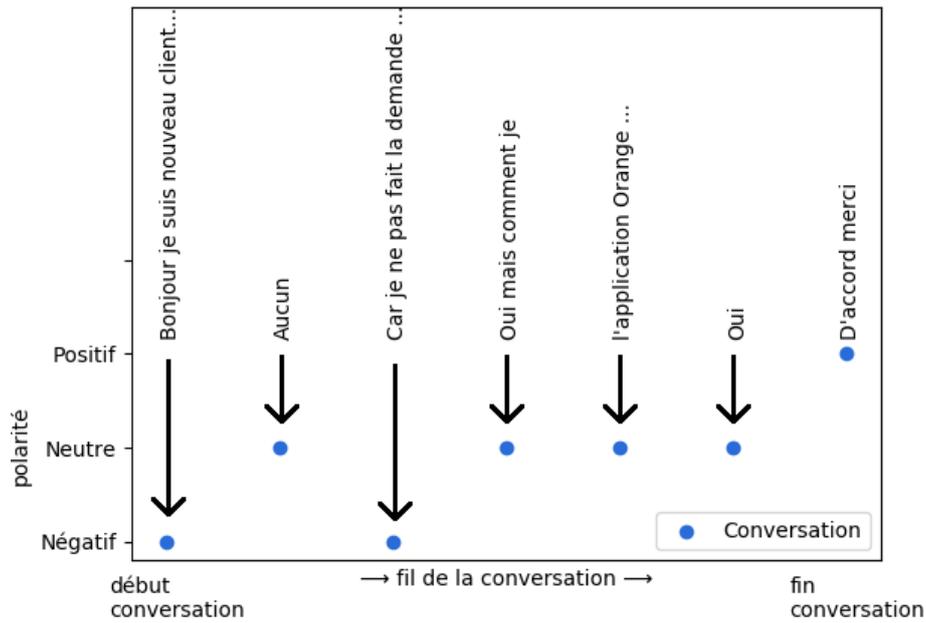


FIGURE 8.1 – Représentation basique de l'évolution de la polarité au cours de la conversation

Nous avons ensuite cherché à tracer le graphique de la moyenne de toutes les conversations. Sachant que les conversations ne font pas toutes la même taille, nous avons élaboré un algorithme permettant d'homogénéiser la longueur des listes vers une taille de liste donnée. Ce paramétrage nous fournit des résultats différents en terme de granularité, mais qui montrent tous la même tendance de fond. Le fonctionnement de l'algorithme est donné en annexe 9. Nous avons ajouté la valeur des écarts-types pour chaque point.

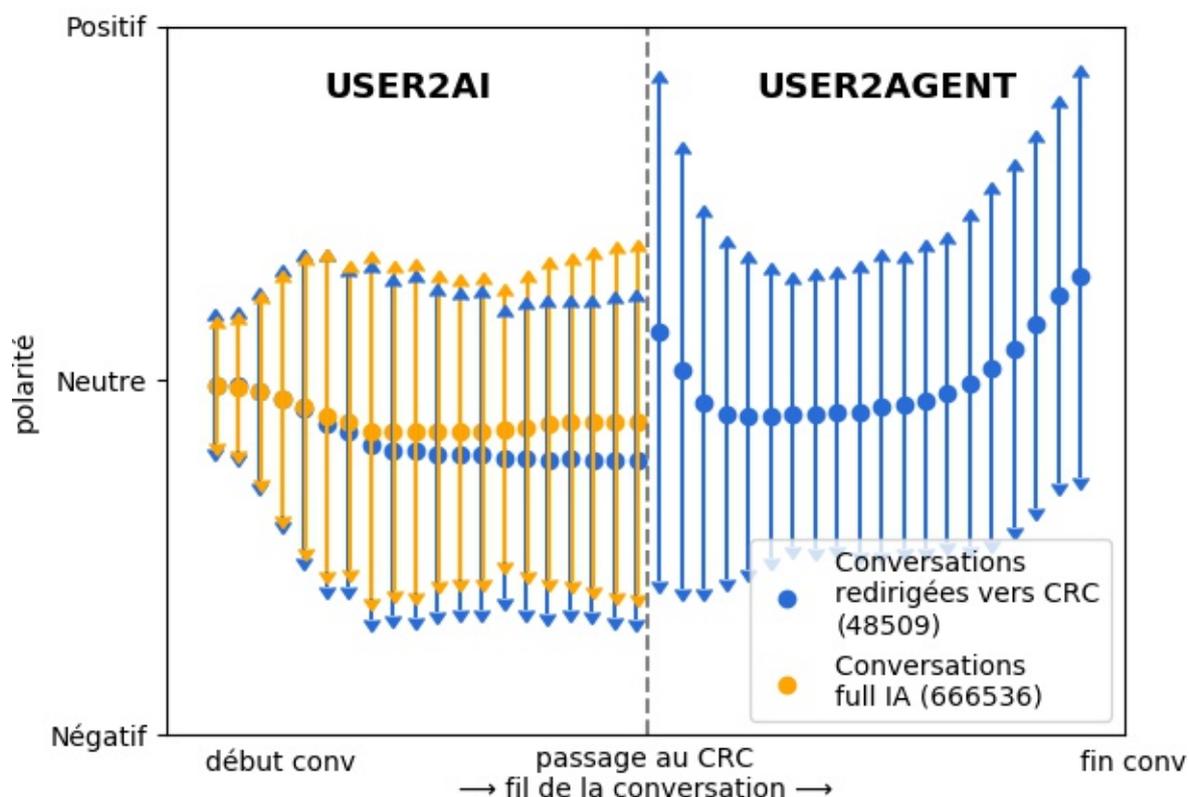
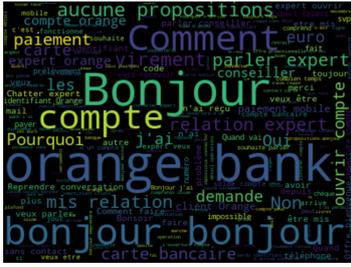


FIGURE 8.2 – Évolution de la polarité des conversations avec passage au CRC. Moyenne sur 12 mois

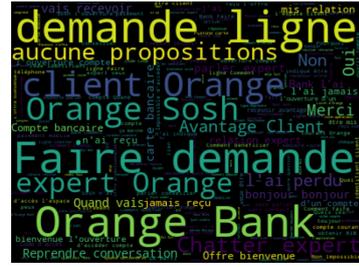
Sur ce graphique, nous remarquons tout d'abord que pour les deux types d'utilisateurs (redirigé et non redirigé (full IA)), la conversation commence avec la même polarité (neutre) en moyenne. Puis, après le premier tiers de conversation, les personnes qui ne seront pas redirigées voient la polarité de leur conversation stagner autour d'une valeur légèrement en dessous de neutre, alors que les personnes qui seront redirigées voient la polarité de leur conversation diminuer jusqu'à prise en charge par un agent. Dès que les personnes sont prises en charge par un conseiller, la polarité de la conversation prend une tournure plus positive (les signes de politesses comme 'bonjour' sont labellisés comme positifs et sont plus présents dans les conversations avec un humain). S'ensuit alors une phase plus neutre, qui correspond généralement à la prise d'information par le conseiller. Puis la dernière phase correspond à une période de remerciements. Nous avons illustré ces propos par des nuages de mots.



(a) Nuage de mots du 1er tiers de la conversation - full AI



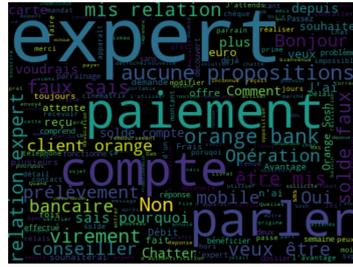
(b) Nuage de mots du 2ème tiers de la conversation - full AI



(c) Nuage de mots du 3ème tiers de la conversation - full AI



(a) Nuage de mots du 1er tiers de la conversation - USER2AI (avant redirection vers le CRC)



(b) Nuage de mots du 2ème tiers de la conversation - USER2AI (avant redirection vers le CRC)



(c) Nuage de mots du 3ème tiers de la conversation - USER2AI (avant redirection vers le CRC)



(a) Nuage de mots du 1er tiers de la conversation - USER2AGENT (après redirection vers le CRC)



(b) Nuage de mots du 2ème tiers de la conversation - USER2AGENT (après redirection vers le CRC)



(c) Nuage de mots du 3ème tiers de la conversation - USER2AGENT (après redirection vers le CRC)

La première phase pour les trois situations est en général marquée par des salutations ou des expositions de situations. Nous remarquons la présence de 'bonjour' pour toutes les situations. Pour les USER2AI (utilisateurs qui vont être redirigés), la présence de 'conseiller' et 'expert' dans le premier tiers de discussion traduit la volonté du client de se faire rediriger dès le début de la conversation¹. Cette demande restera présente tout le long de la conversation pour les USER2AI, ce qui signifie qu'une partie des gens qui va être redirigée le demandait dès le début de la conversation.

Les nuages de mots de la deuxième phase de la conversation attestent de la prise d'information par la présence de mots plus techniques ('carte bancaire', 'opération', 'compte',

1. Si la situation n'a pas été exposée (comme au début de la conversation), Djingo demandera jusqu'à deux fois la raison pour laquelle le client veut être redirigé.

...). Dans la partie USER2AGENT, la présence de mots courts ('oui', 'non') témoigne du dialogue engendré avec l'agent.

Enfin, les troisièmes parties des conversations sont différentes. Dans le cas full IA, nous remarquons la présence de 'demande en ligne ' ou 'faire demande' qui invite l'utilisateur à basculer vers un autre service de l'application. Pour le cas des conversations USER2AI qui seront redirigées, la présence de 'expert', 'être mis en relation' atteste que la redirection vers un conseiller va bien avoir lieu. En revanche, pour la dernière partie de la conversation avec agent (USER2AGENT), beaucoup de remerciements apparaissent dans les nuages ('merci', 'bonne journée', ...), ce qui confirme la remontée de la courbe USER2AGENT sur le graphe précédent.

Chapitre 9

Conclusion

Discussion

Les approches vues dans ce mémoire disposent néanmoins de certaines limites. Tout d’abord, la classification repose sur l’annotation, et il n’est pas toujours évident d’annoter suivant les trois classes de polarité. Dans l’exemple : « *Mon épouse est décédé et je souhaite réaliser une demande de succession* », l’utilisateur de l’agent conversationnel fait part d’un évènement passé ainsi que la volonté de réaliser une action. Or la partie « *Mon épouse est décédé* » aurait été annotée comme négatif, alors que la partie « *je souhaite réaliser une demande de succession* » aurait été annotée en neutre. Une nouvelle classe ‘mix positif-négatif’ aurait pu être utilisée comme dans DEFT 2018, mais aurait demandé un travail différent d’analyse.

D’autre part, la polarité est une information utile, mais ne nous renseigne pas sur la subjectivité du message. Il y a une différence sensible entre un utilisateur se plaignant d’un service particulier d’Orange Bank (exemple « *Ma carte bancaire ne marche pas* », polarité négative) et un utilisateur mécontent sans qu’une raison précise ne soit énoncée (exemple « *Orange c’est vraiment de plus en plus pourri!* », polarité négative). Nous reviendrons sur ce point dans la partie Perspectives.

Enfin, le passage de la polarité à la maille du message à la maille de la conversation, s’est effectué avec une approche par règles, en créant en plus une catégorie mitigée. Or cette catégorie ne tient pas compte de l’intensité que peuvent avoir certains messages. Dans l’exemple de la table 9.1, la conversation dispose d’une polarité mitigée (présence de positif et de négatif), mais reste très négative par la présence du dernier message. Une annotation à la maille de la conversation aurait sans doute permis de classer cette conversation comme négative, mais n’aurait pas fait de différence entre cette conversation très négative et une autre moins négative.

Pour finir, la visualisation en carte thermique nous confère un aperçu de l’évolution de la polarité, mais ne détaille pas les raisons de la variation. De plus, nous n’avons pas trouvé de corrélation pour tous les thèmes entre leurs scores de polarité au mois et leurs taux de redirection. Il est néanmoins possible que cette métrique ne soit pas adaptée pour comparer ces données.

Message	Polarité
bonjour,	positif
association loi 1901 peut elle ouvrir un compte chez vous ?	neutre
compte + association oi 1901	neutre
je ne parle pas aux robots, connards	négatif

TABLE 9.1 – Exemple de conversation classée comme mitigée

Résumé

Dans ce mémoire, nous avons présenté plusieurs applications de l’analyse d’opinion sur les conversations d’agents conversationnels. En élaborant un modèle d’analyse de la polarité (positif, négatif, neutre) via des algorithmes d’apprentissage automatique, nous avons pu exploiter les données pour faire ressortir des tendances. Cette analyse nous permet, via des graphiques, de regarder quelles thématiques de l’agent conversationnel disposent le plus de satisfaction ou insatisfaction client, à un instant précis, comme sur une échelle de temps. Elle donne aussi la possibilité de faire ressortir du corpus des conversations très orientées (très positive ou négative) à des fins pédagogiques pour les agents du centre de relation client. Enfin, cet outil permet d’obtenir une quantification de l’opinion à chaud des clients.

Perspectives

Pour aller plus loin, plusieurs approches sont possibles. Tout d’abord pour répondre à une des problématiques de la partie précédente, une annotation en subjectif / objectif sur les messages peut être utilisée afin de détecter si les messages ayant une polarité non neutre sont le reflet d’un rejet ou d’un dysfonctionnement des services d’Orange Bank.

Ensuite, il serait possible d’incorporer plus de graduation dans la polarité. Soit via une approche par règles (par exemple présence d’insultes) soit en entraînant un autre modèle suivant cinq classes (très positif, positif, neutre, négatif, très négatif).

Enfin, en plus du travail déjà effectué sur l’analyse de la polarité, un travail sur l’analyse des émotions pourrait être utile, afin de mieux connaître le ressenti émotionnel des utilisateurs de Djingo, et permettre une rédaction du message s’adaptant aux émotions de l’utilisateur.

Bibliographie

- [1] Bing Liu. Sentiment analysis and opinion mining, 2012.
- [2] Ferrari Stéphane Zhang Lei. Intensité et polarité : un modèle opératoire articulant plusieurs travaux linguistiques. In *Langue française, /4 (num 184)*, p. . DOI : 10.3917/lf.184.0035., pages 35–54, 2014.
- [3] Guide d’annotations d’opinion/sentiment/émotions dans les microblogs.
- [4] Guide d’annotation pour le défi fouille de textes (deft) 2018.
- [5] W Parrott. Emotions in social psychology : Essential readings, 2001.
- [6] Clayton J. Hutto and Eric Gilbert. Vader : A parsimonious rule-based model for sentiment analysis of social media text., 2014.
- [7] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet : A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC’06)*, pages 417–422, 2006.
- [8] Joseph Lark, Emmanuel Morin, and Sebastian Peña Saldarriaga. CanÉphore : un corpus français pour la fouille d’opinion ciblée. In *Actes de la 22e conférence sur le Traitement Automatique des Langues Naturelles*, pages 418–424, Caen, France, June 2015. Association pour le Traitement Automatique des Langues.
- [9] Thorsten Joachims. Text categorization with support vector machines : Learning with many relevant features, 1998.
- [10] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up : Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, EMNLP ’02*, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.
- [11] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751, 2014.
- [12] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5) :513–523, August 1988.
- [13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *DBLP :journals/corr/abs-1301-3781*, 2013.
- [14] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents, 2014.

- [15] Josh Attenberg, Kilian Weinberger, Alex Smola, Anirban Dasgupta, and Martin Zinkevich. Collaborative spam filtering with the hashing trick, 2009.
- [16] Joseph Lilleberg, Yun Zhu, and Yanqing Zhang. Support vector machines and word2vec for text classification with semantic features, 07 2015.
- [17] Anton Firsov Gael Guibon Liana Ermakova, Hosni Seffih and Guillaume Le Noé-Bienvenu. Multilingual fake news detection with satire on vaccination topic, 2019.
- [18] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv :1607.04606*, 2016.
- [19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, 12 :2825–2830, 2011.
- [20] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you ?" : Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.
- [21] Vladimir Vapnik et A. Lerner. Pattern recognition using generalized portrait method, automation and remote control, 1963.
- [22] Vladimir Vapnik. The nature of statistical learning theory, 1995.

Annexe

Fonctionnement de l'algorithme SVM

Les SVM (Support Vector Machine ou machine à vecteurs de support) sont des algorithmes de classification par apprentissage supervisé. Historiquement, les bases de l'algorithme ont été déposées dans les années 1960 par Vladimir Vapnik et A. Lerner [21], mais les premiers algorithmes n'ont été implémentés que dans le milieu des années 1995 [22].

En entrée l'algorithme prend des vecteurs correspondant aux données annotées. Dans le cadre de ce mémoire, les vecteurs correspondent aux documents, comme vu dans la section 6.1. L'idée de base de l'algorithme consiste à trouver un hyperplan séparant les classes des données. L'image ci-dessous illustre ce cas précis.

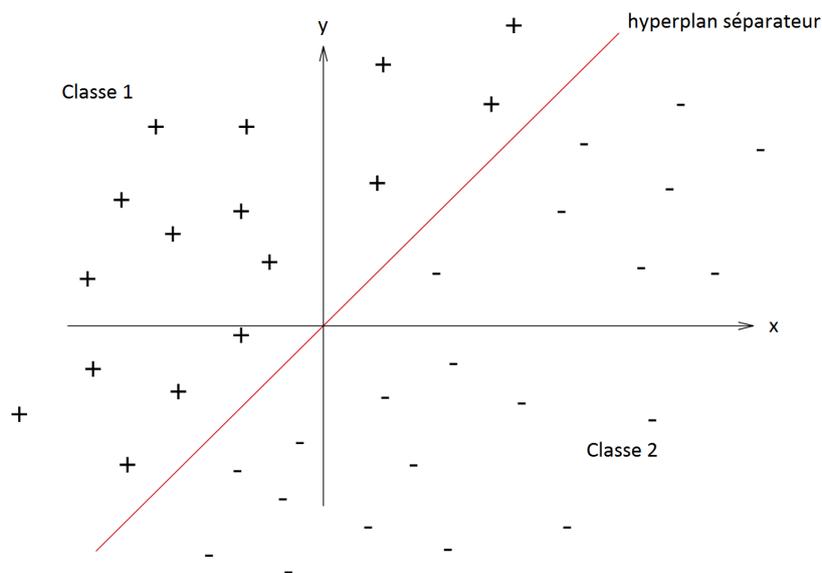
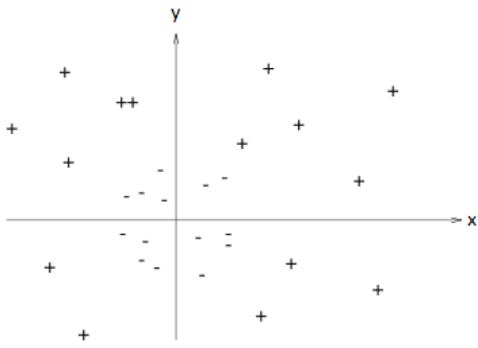
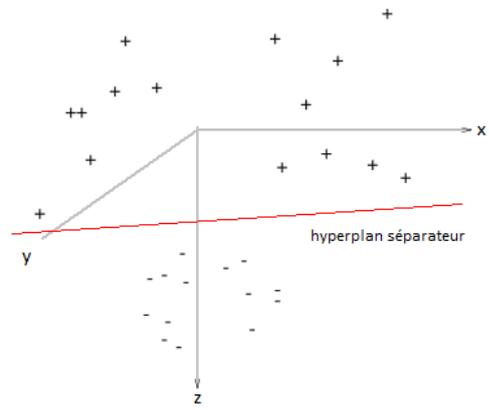


FIGURE 9.1 – Illustration de l'algorithme dans le cas où les données peuvent être séparables linéairement

Dans le cas où les données ne sont pas linéairement séparables, l'algorithme utilise le *kernel trick* (astuce du noyau), qui consiste à représenter les données dans un espace de dimension plus grand. Un hyperplan est ensuite trouvé permettant de discriminer les classes.



(a) Situation où les données ne sont pas séparables linéairement



(b) En basculant dans l'espace, l'algorithme trouve un hyperplan séparant les deux classes des données

Résultats classification Doc2Vec

Représentation utilisée	Sans doublons		Avec doublons	
	f-mesure macro	f-mesure micro	f-mesure macro	f-mesure micro
Doc2Vec seul vs=50, w=7, mincount=2, epochs=10	0.25	0.62	0.28	0.74
Doc2Vec seul vs=50, w=5, mincount=2, epochs=10	0.25	0.62	0.28	0.74
Doc2Vec seul vs=100, w=7, mincount=2, epochs=50	0.25	0.62	0.28	0.74
Doc2Vec seul vs=300, w=7, mincount=2, epochs=50	0.25	0.62	0.28	0.74
Doc2Vec seul vs=300, w=7, mincount=2, epochs=10	0.25	0.62	0.28	0.74
Doc2Vec seul vs=300, w=5, mincount=2, epochs=50	0.25	0.62	0.28	0.74
Doc2Vec seul vs=300, w=5, mincount=2, epochs=10	0.25	0.62	0.28	0.74
Doc2Vec seul vs=100, w=7, mincount=2, epochs=10	0.25	0.62	0.28	0.74
Doc2Vec seul vs=100, w=5, mincount=2, epochs=50	0.25	0.62	0.28	0.74
Doc2Vec seul vs=100, w=5, mincount=2, epochs=10	0.25	0.62	0.28	0.74
Doc2Vec seul vs=300, w=5, mincount=2, epochs=10	0.25	0.62	0.28	0.74
Doc2Vec seul vs=50, w=7, mincount=2, epochs=50	0.26	0.62	0.28	0.74
Doc2Vec seul vs=300, w=5, mincount=2, epochs=200	0.26	0.62	0.29	0.74
Doc2Vec seul vs=50, w=5, mincount=2, epochs=50	0.28	0.63	0.29	0.74
Doc2Vec seul vs=50, w=7, mincount=2, epochs=50	0.28	0.74	0.29	0.74
Doc2Vec seul vs=50, w=7, mincount=2, epochs=10	0.28	0.74	0.29	0.74
Doc2Vec seul vs=50, w=5, mincount=2, epochs=10	0.28	0.74	0.29	0.74
Doc2Vec seul vs=100, w=7, mincount=2, epochs=50	0.28	0.74	0.29	0.74
Doc2Vec seul vs=300, w=7, mincount=2, epochs=50	0.28	0.74	0.29	0.74
Doc2Vec seul vs=300, w=7, mincount=2, epochs=10	0.28	0.74	0.29	0.74
Doc2Vec seul vs=300, w=5, mincount=2, epochs=50	0.28	0.74	0.29	0.74
Doc2Vec seul vs=300, w=5, mincount=2, epochs=10	0.28	0.74	0.29	0.74
Doc2Vec seul vs=100, w=7, mincount=2, epochs=10	0.28	0.74	0.29	0.74
Doc2Vec seul vs=100, w=5, mincount=2, epochs=50	0.28	0.74	0.29	0.74

Doc2Vec seul vs=100, w=5, mincount=2, epochs=10	0.28	0.74	0.29	0.74
Doc2Vec seul vs=300, w=5, mincount=2, epochs=10	0.28	0.74	0.29	0.74
Doc2Vec seul vs=300, w=5, mincount=2, epochs=200	0.29	0.74	0.29	0.74
Doc2Vec seul vs=50, w=5, mincount=2, epochs=50	0.29	0.74	0.29	0.74
Doc2Vec seul vs=300, w=5, mincount=2, epochs=200	0.32	0.65	0.33	0.75
Doc2Vec seul vs=300, w=5, mincount=2, epochs=200	0.33	0.75	0.33	0.75
Doc2Vec seul vs=300, w=7, mincount=2, epochs=200	0.33	0.66	0.35	0.76
Doc2Vec seul vs=300, w=7, mincount=2, epochs=200	0.35	0.76	0.35	0.76
Doc2Vec seul vs=100, w=7, mincount=2, epochs=200	0.45	0.74	0.46	0.8
Doc2Vec seul vs=100, w=7, mincount=2, epochs=200	0.46	0.8	0.46	0.8
Doc2Vec seul vs=100, w=5, mincount=2, epochs=200	0.46	0.75	0.47	0.81
Doc2Vec seul vs=100, w=5, mincount=2, epochs=200	0.47	0.81	0.47	0.81
Doc2Vec seul vs=50, w=5, mincount=2, epochs=200	0.54	0.81	0.56	0.86
Doc2Vec seul vs=50, w=5, mincount=2, epochs=200	0.56	0.86	0.56	0.86
Doc2Vec seul vs=50, w=7, mincount=2, epochs=200	0.56	0.83	0.56	0.86
Doc2Vec seul vs=50, w=7, mincount=2, epochs=200	0.56	0.86	0.56	0.86

TABLE 9.2 – Résultats obtenus pour la classification via Doc2Vec

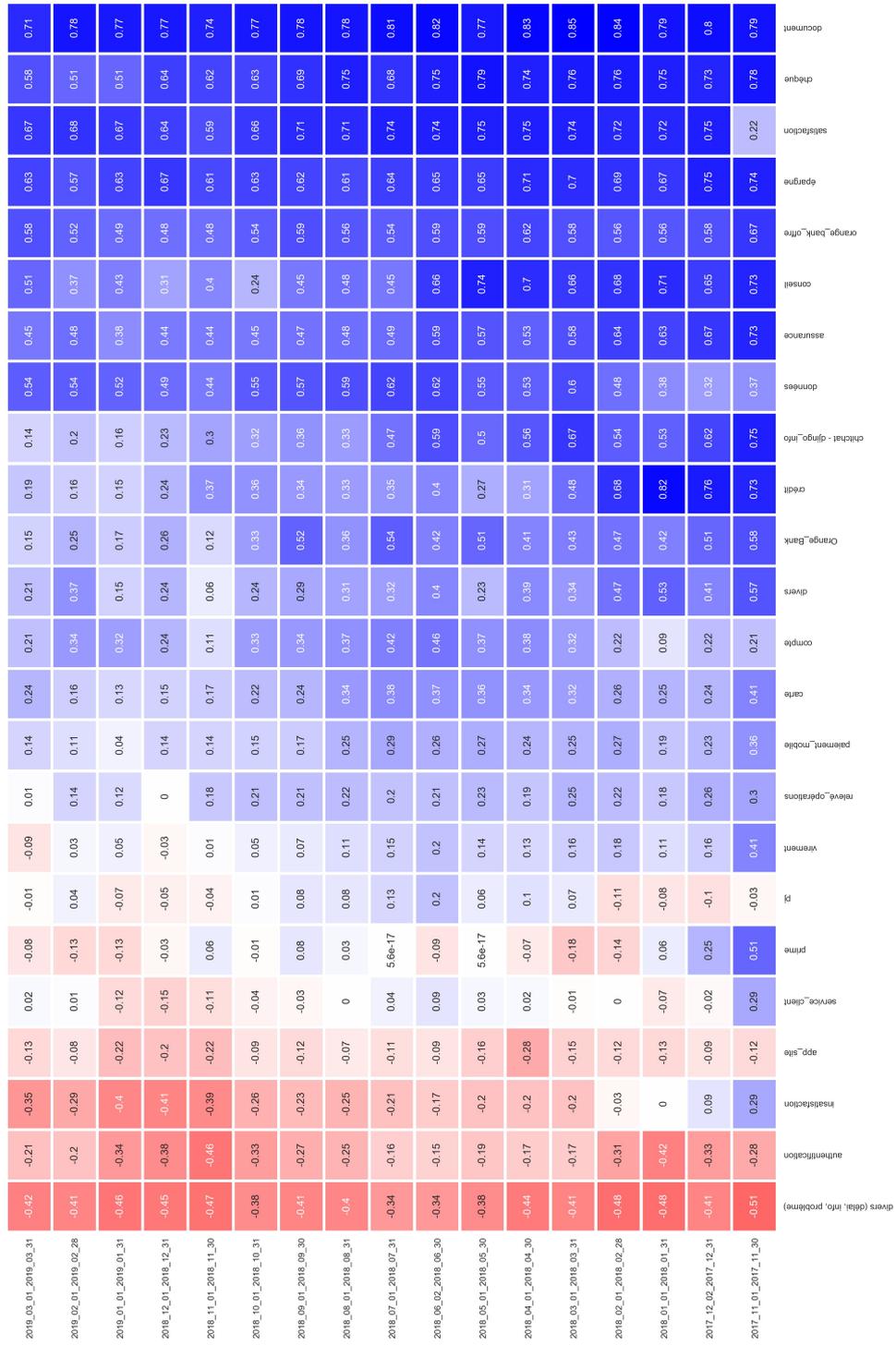


FIGURE 9.4 – Évolution de l’opinion en fonction des thèmes à l’échelle de l’année - Thèmes simplifiés

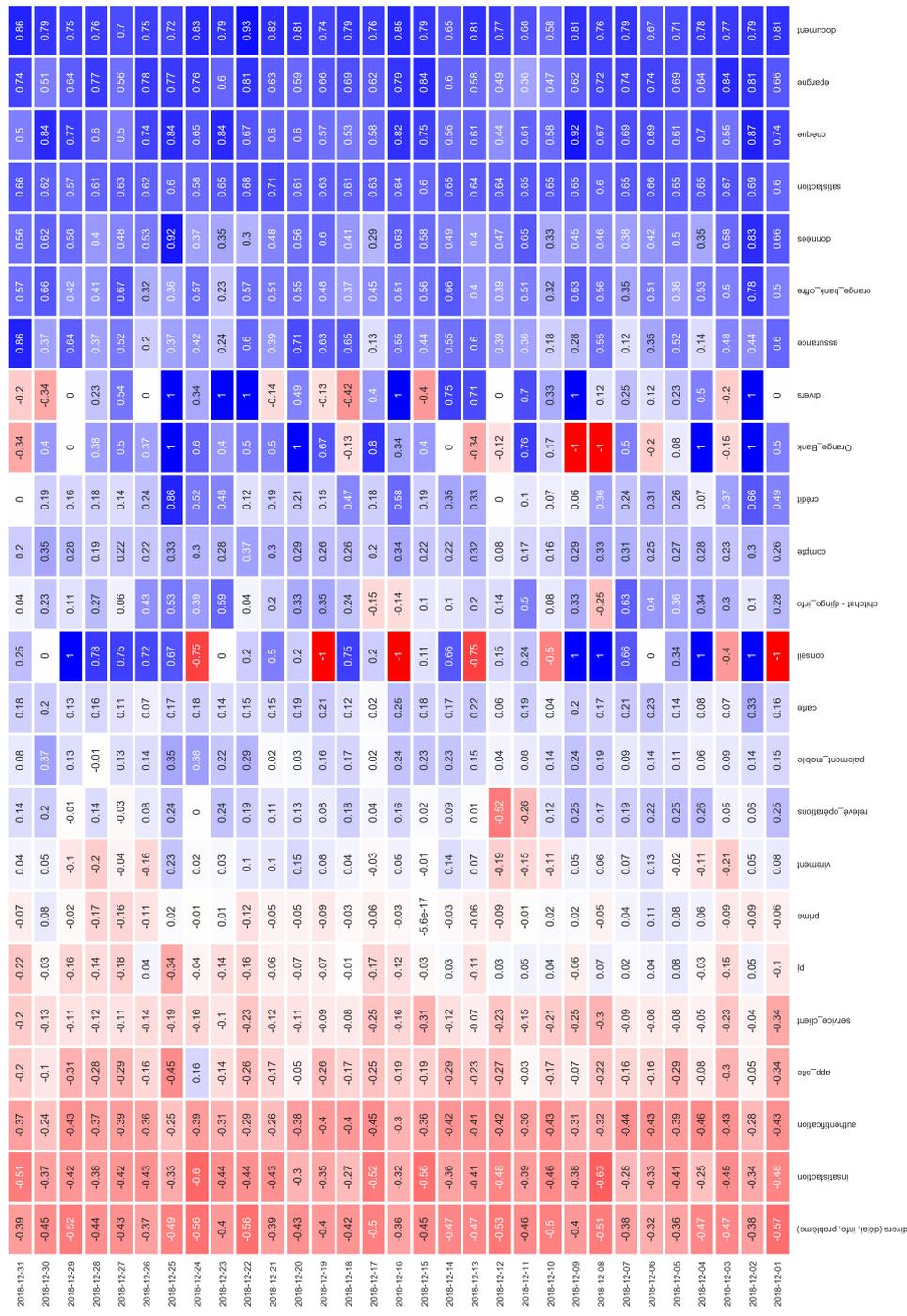


FIGURE 9.6 – Évolution de l’opinion en fonction des thèmes à l’échelle du mois - Thèmes simplifiés

Résultats du calcul du coefficient de Pearson et p-value entre le taux de redirection et le score de polarité en fonction des thèmes

Thème	p-value	r	Nombre de conversations ayant ce thème entre 2017-11 et 2018-12	Nombre moyen de thèmes appartenants aux conversations portant ce thème
Orange_Bank	0.211426	0.319398	2432	2.94
app_site_info	0.031500	-0.522280	10133	2.4
app_site_probleme	0.433381	0.203507	23153	2.01
assurance_info	0.995269	-0.001557	9859	2.29
assurance_resiliation	NaN	NaN	NaN	NaN
authentification_info	0.000024	-0.840584	32793	2.25
authentification_probleme	0.896543	-0.034127	192479	1.5
carte_delai	0.972877	-0.008926	32460	2.17
carte_info	0.769955	-0.076660	74790	2.12
carte_probleme	0.918484	-0.026864	44064	1.77
cheque_info	0.000010	-0.858538	43528	2.03
cheque_probleme	0.000004	-0.876924	14820	2.1
chitchat	0.096119	0.416710	4725	2.79
compte_info	0.002094	-0.691778	94251	2.04
compte_probleme	0.827698	-0.057095	1242	2.08
compte_resiliation	0.003437	-0.667161	52153	1.68
compte_souscription_delai	0.137514	-0.375460	21010	2.26
compte_souscription_info	0.452676	-0.195243	40126	2.43
compte_souscription_probleme	0.022541	-0.548751	90033	2.0
conseil_financier	0.040219	0.501613	2461	2.86
conversation	0.440950	-0.200243	206408	2.1
credit_info	0.190960	-0.333402	8497	2.66
credit_resiliation	0.091180	-0.487344	2331	2.24
credit_souscription_delai	0.596420	-0.155132	3570	2.55
credit_souscription_info	0.005581	-0.640755	7153	2.59
credit_souscription_probleme	0.471348	0.219426	4904	1.84
divers_delai	0.000509	-0.751235	26989	2.53
divers_info	0.345297	-0.243993	10606	3.08
divers_probleme	0.056308	-0.471069	48107	2.53
djingo_info	0.003726	-0.662933	3707	3.13
document_info	0.192054	-0.332630	35150	2.03
donnees_info	0.094481	-0.418583	24239	1.88
epargne_info	0.048872	-0.484237	18775	1.96
epargne_souscription	0.852385	0.048822	3584	2.71
groupama	0.955682	-0.014589	1008	2.32

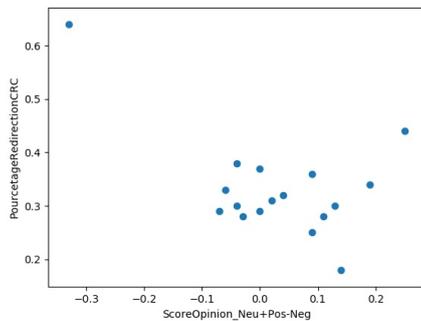
hors_offre_info	0.121146	0.390561	13079	2.25
insatisfaction_incomprehension	0.000345	-0.765135	28073	2.62
jeu_ob	0.000000	1.000000	144	2.45
orange_bank_offre_info	0.030461	0.525025	15870	2.69
paiement_mobile_activation_info	0.812870	-0.062088	12073	2.1
paiement_mobile_activation_probleme	0.398824	-0.218803	2200	2.19
paiement_mobile_info	0.004090	-0.657962	36033	2.0
paiement_mobile_probleme	0.000003	-0.879467	8568	2.11
pj_info	0.006652	-0.630569	19845	2.47
pj_probleme	0.685346	0.106072	36571	1.96
prime_delai	0.317638	-0.257882	81389	2.05
prime_info	0.002883	0.676159	55857	2.26
produits_orange	0.098726	0.413775	2499	2.5
releve_operations_info	0.130425	-0.381837	32350	1.96
releve_operations_probleme	0.801457	0.065945	54195	1.72
satisfaction	0.045791	0.490134	65406	2.6
service_client_contact	0.000029	-0.836467	179969	2.0
service_client_info	0.792869	0.068856	11169	2.31
virement_info	0.309702	-0.261997	31813	2.14
virement_probleme	0.000900	-0.729017	40742	1.71

TABLE 9.3 – Résultats calcul coefficient de Pearson et p-value entre le taux de redirection et le score de polarité en fonction des thèmes

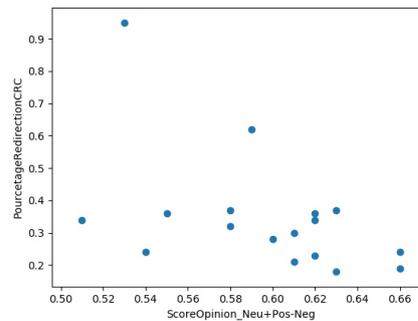
Moyenne du nombre de conversation par thèmes : 34900. Moyenne du nombre de thème par conversation : 2.26.

Mois	Pourcentage Redirection CRC	ScoreOpinion Neu+Pos-Neg
2017-11	0.606938071	0.737874016363338
2017-12	0.550935523	0.7258046306949886
2018-01	0.581320157	0.7035079122484558
2018-02	0.473150006	0.7328714889858339
2018-03	0.45049388	0.7537145351374945
2018-04	0.445752435	0.7538626715371706
2018-05	0.432798012	0.7641914453808236
2018-06	0.413758113	0.777672440835105
2018-07	0.453144879	0.7630892110310996
2018-08	0.467551259	0.7516584835650837
2018-09	0.475984224	0.744895443550257
2018-10	0.482734505	0.7407017021649647
2018-11	0.507350638	0.7047005350341748
2018-12	0.480224393	0.7032604925424903
2019-01	0.487340893	0.7165417337443665
2019-02	0.489110629	0.7476292807488318
2019-03	0.514278068	0.7362889118844245
p-value	0.0078575	
r	-0.620578	

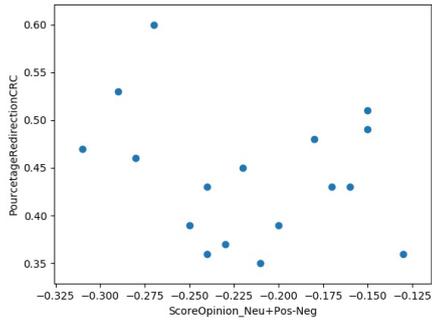
TABLE 9.4 – Résultats calcul coefficient de Pearson et p-value entre le taux de redirection et le score de polarité



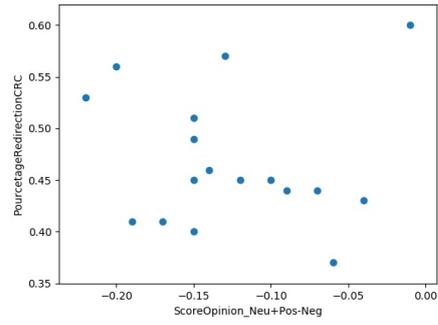
(a) Diagramme de dispersion entre le taux de re-direction vers le CRC et le score d'opinion pour le thème 'app_site_info'
p-value ≈ 0.03 , $r \approx -0.52$



(b) Diagramme de dispersion entre le taux de re-direction vers le CRC et le score d'opinion pour le thème 'epargne_info'
p-value ≈ 0.05 , $r \approx -0.48$



(a) Diagramme de dispersion entre le taux de re-direction vers le CRC et le score d'opinion pour le thème 'divers_info'
 p-value ≈ 0.35 , $r \approx -0.24$



(b) Diagramme de dispersion entre le taux de re-direction vers le CRC et le score d'opinion pour le thème 'paiement_mobile_activation_info'
 p-value ≈ 0.81 , $r \approx -0.06$

Exemples de conversations

Conversation très positive
<p>Q : Bonjour,</p> <p>R : Bonjour Marie Flore, je vous écoute. Que puis-je faire pour vous ?</p> <p>Q : J'aimerais faire un virement à un societe</p> <p>R : Souhaitez-vous effectuer : <clictexte> un virement par SMS </clictexte> <clictexte> un virement standard </clictexte></p> <p>Q : un virement standard</p> <p>R : Pour effectuer un virement, voici la marche à suivre :- si votre bénéficiaire n'est pas encore enregistré :</p> <p>depuis l'écran principal de la rubrique virement , sélectionnez l'icône carnet d'adresse en haut à droite de votre application mobile puis enregistrez votre bénéficiaire - une fois votre bénéficiaire enregistré :</p> <p>depuis l'écran d'accueil, glissez le curseur de paiement vers la droite, direction envoyer</p> <p>Q : Ok merci</p> <p>R : C'est un plaisir</p> <p>Q : Bonne journée</p> <p>R : À bientôt !</p>
Conversation positive
<p>R : Bonjour Pascale, ravi de vous revoir. En quoi puis-je vous aider ?</p> <p>Q : Bonjour. Je possède déjà un livret A dans une autre banque. Est ce que je peux en ouvrir un autre chez Orange bank ?</p> <p>R : De quel compte souhaitez-vous connaître les conditions ? <clictexte> Compte bancaire </clictexte><clictexte> Le Livret </clictexte></p> <p>Q : Compte bancaire</p> <p>R : Pour ouvrir un compte Orange Bank, vous devez être majeur (e) résidant fiscalement en France. Bon à savoir : il n'y a pas de condition de revenus !</p> <p>Q : Et le livret</p> <p>R : Je ne suis pas sûr de bien comprendre votre demande. Vous voulez dire :</p> <p><clictexte> Comment fonctionne Le Livret d'Orange Bank ? </clictexte> Faut-il payer des impôts sur les revenus du Livret d'Orange Bank ? </clictexte> <clictexte> – aucune de ces propositions – </clictexte></p> <p>Q : Comment fonctionne Le Livret d'Orange Bank ?</p> <p>R : Le Livret d'Orange Bank est idéal pour votre épargne : votre argent reste à disposition et il n'y a aucune limite de montant. Le solde minimum est de 10 euros, mais vous n'avez aucune obligation de versement complémentaire</p> <p>Q : Ok merci</p> <p>R : C'est toujours un plaisir de vous rendre service</p>
Conversation neutre

R : Bonjour Pascal, ravi de vous revoir. En quoi puis-je vous aider ?

Q : Comment faire avoir un chéquier

R : Pour commander un chéquier, vous pouvez faire une demande en ligne dans la rubrique " profil " puis " mes demandes ".

Sélectionnez " mes moyens de paiement " et enfin " commander un chéquier " Vous n'avez pas besoin de fournir de pièce justificative

Q : Est il payant de commander un chequier

R : J'hésite. Voulez-vous dire : <clictexte> Quels sont les différents moyens de paiement Orange Bank ? </clictexte><clictexte> Combien coûte la commande d'un moyen de paiement ? </clictexte>

<clictexte> – aucune de ces propositions – </clictexte>

Q : Combien coûte la commande d'un moyen de paiement ?

R : Votre chéquier est totalement gratuit. Cependant, si vous demandez un envoi en recommandé, des frais d'envoi vous seront facturés

Conversation négative

Q : je veux modifier mes plafonds

R : Quel plafond souhaitez-vous modifier ?

<clictexte> Plafond de carte bancaire </clictexte>

<clictexte> Plafond de paiement sans contact par carte bancaire </clictexte>

<clictexte> Plafond de paiement mobile </clictexte>

<clictexte> Plafond de virement SMS </clictexte>

<clictexte> Plafond de virement </clictexte>

<clictexte> Plafond de découvert autorisé </clictexte>

Q : Plafond de carte bancaire

R : Pour modifier votre plafond de carte bancaire

(sous réserve d'acceptation par Orange Bank) , rendez-vous dans la rubrique

" paiement " de votre application mobile, sélectionnez votre carte,

puis choisissez " modifier mes plafonds " Vous avez la possibilité de

faire une modification temporaire ou permanente. Par défaut votre modification

de plafond sera temporaire, si vous voulez qu'elle soit permanente, il vous

suffira de décocher la case correspondante. Un justificatif de revenus

vous sera demandé <clictexte> Quels sont les justificatifs de revenus acceptés </clictexte>

Q : Quels sont les justificatifs de revenus acceptés

R : Les documents suivants sont acceptés comme justificatifs de revenus

- dernier bulletin de salaire - dernier avis d'imposition

Q : le document ne charge pas

R : Si vous rencontrez un problème avec votre pièce justificative,

vous pouvez : - télécharger / photographier un autre document

- Si vous n'en avez pas d'autres, réessayer ou cocher la case "

je n'ai pas d'autre justificatif ". Nos équipes parviendront à accéder

à votre pièce Si ça ne fonctionne pas, je passe le relais <crchat>

Conversation très négative

Q : je n'ai pas reçu un virement en ma faveur du 5 mars
R : Pouvez-vous me préciser s'il s'agit du premier versement pour l'ouverture de votre Livret d'Orange Bank ?
<clictexte> oui </clictexte><clictexte> non </clictexte>
Q : non
R : Je comprends que vous rencontrez un problème avec un virement que vous attendez. Je vous invite à prendre contact avec un expert Orange Bank <crchat>
Q : j'attends depuis 9 minutes alors que le message disait 3 minutes
R : J'hésite. Voulez-vous dire : <clictexte> Le délai est dépassé? </clictexte>
<clictexte> Je n'ai pas reçu le message envoyé par Orange Bank? </clictexte>
<clictexte> – aucune de ces propositions – </clictexte>
Q : Le délai est dépassé?
R : Je suis désolé pour ce retard <crchat>

Conversation mitigée

Q : Je voudrais avoir un expert Orange
R : Pouvez-vous me poser votre question ? Je devrais pouvoir vous aider.
Si besoin, je passerai le relais à un expert
Q : Je voudrais savoir pourquoi ma carte ne fonctionne pas sur le site cdiscount
R : J'hésite. Voulez-vous dire : <clictexte> Que faire en cas de problème avec un moyen de paiement ou un virement ? </clictexte><clictexte> Pourquoi je n'arrive pas à réaliser un paiement ? </clictexte><clictexte> – aucune de ces propositions – </clictexte>
Q : Pourquoi je n'arrive pas à réaliser un paiement ?
R : Quel est le montant de votre achat ?
Q : 347euro que je paye en 4 fois
R : Vérifiez que vous avez bien saisi les bonnes informations (date d'expiration et cryptogramme de la carte) .
Si vous avez bien saisi ces infos, nous allons voir ensemble d'où peut provenir le problème. Pouvez-vous me dire à quelle étape vous vous trouvez ? <clictexte> Je suis sur la page de redirection 3D Secure et j'ai un problème </clictexte><clictexte> Je suis à la fin du parcours de paiement et j'ai un message qui s'affiche </clictexte>
Q : Je suis à la fin du parcours de paiement et j'ai un message qui s'affiche
R : Certains commerçants sur Internet n'ont pas mis à jour la liste des cartes acceptées. Cette situation devrait changer rapidement !
Q : D'accord merci
R : Tout le plaisir est pour moi
Q : Bonne soirée
R : À bientôt !

TABLE 9.5 – Exemple de conversations obtenue via la méthode par règle

La présence de balises de type SGML (clictexte, crchat, etc.) correspondent à l'affichage de boutons dans la fenêtre de chat. La fenêtre de texte est grisée et l'utilisateur n'a pas d'autre choix que de presser sur un des choix indiqués.

Explications du classifieur via Lime

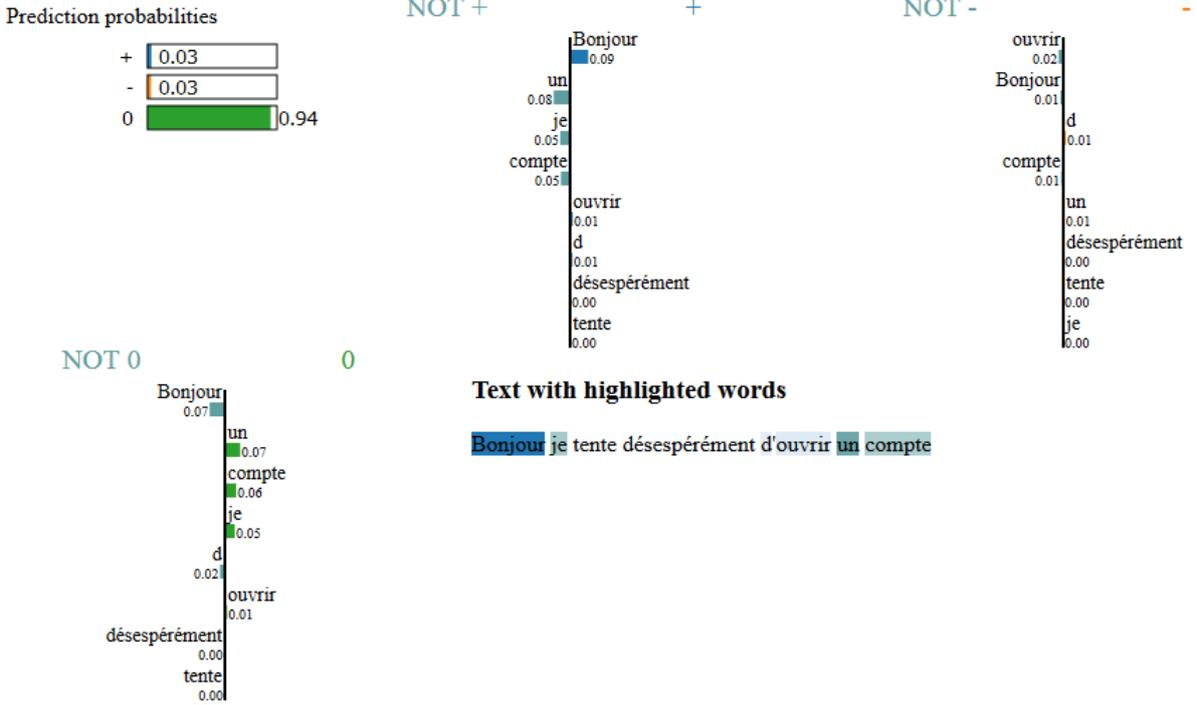


FIGURE 9.10

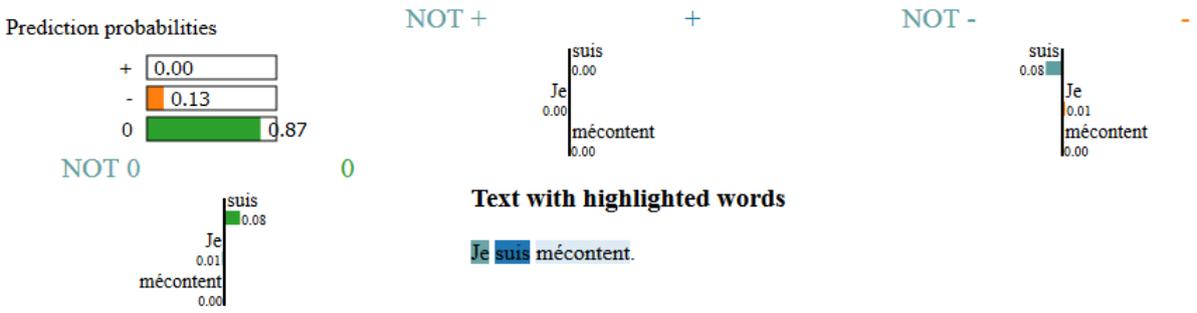


FIGURE 9.11

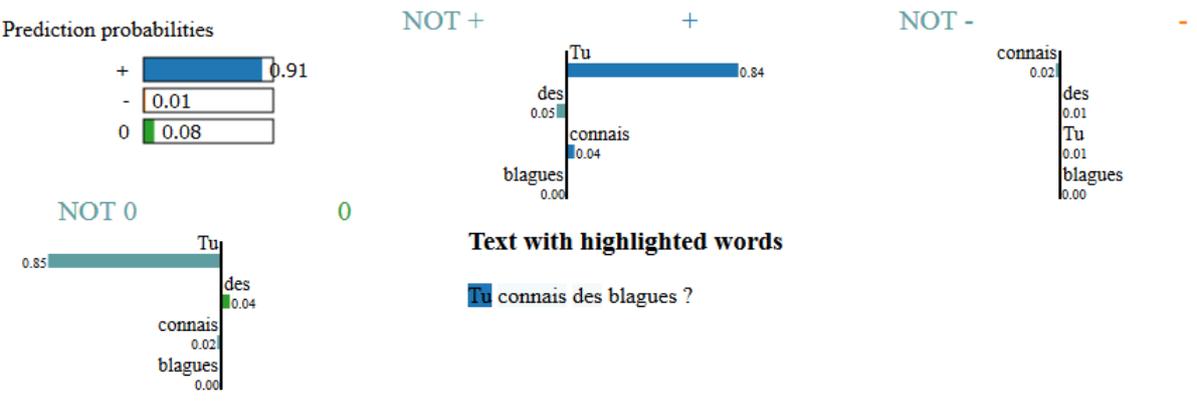
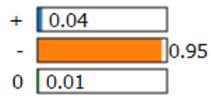


FIGURE 9.12

Prediction probabilities

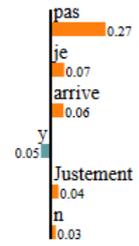


NOT +

+

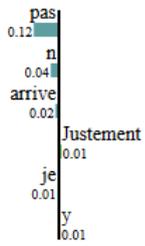
NOT -

-



NOT 0

0



Text with highlighted words

Justement, je n'y arrive pas . . .

FIGURE 9.13

Fonctionnement de l'algorithme d'étalement des listes

Explications

Cet algorithme a été implémenté dans le but de transformer des listes d'entiers de longueur variable vers des listes de réels de dimension fixée.

Pour ce faire, il parcourt la liste donnée en argument, et associe à chaque indice de la liste renvoyée, la moyenne pondérée des termes se trouvant proches.

Le code de cet algorithme est disponible à l'adresse suivante : <https://github.com/GuillaumeLNB/perso/blob/master/rounding.py>.

Exemples de sorties

```
[1,0,2,0],          fenêtre=3 :  
-> [0.75, 1, 0.5]  
[1,0],             fenêtre=3 :  
-> [1.0, 0.5, 0.0]  
  
[1, 0, 2], fenêtre=4 :  
-> [1.0, 0.33, 0.66, 2.0]  
[1, 0, 2], fenêtre=5 :  
-> [1.0, 0.66, 0.0, 1.33, 2.0]  
[1, 0, 2], fenêtre=6 :  
-> [1.0, 1.0, 0.0, 0.0, 2.0, 2.0]  
[1, 0, 2], fenêtre=7 :  
-> [1.0, 1.0, 0.33, 0.0, 0.66, 2.0, 2.0]  
[1, 0, 2], fenêtre=8 :  
-> [1.0, 1.0, 0.66, 0.0, 0.0, 1.33, 2.0, 2.0]  
[1, 0, 2], fenêtre=9 :  
-> [1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 2.0, 2.0, 2.0]  
[1, 0, 2], fenêtre=10 :  
-> [1.0, 1.0, 1.0, 0.33, 0.0, 0.0, 0.66, 2.0, 2.0, 2.0]  
  
[1, 0, 0, 2, 0, 1, 0, 1], fenêtre=3 :  
-> [0.375, 0.875, 0.625]  
[1, 0, 0, 2, 0, 1, 0, 1], fenêtre=4 :  
-> [0.5, 1.0, 0.5, 0.5]  
[1, 0, 0, 2, 0, 1, 0, 1], fenêtre=5 :  
-> [0.625, 0.25, 1.0, 0.625, 0.625]  
[1, 0, 0, 2, 0, 1, 0, 1], fenêtre=6 :  
-> [0.75, 0.0, 1.5, 0.25, 0.5, 0.75]  
[1, 0, 0, 2, 0, 1, 0, 1], fenêtre=7 :  
-> [0.875, 0.0, 0.75, 1.0, 0.625, 0.25, 0.875]  
[1, 0, 0, 2, 0, 1, 0, 1], fenêtre=8 :  
-> [1.0, 0.0, 0.0, 2.0, 0.0, 1.0, 0.0, 1.0]  
[1, 0, 0, 2, 0, 1, 0, 1], fenêtre=9 :  
-> [1.0, 0.125, 0.0, 1.25, 1.0, 0.375, 0.75, 0.125, 1.0]  
[1, 0, 0, 2, 0, 1, 0, 1], fenêtre=10 :  
-> [1.0, 0.25, 0.0, 0.5, 2.0, 0.0, 0.75, 0.5, 0.25, 1.0]
```