

Compte : **Florian.breton1@u-psud.fr**

Identifiant :

Titre : **Memoire_breton_florian_m2.pdf**

Dossier : **Dossier par défaut**

Commentaires : *Non renseigné*

Chargé le : 16/08/2019 10:28

Similitudes document :

 **1%**

INFORMATIONS DÉTAILLÉES

Titre : memoire_breton_florian_m2.pdf

Description :

Analysé le : 16/08/2019 10:58

Identifiant : ntzg3oab

Chargé le : 16/08/2019 10:28

Type de chargement : Remise manuelle des travaux

Nom du fichier : memoire_breton_florian_m2.pdf

Type de fichier : pdf

Nombre de mots : 8656

Nombre de caractères : 63507

Taille originale du fichier (kb) : 5736.95

TOP DES SOURCES PROBABLES- PARMI 5 SOURCES PROBABLES

1.  tel.archives-ouvertes.fr/.../tel-02077011/document  <1%
2.  pdfs.semanticscholar.org/.../d9f8c1f3c3b20b021c...83c2ca255f2a2a.pdf  <1%
3.  tel.archives-ouvertes.fr/.../tel-00818970/document  <1%
4.  kickmybot.com/.../articles/intro_NLP_01_07_18.pdf  <1%
5.  acpr.banque-france.fr/.../20190617_decision_...ur_publication.pdf  <1%

SIMILITUDES TROUVÉES DANS CE DOCUMENT/CETTE PARTIE

Similitudes à l'identique : <1 % 

Similitudes supposées : <1 % 

Similitudes accidentelles : <1 % 

Sources très probables - 5
Sources peu probables - 20

Sources accidentelles- 3 Sources
Sources ignorées - 0 Sources









































SOURCES TRÈS PROBABLES

5 Sources

		Similitude
1.	 tel.archives-ouvertes.fr/.../tel-02077011/document	 <1%
2.	 pdfs.semanticscholar.org/.../d9f8c1f3c3b20b021c...83c2ca255f2a2a.pdf	 <1%
3.	 tel.archives-ouvertes.fr/.../tel-00818970/document	 <1%
4.	 kickmybot.com/.../articles/intro_NLP_01_07_18.pdf	 <1%
5.	 acpr.banque-france.fr/.../20190617_decision...ur_publication.pdf	 <1%

SOURCES PEU PROBABLES

20 Sources

		Similitude
1.	 hal.archives-ouvertes.fr/.../tel-01154811/document	 <
2.	 www.irit.fr/.../actes_TALN-RECITAL..._CH_PFIA2019-2.pdf	 <
3.	 www.aclweb.org/.../anthology/F14-3	 <
4.	 tel.archives-ouvertes.fr/.../file/DDOC_T_2018_0008_ZIMMER.pdf	 <
5.	 www.aepq.ca/.../07/RP_v50n2.pdf	 <
6.	 fr.wikipedia.org/.../wiki/Apprentissage_automatique	 <
7.	 billetdebanque.panorabanes.com/.../a-quoi-sert-l'autor...l-et-de-resolution	 <
8.	 dumas.ccsd.cnrs.fr/.../dumas-01871298/document	 <
9.	 www.eyrolles.com/.../9782212675221/9782212675221.pdf	 <
10.	 tel.archives-ouvertes.fr/.../tel-00303679/document	 <
11.	 hal-univ-tlse2.archives-ouvertes.fr/.../tel-01249652/document	 <
12.	 oraprdnt.uqtr.quebec.ca/.../GSC21/F3099_NLP_IA_96.pdf	 <
13.	 learnigdigital.withgoogle.com/.../lesson/42	 <
14.	 www.telecom-paris.fr/.../Algorithmes-Biais-...ination-equite.pdf	 <
15.	 lepetitjournal.com/.../comprendre-les-age...vi-ben-ezra-257432	 <
16.	 acpr.banque-france.fr/.../medias/documents	 <
17.	 tel.archives-ouvertes.fr/.../tel-01734790/document	 <
18.	 pastel.archives-ouvertes.fr/.../tel-00005759/document	 <
19.	 www.theses.fr/.../2017GREAS047.pdf	 <
20.	 tel.archives-ouvertes.fr/.../tel-00457820/document	 <

SOURCES ACCIDENTELLES

3 Sources

		Similitude
1.	 tel.archives-ouvertes.fr/.../tel-00118602v2/document	 <1%

2.  www.math.univ-toulouse.fr/.../pub/Appren_stat.pdf
3.  www.lemagit.fr/.../definition/Traitement-du-langage-naturel-TLN



<1%

<1%

SOURCES IGNORÉES

*0 Source**Similitude*

TEXTE EXTRAIT DU DOCUMENT

VISA ENTREPRISE

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

1

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

2

Remerciements

Grâce au concours de quelques personnes, j'ai eu l'opportunité d'étudier des sujets qui ont suscité un réel intérêt chez moi, en l'occurrence le traitement automatique du langage naturel. Je souhaite remercier l'équipe d'enseignants chercheurs du Master en informatique pour la science des données de l'université Paris-Saclay, ainsi que tous les intervenants extérieurs.

Je remercie le professeur Khaldoun Al Agha d'avoir été mon référent lors de ces deux années d'alternance.

Des projets d'intelligence artificielle et de visualisation de données très riches d'enseignements m'ont été conés. Pour cela, pour leur conance et leur soutien je tiens à remercier Jean-Marc Mathias et Otilia Popa, qui ont porté une attention particulière au bon déroulement de mon alternance au sein d'Orange.

Je remercie également toute l'équipe GRIN, dans laquelle j'ai pu m'exercer, évoluer et grandir grâce à la bienveillance qui y règne.

J'ai à coeur également de remercier mes camarades de promotion avec qui j'ai pu apprendre et partager tout au long des ces deux années d'études.

Enn, je remercie inniment mes parents qui ont réuni toutes les conditions an que je réussisse ce projet.

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

3

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

SOMMAIRE

Avant-Propos.....	6
Introduction.....	7
1. Compréhension du langage naturel	8
1.1. Représentations d'un mot	8
1.2. Extraction d'information	11
1.2.1. Reconnaissance d'intentions	12
1.2.2. Reconnaissance d'entités.....	14
1.3. Services de compréhension du langage	15
2. Optimisation d'un modèle de compréhension du langage	16
2.1. Techniques liées aux corpus d'apprentissage	16
2.1.1. Volume.....	16
2.1.2. Véracité	20
2.2. Techniques liées aux algorithmes d'apprentissage	21
3. Proposition d'approche de construction d'un corpus métier.....	25
3.1. Motivation	25
3.2. GCM	26
Conclusion.....	28
BIBLIOGRAPHIE/WEBOGRAPHIE	29
GLOSSAIRE.....	30
ANNEXES	31
OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL	

Avant-Propos

Ce mémoire est réalisé dans le cadre du Master en Informatique pour la Science des Données de l'Université Paris Saclay en collaboration avec Orange et le CFA AFIA.

Les encadrants de cet exercice sont Jean-Marc Mathias et Khaldoun Al Agha.

Toutes les ressources nécessaire au expériences réalisées et présentées dans ce mémoire sont disponibles en ligne et constituent les annexes de ce document.

Je garantis que tous les éléments rassemblés dans ce document sont originaux.

Ce mémoire se veut didactique et il a pour objectif de présenter un panel d'outils et de méthodes participant à l'élaboration et à l'optimisation de systèmes de compréhension du langage naturel, néanmoins il n'a pas pour but de détailler chacun de ses items.

Ce document peut servir à des responsables de SI, ou à des développeurs n'étants pas acculturés aux technologies du traitement du langage souhaitant acquérir une connaissance élargie dans ce domaine, par l'exemple et la pratique.

Pour que le lecteur dispose de l'intégralité des annexes, elles font l'objet d'un

hébergement en ligne.

Lien vers les annexes: https://github.com/obre/memoire_2019

Florian Breton

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

6

Introduction

« L'intelligence artificielle serait la version ultime de Google. Le moteur de recherche optimal qui comprendrait tout sur le web. Il comprendrait exactement ce que vous voulez, et vous fournirait le bon contenu. Nous sommes loin d'être en passe d'atteindre ce niveau de service. Malgré tout, nous pouvons nous en rapprocher par petits incréments, et c'est ce sur quoi nous travaillons. » Larry Page, le co-fondateur de Google à l'origine de cette citation, exprime ce qui est l'objet de ce mémoire. En effet il s'agit ici de présenter les méthodes et outils permettant d'optimiser la compréhension du langage naturel par les systèmes informatiques.

Dès les origines de l'informatique, la communication en langage naturel entre Hommes et machines a été envisagée. Pour ce faire il faut que la machine comprenne le langage humain et qu'elle soit en mesure de répondre dans le même langage. Dans le contenu de ce mémoire, il ne sera pas question de la partie « réponses » des systèmes vers les humains, mais uniquement de la partie compréhension du langage naturel. Fort d'une expérience en traitement automatique du langage naturel (TALN) acquise au sein d'Orange lors de mon alternance en Master d'Informatique pour la Science des Données, j'ai été confronté à certaines problématiques liées à la compréhension du langage naturel par un système en particulier: les agents conversationnels. La suite de ce mémoire aura par conséquent comme I conducteur, les challenges rencontrés dans la mission d'optimisation de la compréhension d'un chatbot. Ce tome ne fait pas exactement référence au mémoire de 2018 mais complète ce dernier.

Dans un premier chapitre, nous définirons ce que signifie « comprendre » une phrase pour un système informatique. Pour ce faire nous aborderons certaines techniques et algorithmes mis en oeuvre dans le cadre de la compréhension du langage. Par la suite, il s'agira de traiter la problématique d'optimisation de cette compréhension. Enn dans un dernier chapitre je tenterai d'apporter une solution personnelle alternative à celles présentées précédemment pour afner la compréhension du langage naturel par les systèmes conversationnels.

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

7

1. Compréhension du langage naturel

An de poser un contexte commun avec le lecteur, seront exposées dans cette partie, différentes « tâches » de traitement automatique du langage qui concourent à

former un modèle de compréhension du langage naturel. Pour remplir ces tâches, il est souvent nécessaire de remettre en question le type de modélisation utilisée, notamment la représentation des mots eux mêmes.

Une fois un mot, une phrase, ou un document modélisé, il faut exploiter cette représentation dans un but d' extraction d'information. Sémantique, syntaxe, polarité, thématique, ou entités nommées, toutes ces informations vont permettre à un système d'interpréter la requête d'un utilisateur humain an d'y répondre. Nous aborderons en particulier les techniques de reconnaissance d'entités nommées ainsi que la classification en intentions (thème abordé au travers d'une requête utilisateur).

Enn, dans le but de faire le lien avec l'industrie du NLP, certains outils proposant des modèles de compréhension du langage naturel pré-implémentés seront présentés.

De nombreuses pistes d'optimisation des modèles de NLU ne seront pas traitées ici, notamment les pré-traitements liés au découpage de documents en tokens, leur racinisation, et bien d'autres paramètres inuençables par les développeurs pouvant améliorer la qualité d'un modèle.

1.1. Représentations d'un mot

Il n'est pas de modélisation idéale du mot dans l'absolu. En effet, le choix de représenter un mot pour la machine d'une manière ou d'une autre, est fortement dépendant du type de tâche à réaliser.

Pour un cas d'usage bien précis, et dans un contexte de vocabulaire très réduit, il peut s'avérer pertinent d'utiliser la représentation en vecteur « one-hot ».

Cette représentation du mot est construite en fonction de la taille du vocabulaire, chaque mot est représenté par un vecteur de cette taille. L'indice 1 du vecteur représente le premier mot du vocabulaire, l'indice 2 correspond quant à lui au deuxième mot du vocabulaire et ainsi de suite. De cette façon, tous les éléments du vecteur représentant un mot sont à 0, sauf celui correspondant à la position du mot dans le vocabulaire. La gure 1.1 illustre la modélisation des mots d'un vocabulaire de 3 mots. Le vocabulaire utilisé est le suivant: {«processeur », « mémoire », « serveur »}

processeur

Mémoire

serveur

[1,0,0]

[0,1,0]

[0,0,1]

Fig 1.1. Représentation des mots en vecteurs creux

Cette représentation des mots d'un vocabulaire est certes pratique pour la machine de part sa propriété binaire, cependant elle présente de nombreux désavantages dans des contextes rencontrés fréquemment en traitement du langage. Le vecteur devient très lourd à manipuler dès que la taille du vocabulaire augmente, ce qui le rend inutilisable pour des

systèmes complexes de modélisation de la langue. Mais de plus, cette modélisation ne

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

8

permet en aucun cas de capter la sémantique d'un mot. Dans l'exemple précédent, il serait impossible de distinguer le mot serveur informatique du mot serveur en restauration. Et ce, du fait que ce processus de modélisation ne prend pas en compte les mots environnants pour former la représentation.

Dans un souci d'amélioration d'un système de compréhension du langage, la représentation en vecteurs « one-hot », présente de nombreux défauts auxquels d'autres représentations peuvent pallier. Nous recherchons au travers d'une représentation numérique, à capter du sens, et des relations.

Pour capter des relations, il est possible de construire une matrice de cooccurrence. Pour cela il faut convenir que le modèle se construit sur une base documentaire contenant des phrases, permettant ainsi de calculer les cooccurrences des mots entre eux. Cette méthode renvoie un mot, à son vecteur représentant une mesure d'appartenance à un contexte particulier.

Les avantages de cette technique provoquent également ses inconvénients. Par exemple la taille de la matrice explose avec la taille du vocabulaire, malgré tout, pour que les cooccurrences des mots soient porteuses de sens, il faut que le vocabulaire soit de taille suffisante. Cette base documentaire doit donc atteindre un équilibre entre qualité et taille raisonnable. Cet équilibre, pour être atteint demande de faire des choix quant aux types de phrases utilisées, il faudra par exemple sélectionner un corpus à thème pour que le modèle soit efficace sur ce thème, le registre également doit être bien choisi, entre un registre soutenu et un registre commun, les cooccurrences diffèrent. C'est pourquoi la matrice de cooccurrences représente également un choix d'implémentation pertinent en fonction du cas d'usage associé.

Plus largement, les matrices de cooccurrences peuvent calculer les cooccurrences sur les phrases, les paragraphes, ou même les documents.

Il est également nécessaire de considérer la modélisation non pas seulement des mots seuls, mais de construire un modèle basé sur des n-grammes (suite de taille n de mots qui se suivent dans une phrase) . Cela permet d'ajouter un contexte aux occurrences des mots d'un document, ou d'une phrase. Toute modélisation peut être étendue aux n-grammes.

Nous avons jusque là abordé des représentations de mots construites à partir de comptages. D'autres approches mettant en oeuvre des algorithmes d'apprentissage profonds existent et sont très utilisées de nos jours dans la plupart des applications du TAL. C'est la méthode Word2Vec qui sera développée dans la suite de ce chapitre. L'objectif de Word2Vec et d'autres méthodes de prolongements lexicaux est d'associer à

chaque mot un vecteur dans un espace continu. Deux mots ayant un sens comparable, doivent être proches dans l'une ou plusieurs des dimension de cet espace. Cet espace vectoriel résultat du passage de l'algorithme sur un corpus d'apprentissage doit permettre des opérations logiques sur les vecteurs comme l'illustre la gure 1.2.

$\text{Vec}(\text{« chauve »}) - \text{Vec}(\text{« personne »}) = \text{Vec}(\text{« cheveux »})$

Fig 1.2. Liens sémantiques construits via Word2Vec

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

9

L'algorithme word2vec utilise l'algorithme skip-gram qui doit être expliqué au préalable.

Nous cherchons à obtenir une représentation de mots sous forme de vecteur distribué, pour cela il faut malgré tout commencer par encoder les mots d'une manière compréhensible pour une machine, et il est d'usage d'utiliser l'encodage « one-hot » dans un premier temps.

Couche cachée, n neurones

linéaires, correspondant à la

dimension souhaitée du vecteur

v neurones de sortie, fonction

d'activation softmax pour récupérer une

probabilité (somme a 1)

« capillaires »

probabilité qu'un mot

environnant soit:

« souris »

probabilité qu'un mot

environnant soit:

« tête »

Vecteur d'entrée

encodé en one-hot.

De dimension v

(taille du vocabulaire)

probabilité qu'un mot

environnant soit:

« maintenant »

Fig 1.3. Réseau de neurones Skip-Gram

La gure 1.3 illustre le fonctionnement du réseau de neurones à une couche cachée (sans

fonction d'activation) Skip-gram. Il a pour entrée un mot du vocabulaire encodé en « onehot » et en sortie un vecteur de la taille du vocabulaire contenant les probabilités de

chaque mot du vocabulaire d'appartenir au contexte du mot en input. En somme c'est un

réseau de neurones simple permettant de prédire le contexte d'un mot.

Contrairement aux modélisations précédentes issues de comptage d'occurrences et de cooccurrences de mots qui se font par « batch » i.e. en une fois sur l'ensemble des données d'apprentissage, cette méthode se fait par itération sur des lots de données, ce qui permet d'affiner les poids de la couche cachée. En effet ce réseau de neurones a pour sortie une donnée certes intéressante: le contexte probable d'un mot. Cependant ce qui nous intéresse particulièrement ici est la couche cachée, car elle contient la représentation vectorielle des mots du vocabulaire.

En effet durant l'apprentissage de ce réseau de neurones, la couche cachée constitue une matrice de taille $VOC \times dim$, avec VOC la taille du vocabulaire et dim , la dimension des prolongements lexicaux souhaitée (correspondant au nombre de neurones de la couche cachée).

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

10

C'est en fait la matrice des prolongements lexicaux de chaque mots du vocabulaire. Et cette matrice ajuste ses poids par itération en comparant ses prédictions à une fonction objectif.

La vertu de Skip-Gram est due au fait qu'il s'entraîne à prédire le contexte d'un mot au travers d'un vecteur de sortie. En ajustant sa couche cachée dans ce sens, des mots ayant le même contexte doivent avoir des vecteurs de sortie proches, or des mots ayant le même contexte ont probablement également un sens proche.

Word2Vec extrait la couche cachée de Skip-Gram pour l'utiliser comme représentation vectorielle des mots du vocabulaire. Cette méthode a l'avantage de fournir une représentation dense des mots et de capter des similarités ou dissimilarités entre eux, ce qui constitue une nette amélioration face aux méthodes précédemment citées.

Finalement, nous avons parcouru quelques principes de modélisation de mots, mais il en existe de nombreux autres, le choix d'une représentation ou d'une autre appartient aux architectes de solutions, aux développeurs, et est motivé par des cas d'usages précis. Choisir la représentation la plus efficace en algorithmie est généralement question de compromis.

Après cette brève présentation des méthodes de modélisation du langage pour la machine, la section suivante décrira par quels moyens, il est possible d'extraire de l'information à partir d'un corpus ou d'une phrase.

1.2. Extraction d'information

Le terme d'information contenue dans une phrase est le plus large possible.

L'information portée par le langage naturel est si vaste que les algorithmes d'aujourd'hui ne permettent d'en capter qu'une infime partie. Des noms, des dates, des adresses, des urls, des institutions, ou bien des nombres. Mais aussi des sentiments, de l'ironie, un imaginaire associé à un mot ou une formulation, une odeur, un souvenir commun ravivé par une expression. Toutes ces données potentiellement transportées par une phrase ne

sont pas extraites ni identifiables par apprentissage automatique. Cependant, les outils développés actuellement permettent d'ores et déjà d'extraire de nombreuses informations issues de données textuelles. Et c'est à partir de là, par petits incréments, que la machine pourrait comprendre un jour ce que l'on souhaitera lui exprimer.

L'extraction d'information la plus commune concerne deux informations portées par la totalité des phrases ou documents possibles. Il s'agit de l'intention et des entités.

L'intention d'une phrase représente le thème abordé, ou plus simplement, l'objectif de l'utilisateur (d'un agent conversationnel) au travers de sa phrase. Les entités, elles, représentent les variables de la phrase, les données « intéressantes » pour la compréhension. Par exemple dans la phrase « quel temps fait-il à Paris ? », l'intention correspond à une demande de renseignement météorologique, et l'entité contenue est une ville qui a pour valeur Paris.

La suite de ce chapitre détaille certaines techniques d'extraction d'information appliquées à la reconnaissance d'intentions et d'entités. Ceci constituera une base pour l'optimisation des modèles de compréhension du langage naturel.

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

11

1.2.1. Reconnaissance d'intentions

La classification de texte en termes d'intentions consiste donc à identifier le thème auquel se rattache une citation. Pour cela il est possible de considérer une approche non supervisée en découpant l'ensemble d'un corpus en clusters, différents algorithmes de classification non supervisée conviennent pour ce type de tâches tels que les K-moyennes, ou les mixtures gaussiennes. Malgré tout, dans un système de compréhension du langage naturel, il convient de prédéfinir les différentes intentions qu'un utilisateur peut exprimer. Le clustering non supervisé ne permet pas de prédéfinir ce qui rapproche deux phrases. En effet les groupes constitués par de tels algorithmes ne sont pas systématiquement interprétables. Par exemple dans un corpus de phrases issues des FAQ d'une université, les questions traitent généralement de l'administration, des UE enseignées par formation, et des activités associatives proposées. Si l'on applique un modèle de clustering non supervisé sur les questions de ce corpus en lui signifiant de créer 3 groupes dans le but de représenter les 3 thèmes abordés par les étudiants, il est possible que le système renvoie trois groupes correspondant respectivement aux phrases courtes, aux phrases longues, et aux phrases moyennes. Ce découpage n'aurait pas d'utilité dans un objectif de compréhension du langage. C'est pourquoi la tâche de reconnaissance d'intention est généralement complétée par des algorithmes d'apprentissage supervisé.

À la place de classer des phrases de manière supervisée, il faut constituer un corpus d'apprentissage annoté. Chaque élément de ce corpus sera composé d'un couple (phrase , intention). La phrase doit être encodée à partir des vecteurs des mots qui la

constituent. Il est fréquent de calculer le vecteur moyen des mots de la phrase pour la représenter. L'intention, elle, sera représentée par un entier.

Une fois un corpus d'apprentissage construit il faut choisir un algorithme de classification adapté.

Un algorithme qui se prête bien à la classification d'intentions est le SVM, les machines à vecteurs de support. Considérant l'espace vectoriel associé à un modèle de prolongement lexical, l'objectif de cette démarche est de trouver les hyperplans qui séparent le plus précisément possible les vecteurs moyens représentant les phrases à classier. Plus la distance entre les éléments d'une intention et l'hyperplan est grande, plus il est bien placé (moins il fera l'objet d'overfitting). On appréhende ce modèle en « one versus all », c'est à dire que si l'on a 10 intentions à reconnaître, on créera 9 modèles, chacun discriminant une intention face aux 9 autres sans faire de distinction entre les 9 intentions restantes.

110 kg

Fig 1.4. Illustration SVM

hommes

femmes

90 kg

70 kg

50 kg

150 cm

162,5 cm

175 cm

187,5 cm

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

200 cm

12

Sur la figure 1.4., des individus sont représentés par leur poids et leur taille, on utilise cet exemple en deux dimensions à des fins pédagogiques, mais il faut considérer que les points de ce graphique peuvent être des phrases à classier encodées en vecteurs à x dimensions. La droite quant à elle serait alors remplacée par un hyperplan.

L'apprentissage d'un modèle SVM se fait via la résolution d'un problème d'optimisation sous contraintes. L'objectif de la résolution de ce problème est de maximiser la marge (distance entre les deux droites vertes). Les contraintes qui lui sont associées représentent le fait que les points les plus proches des droites vertes (support de vecteurs) ne doivent pas être situés dans la marge. La résolution de ce problème d'optimisation se fait grâce aux facteurs de Lagrange.

Plutôt qu'une explication mathématique, voici des éléments pratiques permettant de comprendre ce qui est réalisé par une machine à vecteurs de support. Le problème

initial de reconnaissance d'intentions a été ramené à un problème de classification binaire. La figure 1.5. affiche la matrice de dispersion des vecteurs représentant des critiques sur des lms. Ces critiques peuvent être positives ou négatives et ont été préalablement nettoyées de certains mots de liaison, urls, liens. Par la suite elles ont subi un encodage grâce à l'implémentation de doc2vec, parent proche de word2vec appliqué à des paragraphes. Chaque paragraphe est donc représenté par un vecteur de 10 dimensions.

On observe grâce à cette matrice de dispersion, que certaines dimensions de l'espace vectoriel prises 2 à 2 permettent de voir une nette séparation entre les points jaunes (critiques positives) et les points noirs (critiques négatives). Cela signifie que la représentation numérique utilisée a capté suffisamment de sémantique pour rapprocher géographiquement les critiques semblables dans l'espace vectoriel généré. Ainsi, la machine à vecteurs de support sera capable d'exploiter une combinaison linéaire des dimensions les plus séparatrices pour créer l'hyperplan optimal.

Fig 1.5. Matrice de dispersion des prolongements lexicaux des critiques de lms

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

13

1.2.2. Reconnaissance d'entités

Autre information riche d'indices pour un système de compréhension du langage, l'entité nommée. Les entités nommées peuvent être des lieux, des dates, des organisations commerciales ou étatiques, des marques, des produits ou encore des adresses IP, des GB, des noms propres.

Une fois encore, de nombreuses approches sont envisageables pour extraire des entités nommées et c'est pour cette raison que cette partie d'un système de compréhension du langage naturel représente un levier d'optimisation.

Une première approche pour un cas limité à la reconnaissance des noms de pays par exemple, consiste à stocker un dictionnaire contenant la liste des pays du monde. La comparaison des phrases d'un corpus à ce dictionnaire permet d'extraire les noms de pays de ce corpus.

Cependant, ce type de système présente de nombreuses faiblesses liées au besoin de maintenir une liste d'objets à comparer au corpus. Dès lors que le cas d'usage s'élargit à la reconnaissance de nombreuses autres entités, il ne devient plus possible de lister toutes les valeurs que peut prendre un nom de produit par exemple. De plus une faute d'orthographe sur un nom de pays empêcherait le système de reconnaître ce dernier, même si le contexte indique que c'est effectivement le pays dont il est fait mention.

Une légère amélioration de la reconnaissance d'entités peut se faire en complétant le

système précédent par des expressions régulières afin de représenter le maximum de valeurs possibles. Cette piste a également ses limites et n'offre pas de performances suffisantes pour un bon niveau de compréhension.

La suite de ce chapitre expose les champs conditionnels aléatoires, une méthode statistique éprouvée et efficace en extraction d'entités nommées.

Les implémentations de CRF (champs conditionnels aléatoires) sont des algorithmes très utilisés également en modélisation de l'ADN et plus généralement dans le cadre de la reconnaissance d'information dans des données séquentielles. La théorie autour des CRF consiste à évaluer la probabilité conditionnelle suivante:

$$P(y_0, y_1, \dots, y_T \mid x_0, x_1, \dots, x_T)$$

Pour illustrer ce que contient cette formule, voici ce qu'un CRF permet d'obtenir pour une phrase en entrée.

Input :

Output :

[« ce », « lm », « produit », « par », « Clint », « Eastwood »]

[RAS, RAS, RAS, RAS, PROD, PROD]

Les x_i représentent la séquence d'entrée tandis que la sortie correspond aux y_i .

En somme un CRF permet à partir d'une séquence en entrée, d'associer à chaque terme de cette séquence, un label. Pour cela, le CRF prend en compte le contexte autour d'un mot.

Ici la séquence « produit par... » implique généralement que la chaîne de caractères suivante soit un producteur (de cinéma).

Pour obtenir ce type de résultats le CRF se base sur un corpus d'apprentissage.

OPTIMISATION DES MODÈLES DE COMPRÉHENSION DU LANGAGE NATUREL

14

Le CRF dispose de multiples leviers d'optimisation, notamment l'enrichissement du contexte via l'augmentation des caractéristiques des mots. Ceci sera développé dans la partie suivante, « optimisation d'un modèle de compréhension du langage ».

1.3. Services de compréhension du langage

Un modèle de compréhension du langage comporte un ensemble de données et de programmes permettant d'extraire de l'information sémantique à partir d'un texte en langage naturel.

Le marché du NLP est en plein essor, et les entreprises proposant des outils de compréhension du langage naturel sont de plus en plus nombreuses.

Ces outils souvent accessibles sous forme d'API, utilisent les principes et algorithmes cités précédemment avec quelques spécificités.

Ce chapitre est dédié à la description des plateformes de NLP.

Amazon Comprehend, IBM Watson Natural Language Understanding, ou Google Cloud

Natural Language proposent des solutions cloud pour le traitement automatique du langage naturel.

Google met à disposition des développeurs d'applications, une API de machine learning nommée « Natural Language ». Natural Language permet l'analyse de sentiments, l'extraction d'entités, l'analyse de syntaxe et la classification thématique de contenus.

Cette API, n'est pas open source, et s'utilise en boîte noire. Un appel à l'API renvoie la réponse du système. Il n'est pas possible par exemple d'utiliser un algorithme ou un autre pour classer du texte, le choix est prédéfini et optimisé par Google. Cependant, il est toujours possible de soumettre son propre corpus d'apprentissage annoté pour personnaliser la classification.

Cet ensemble d'outils et de fonctions forme un système de compréhension du langage naturel. Si l'on reprend l'exemple de l'analyse de sentiment précédent, ce système a l'avantage de permettre aux développeurs d'envoyer un texte via l'API, et de recevoir en retour une estimation du sentiment exprimé dans ce texte sous forme de score et de magnitude. Le score correspond au degré de positivité/négativité, et la magnitude représente la charge émotionnelle du texte. Le couple score et magnitude représente pour Google le sentiment exprimé dans un texte. L'unique paramètre ajustable pour l'utilisateur de l'analyse de sentiment Google, est le seuil au-delà duquel le score d'une phrase la classe comme étant positive ou négative.

En somme le développeur consomme un service sans se soucier de l'implémentation du modèle de compréhension du langage. Cela représente un gain certain en termes de temps de développement.

Les étapes de récolte de données massives (web scraping, achat de données), de nettoyage de corpus (tokenisation, lemmatisation, racinisation, choix des stop-words), de modélisation de la langue (n-grammes, TFIDF, Word2Vec), et de classification ou régression (association d'un score à un texte) ne sont donc plus à la main des concepteurs de l'application mais sont réalisées par Google.

Les services de compréhension du langage n'embarquent que très peu de leviers d'optimisation, ainsi, pour des besoins spécifiques il est généralement recommandé de concevoir son propre pipeline de compréhension du langage.

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

15

Bien qu'ils ne contiennent pas de leviers d'optimisation, ces services doivent être utilisés pour comparer les performances d'un système personnalisé. Si un système généraliste tel que celui de Google se révèle plus performant qu'un système implémenté localement avec des jeux de données adéquats, c'est un signe qu'il faut chercher à optimiser les performances du système cible.

Les services de traitement du langage ont donc également un rôle à jouer dans

l'optimisation d'un système de compréhension du langage.

2. Optimisation d'un modèle de compréhension du langage

Le 3 juillet 2019, des chercheurs de l'université de Berkeley, ont optimisé leurs algorithmes et jeux de données, au point de prédire de nouvelles découvertes dans le champs de l'étude des matériaux. Word2Vec, brique principale de cet algorithme a été « tuné » dans le but d'ingérer des articles scientifiques et d'établir des liens entre les mots, ces liens sémantiques ont par exemple permis au système de reconstituer le tableau périodique des éléments de Mendeleïev.

Concrètement, le système développé a analysé des articles scientifiques antérieurs à 2009 et grâce à la capacité de Word2Vec de prédire le contexte et les mots suivants, il a été capable de recommander un matériau découvert en 2014.

Cela suggère que l'adaptation d'un algorithme d'apprentissage automatique ou la sélection d'un corpus d'apprentissage a une importance majeure dans la compréhension d'un système.

Dans cette partie, il sera question des caractéristiques d'un corpus d'apprentissage, ainsi que des paramètres à optimiser pour augmenter les performances d'un système de compréhension du langage naturel.

2.1. Techniques liées aux corpus d'apprentissage

Depuis les années 70, en informatique, les techniciens sont d'accord sur l'affirmation suivante « Garbage in, garbage out » attribuée à Wilf Hey (ingénieur IBM en 1965). Appliqué à l'apprentissage automatique, cela donne: « Good data trains good models ».

Le premier point d'optimisation à considérer réside dans la qualité des données d'apprentissage elles-mêmes. En effet tout algorithme d'apprentissage se base sur des observations issues des datasets d'apprentissage. Un dataset biaisé fournira un modèle biaisé. Ce chapitre abordera deux des cinq « V » du big data, à savoir le volume et la véracité des données.

2.1.1. Volume

Le volume concerne deux aspects des données, le volume global d'instances dans le jeu de données mais aussi la distribution des exemples d'apprentissage. Une bonne répartition des classes à prédire prévient certains problèmes lors de l'apprentissage. On

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

16

cherche à travers un jeu de données équilibré, à ne pas fournir d'a priori au modèle quant à la tendance générale à appartenir à une classe ou à une autre.

Le déséquilibre des jeux de données d'apprentissage est un problème rencontré fréquemment lors de l'élaboration de modèles de classification. En effet pour le cas d'un

agent-conversationnel devant classer des phrases en intentions, les intentions que doit traiter le système sont rarement distribuées uniformément. Les utilisateurs d'un chatbot d'université ont majoritairement tendance à poser des questions concernant les contacts et les formations dispensées. Les autres intentions telles que les renseignements sur l'histoire de la faculté et les questions portant sur l'incubateur de startup sont bien moins nombreuses. Or il est nécessaire que le chatbot reconnaisse toutes ces intentions sans présupposer que toutes les questions des utilisateurs traiteront d'un seul sujet, le plus représenté.

Pour prévenir l'introduction de biais, il existe différents outils, dont l'influence sur les performances en classification seront présentées.

Voici en figure 2.1, la distribution d'un jeu de données d'apprentissage déséquilibré.

Ce jeu de données est issu des questions fréquemment posées par des collaborateurs d'Orange. Chaque intention est identifiée par un entier en abscisse, tandis que la quantité d'exemples de chaque intention est représentée en ordonnée.

Fig 2.1. Distribution des
exemples d'un corpus
d'apprentissage déséquilibré

Chaque instance de ce jeu de données est une question en langage naturel, encodée via Word2Vec. Une fois traitée pour alimenter un algorithme d'apprentissage automatique, ici, une machine à vecteurs de support, nous obtenons un système de classification biaisé comme l'indique la figure 2.2.

Le manque d'exemples d'intention 1, 6, 7 et 12, ne permet pas de prédire la classe de ce type de phrases. Ces intentions obtiennent une précision de classification nulle.

A contrario, nous observons que la classe 11, la plus représentée, a une précision inférieure à son rappel, ce qui signifie que le système a une tendance à classer d'avantage de phrases dans la classe 11. C'est une observation qui illustre l'a priori du système.

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

17

Fig 2.2. Métriques d'évaluation par
classe, d'un modèle SVM, sur un Dataset
déséquilibré

Si le volume du jeu de données le permet, c'est-à-dire s'il est assez grand, il est envisageable de supprimer des données aléatoirement dans les classes sureprésentées.

Malgré tout, dans la plupart des cas, la donnée est une denrée rare ou chère, et cette approche ne sera pas développée notamment car elle présente l'inconvénient de potentiellement supprimer des échantillons riches d'information.

Dans le but d'optimiser les performances de classification de ce type de système, il est possible de créer des données artificielles. À partir du jeu d'apprentissage existant, la

méthode SMOTE (Synthetic Minority Oversampling Technique) permet de générer des données, en l'occurrence ici des vecteurs représentant des questions. Ces données sont créées de la manière suivante: SMOTE sélectionne un individu d'une classe sous représentée et identifie n voisins. Pour générer un nouveau vecteur artificiel à partir de là, la différence entre le vecteur sélectionné et un de son voisin est calculée puis multipliée par un facteur compris entre 0 et 1. Ainsi, la génération de données artificielles par SMOTE, se fait sur des axes passant par deux données réelles, ce qui garantit, une relative cohérence.

La figure 2.3 présente la distribution des données d'apprentissage du chatbot, suite à l'application de SMOTE.

Fig 2.3. Distribution des classes d'un

Dataset après application SMOTE

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

18

Après avoir appliqué cette méthode, sur un jeu de données asymétrique (classification binaire) ou déséquilibré (classification multi-classes), les performances du système sont nettement améliorées.

La figure 2.4 est le compte rendu des métriques du même système que celui de la figure 2.2, appliqué au nouveau jeu de données contenant des données artificielles. On observe que le système est désormais capable de classer une donnée en classes 1, 6, 7, ou 12. La précision globale, elle augmente. Les apriori venant du déséquilibre initial, sont estompés.

Fig 2.4. Métriques d'évaluation par

classes, d'un modèle SVM, sur un

Dataset équilibré

L'intention 1 (ou classe 1) reste quant à elle difficile à classer, cela peut être dû à d'autres facteurs, notamment au type d'information qu'elle contient. Si les questions d'intention 1 n'ont pas assez de particularités communes. C'est à dire que chaque question est très différente des autres questions de cette même classe. Alors un moyen d'amélioration du système serait de découper cette intention en plusieurs intentions distinctes, dans le but que chaque nouvelle intention ne regroupe que des phrases ayant des points communs. Ceci constitue également une méthode d'optimisation des performances d'un système de compréhension du langage qui s'applique de préférence à l'origine de la conception du système

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

19

2.1.2. Véracité

Annotations hasardeuses, biais, données manquantes, de nombreux problèmes

surviennent lors de l'élaboration d'un jeu d'apprentissage. Faut-il conserver une donnée qui semble contredire l'idée que l'on se fait du modèle ? Faut-il souhaiter le volume au détriment de la véracité des données ? A quel titre faudrait-il remettre en question l'intégrité de données issues du monde réel ?

Ce chapitre ne prétend pas apporter de réponses à ces questions, mais a pour but d'exposer les conséquences possibles de l'introduction de données « de mauvaise qualité » dans un jeu d'apprentissage.

L'ACPR (Autorité de Contrôle Prudentiel et de Résolution de la banque de France), a dans son rapport « Intelligence artificielle : enjeux pour le secteur bancaire » de décembre 2018, cité le problème de la qualité des données. En effet dans cette étude il fait mention de la véracité des données dans un contexte de prêt bancaire. Si les prochains conseillers sont des algorithmes d'intelligence artificielle, alors quelles données d'apprentissage leur fournir sachant que les données dont nous disposons, contiennent de nombreuses variables discriminatoires telles que la nationalité, le département, l'âge ou le sexe. Dans ce cas comment garantir que les données d'apprentissage ne contiennent pas de biais de nature misogyne ou raciste ? Les algorithmes étant conçus pour apprendre à partir de comportements humains, ce problème se pose pour différents domaines.

Heureusement les risques d'une donnée non véridique pour un agent conversationnel, restent limités. Seules les performances du système de compréhension en seraient affectées.

Dans le cadre des agents conversationnels, un corpus de mauvaise qualité se traduit par un ensemble d'intentions mal conçu. Un mauvais découpage des thèmes abordés, mène à un modèle approximatif. Les intentions deux à deux doivent comporter le plus de différences possible en terme de vocabulaire, formulations, tons. Si un analyste de données humain, ne parvient pas facilement à déterminer à quelle intention appartient une question, alors un algorithme de classification tel que SVM aura également des difficultés à la classer.

Au delà d'un découpage judicieux des intentions, le choix des phrases utilisées pour l'apprentissage doit être en accord avec le cas d'usage du chatbot. En effet si un agent conversationnel a pour but d'orienter un salarié dans des processus métiers, alors le corpus doit contenir des questions qu'un utilisateur pourrait potentiellement poser. Les FAQ, les emails, et les historiques de skype ne suffisent pas à représenter le champ des possibles en terme de questions utilisateurs. Au lieu de se rapprocher au plus des futures questions posées au chatbot, il faut mettre en situation des utilisateurs « cobaye ». Ce processus de simulation de conversation permet d'ajouter des questions posées à un chatbot, car les sources de données classiques citées précédemment ne sont pas issues de conversations entre humains et chatbots, or un utilisateur ne s'adresse pas de la même façon à un collègue par e-mail, qu'à un agent conversationnel dans une interface de chat.

La simulation de conversation permet d'ajouter de la véracité au corpus d'apprentissage.

Enn, an d'assurer une annotation sans failles, il est possible de procéder à des triples annotations des intentions et des entités dirigées par un guide d'annotation précis.

Moins le guide d'annotation permet l'interprétation, plus il est efficace. Cependant ce procédé est coûteux en ressources humaines et en temps.

Il n'y a donc pas d'indice de mesure de la véracité d'un corpus d'apprentissage, elle est strictement relative au cas d'usage.

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

20

2.2. Techniques liées aux algorithmes d'apprentissage

Les algorithmes utilisés pour la classification d'intentions et pour la reconnaissance d'entités disposent de nombreux paramètres influençant les performances d'un système de compréhension du langage. Le cas de la reconnaissance d'entités nommées via champs conditionnels aléatoires sera développé dans ce chapitre.

Pour rappel un CRF prend comme données d'apprentissage des phrases annotées du type suivant:

`[(mot_1, entité_1), ... ,(mot_n, entité_n)]`

Les entités peuvent être nulles ou peuvent représenter une information (date, lieux, noms).

Pour entraîner un modèle de reconnaissance d'entités à partir de ce corpus, l'implémentation du CRF de Sklearn permet de choisir les « features » permettant de prédire au mieux le label d'un mot d'une phrase. Ces features donnent pour chaque mot d'une phrase, des caractéristiques supplémentaires qui contextualisent le mot. Grâce à ces features le CRF a plus de moyens pour découvrir des schémas dans la succession d'entités.

Voici une liste de features susceptibles d'améliorer la reconnaissance d'entités par un CRF:

Le mot commence-t-il par une majuscule ? (caractéristique des noms propres)

Le mot est-il placé en premier dans la phrase ou en dernier ? (certaines entités ont d'avantage tendance à être placées en début ou en fin de phrase)

Le mot contient-il des caractères spéciaux ?

Le mot est-il entièrement écrit en lettres capitales ? (caractéristique des noms d'organisation, ou des acronymes en général)

Le mot est-il un verbe à l'infinitif ? au participe passé ?

Toutes ces informations sur les mots d'une phrase sont déjà contenues dans le corpus d'apprentissage, encore faut-il les rendre lisibles pour que l'algorithme puisse exploiter ces indices.

L'ensemble de ces données va permettre au CRF d'observer des patterns dans les

phrases et d'extraire les entités recherchées.

Si l'on considère uniquement la séquence de mots sans augmenter les informations, la précision et le rappel d'un système de reconnaissance d'entités seront limités comme le montre l'exemple suivant.

Dans cet exemple, un corpus de phrases labélisées par leurs POS Tag (Part Of Speech tags) sert à l'apprentissage du modèle. Le but du modèle est de reconnaître les entités (ici les étiquettes POS). Au lieu de montrer la capacité d'optimisation d'un CRF, deux modèles ont été entraînés différemment. Le premier modèle extrait uniquement les séquences de mots sans ajouter de caractéristiques supplémentaires (présence de majuscules, début ou fin de phrase).

Fig 2.5. Fonction de caractérisation des mots du corpus (de base)

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

21

En figure 2.5, la fonction `features` renvoie les caractéristiques de chaque mot d'une phrase. Ici les caractéristiques renvoyées se limitent au mot suivant. Ainsi ce modèle ne considérera que les enchaînements de mots pour déterminer la présence d'entités.

Les données d'entraînement dans ce cas ont la forme suivante:

Variables d'entraînement

Label (entités)

Après entraînement du CRF sur ce modèle de données, nous obtenons un f1-score de 0.54 très faible. Pour améliorer ce score, il faut fournir plus d'informations au CRF sur les liens qui existent entre les mots. La nouvelle fonction `features` de la figure 2.6 crée un ensemble de caractéristiques plus complet :

Fig 2.6. Fonction de caractérisation des mots du corpus (augmentation de séquence)

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

22

On observe que cette fonction associe à chaque mot du corpus, des compléments d'information, notamment si il contient un tiret, un chiffre, une majuscule, le mot suivant, le mot précédent etc ...

Avec cette quantité d'information supplémentaire, dans les mêmes conditions d'entraînement (hyperparamètres identiques) le CRF obtient un score nettement supérieur de 0.973 (F1-score).

Le temps d'entraînement de l'algorithme est néanmoins beaucoup plus long étant donné la quantité d'information supplémentaire à traiter.

Le CRF lors de son entraînement considère alors les données suivantes:

Variables d'entraînement:

label (entités)

Le CRF a donc la possibilité de s'adapter à des données plus ou moins exhaustives, cet algorithme a de nombreux leviers d'optimisation, notamment ses hyperparamètres.

An d'optimiser l'entraînement de cet algorithme il faut tester différents paramètres successivement et relever la performance globale du système de reconnaissance d'entités.

Pour ce faire le processus GridSearch permet de tester de nombreuses combinaisons de paramètres automatiquement.

Il y a quatre éléments à faire varier dans un processus de GridSearch: La fonction objectif qui retourne un score rééant la performance du modèle, le domaine dans lequel faire varier les hyperparamètres, l'algorithme ou méthode permettant de choisir le prochain hyper paramètre à tester.

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

23

Fig 2.7. GridSearch

C1 et C2 sont des hyperparamètres (paramètres de l'algorithme CRF indépendants des données à xer avant l'entraînement).

An de choisir les valeurs idéales de C1 et C2, il est possible de procéder manuellement, à partir d'une intuition ou à partir de sa propre expérience au choix de ces paramètres. Cependant cet méthode présente le risque d'ignorer une combinaison de paramètres qui pourrait améliorer considérablement les scores d'un modèle. C'est là qu'intervient GridSearch, comme le montre la gure 2.7, avant d'entraîner un modèle, il faut spécier à la fonction GridSearch l'ensemble de paramètres à faire varier, ainsi qu'une liste de valeurs à tester.

Dans l'exemple 2.7 seuls deux des vingt hyperparamètres du CRF sont testés, bien qu'il soit possible au même titre de faire varier tous ces paramètres.

GridSearch renvoie alors la combinaison de paramètres C1 et C2 qui offre le meilleur score (le score peut être personnalisé entre précision, f1, rappel ...). Ainsi lors de l'entraînement sur le corpus d'apprentissage complet, ces valeurs de C1 et C2 seront connues et utilisées avec conance.

Les implémentations des champs conditionnels aléatoires (CRF) disposent d'un tel niveau de conguration, qu'il n'existe pas deux CRF semblables en termes de features et d'hyperparamètres, c'est pourquoi l'optimisation de ce type d'algorithmes représente une grande partie de la conception d'une application de traitement automatique du langage.

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

24

3. Proposition d'approche de construction d'un corpus métier

Beaucoup de données sont disponibles en open data, notamment des corpus de textes littéraires, des corpus issus de la presse internationale, des corpus rassemblant tous les articles Wikipédia. Ces données en accès libre permettent de nombreux

développements en traitement automatique du langage. Les notions généralistes abordées dans ces textes rendent possible une modélisation de la langue grâce à leur volume important. Cependant ces modélisations issues de tels corpus n'embarquent pas de connaissances sur des domaines très particuliers tels que les sujets métiers d'entreprises de secteurs divers et variés. Si bien que des notions inhérentes à l'activité professionnelle de chacun sont impossibles à capter au travers de ces corpus.

Ayant fait face à cette problématique lors de développements pour des métiers d'Orange, j'ai pu constater le manque de connaissances spécifiques, et le manque de données disponibles, à fournir à nos algorithmes d'apprentissage automatique.

Les assistants virtuels font leur entrée dans le panel d'outils mis à la disposition des salariés. La volonté de faciliter la tâche des collaborateurs à travers l'utilisation de chatbot est une réalité, seul le manque de corpus de données spécialisées représente un frein à leur démocratisation.

3.1. Motivation

Chaque métier a son jargon. Certains jargons sont répertoriés et connus de tous, car venant de métiers historiques. Chez Orange, à la Direction des Infrastructures (DIF), le vocabulaire utilisé pour désigner des réseaux, des équipements ou des zones logiques dans des data-centers, est parfois déroutant pour un système de compréhension du langage. En effet un vocabulaire exclusivement utilisé dans ce contexte ne se retrouve pas dans des corpus d'apprentissage généralistes en open data.

L'objectif de cette proposition est donc de donner la main aux experts métiers sur la constitution d'un corpus qui servira à l'entraînement de leur assistant virtuel.

Grâce à une application, les experts métiers pourront enseigner à leur assistant virtuel les celles de leur activité en rédigeant des phrases qui contextualisent leur jargon, ou bien en répondant à des questions.

Bien entendu cet ajout de phrases directement associées à un métier doit se faire vis à vis d'un corpus d'apprentissage généraliste, au quel les règles de base de syntaxe, de grammaire, et les liens sémantiques classiques soient toujours connus du modèle.

Comme nous avons montré l'importance des prolongements lexicaux (vecteurs mots) dans la compréhension d'un système, la proposition d'optimisation de ces vecteurs consiste à enrichir le corpus d'apprentissage avec des phrases utilisant les mots du jargon métier.

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

25

3.2. GCM

Cette outil (Générateur de Corpus Métier, GCM) de création de corpus d'apprentissage a pour fonction de générer des phrases ou questions à partir d'une liste de vocabulaire spécifique. Par exemple un directeur d'entité Orange souhaitant l'implémentation d'un assistant virtuel pour assister ses collaborateurs dans l'exécution

des processus devrait fournir une liste de vocabulaire métier à l'outil GCM.

A partir de cette liste, le GCM devra solliciter les acteurs du métier ciblé afin qu'ils produisent des phrases et questions contenant ces termes.

Pour solliciter les collaborateurs à même d'utiliser le vocabulaire à bon escient, le GCM devra être placé en amont de l'ouverture des outils du SI utilisés par ce métier. Il est envisageable qu'à chaque nouvelle ouverture d'une fenêtre d'un outil, le GCM propose à l'utilisateur de composer une phrase qu'il intégrera ensuite dans sa base d'apprentissage.

Voici le schéma explicatif du GCM en figure 3.1:

Bonjour, je vais vous poser
quelques questions pour apprendre
votre vocabulaire métier. Je vous
propose des mots, si c'est possible
formulez une phrase ou une
questions à partir de ces mots

OK, allons y

BLOC et BAIE

Un bloc se compose de deux
baies

C'est noté, maintenant: ZP

ZP est un type de bloc

Merci! Maintenant répondez à:

« Ou se trouve le CBI ? »

Sur piazza ...

Fig 3.1. GCM

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

26

Comme l'indique le schéma, le GCM peut prendre la forme d'une interface de chat.

Au travers de cette interface, le GCM prend aléatoirement un ou deux mots de la liste du vocabulaire métier, et propose à l'utilisateur de composer une phrase ayant un sens métier à partir de ces deux mots. La phrase ainsi générée peut être soit une phrase déclarative soit une question, le point d'interrogation identifiant cette dernière.

Les phrases générées sont directement ajoutées au jeu d'apprentissage généraliste.

Pour offrir plus de variété aux phrases générées, le GCM reprend chaque question qui lui a été fournie et demande à l'utilisateur d'y répondre, ainsi le corpus contient des questions contenant des mots du vocabulaire métier, mais aussi leur réponses qui sont également susceptibles d'enrichir le corpus.

Enn, car toutes les formulations possibles convergent rapidement, le GCM calcule la distance (de Jaro, de Levenshtein ou de Hamming) entre la phrase générée et les

phrases du corpus métier, afin de déterminer si une phrase soumise par un utilisateur n'est pas déjà contenue dans le corpus. Si tel est le cas, le GCM propose à l'utilisateur de reformuler sa phrase en d'autres termes.

Si un expert métier participant à ce processus, considère que les deux mots proposés pour former une phrase ne sont pas associables, alors il a la possibilité d'effectuer de nouveaux tirages jusqu'à ce que le GCM lui propose deux mots pouvant faire l'objet d'une même question.

Ainsi l'outil constituera peu à peu, en fonction du nombre d'utilisateurs quotidiens, une base de données suffisante pour contextualiser les termes de l'activité cible.

Le GCM n'est pas proposé sur le marché à ma connaissance.

L'outil GCM n'est certes pas une révolution dans la conception d'application de traitement du langage, néanmoins il va dans le sens de l'optimisation des systèmes pour lesquels encore peu de données sont disponibles, en créant de façon peu onéreuse, un jeu d'apprentissage thématique associé aux activités spécifiques et en constant renouvellement.

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

27

Conclusion

Etats, multinationales, PME, individus, tous participent à l'évolution de la compréhension du langage par les machines. Les états mettent à disposition quelques données, développent des algorithmes pour la sécurité nationale. Les multinationales et PME ciblent leur clientèle, évaluent les commentaires. Les individus, dernier maillon de la chaîne fournissent les données et utilisent les services.

L'optimisation des systèmes de compréhension du langage naturel est un travail qui a deux facettes. L'une d'entre elles, consiste à élaborer des modèles toujours plus performants au travers de méthodes statistiques et neuronales. Ce champ de la recherche évolue quotidiennement et les ressources investies ne cessent d'augmenter.

L'autre facette de l'optimisation de ces systèmes, réside dans l'utilisation qui en est faite. En effet, la mise à disposition massive d'information de services, d'objets connectés et d'assistants virtuels demande une certaine configuration pour et par l'utilisateur.

Le traitement automatique du langage évolue grâce aux avantages économiques qu'il confère aux entreprises qui les maîtrisent. Malgré tout, certains marchés spécialisés ne bénéficient pas encore de ces avantages car ils ne disposent pas d'historiques suffisants concernant leurs métiers, bloquant ainsi une transition vers ces systèmes.

Le GCM a pour vertu d'impliquer consciemment l'individu dans une démarche de construction de son service d'intelligence artificielle. Moins d'opacité et de délégation vers les grands acteurs du numérique, pour que l'individu ne soit plus le dernier maillon d'une chaîne, mais pour qu'il soit au centre de son prochain outil de travail assisté par IA.

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

BIBLIOGRAPHIE/WEBOGRAPHIE

- Skip gram

<http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/>

- Définition du perceptron

<https://openclassrooms.com/fr/courses/4470406-utilisez-des-modeles-supervises-nonlineaires/4730716-entraenez-un-reseau-de-neurones-simple>

- Premier papier sur CRF

<https://people.cs.umass.edu/~mccallum/papers/crf-tutorial.pdf>

- Chatito pour générer des datasets automatiquement

<https://rodrigopivi.github.io/Chatito/>

- Classification d'intentions

<https://blog.rasa.com/rasa-nlu-in-depth-part-1-intent-classification/>

- Hash de sous mots pour intention classification petits corpus

<https://arxiv.org/pdf/1810.07150.pdf>

- Doc de linear SVM

<https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learningalgorithms-934a444fca47>

<https://www.youtube.com/watch?v=TtKF996oEI8>

- Doc2vec

<https://radimrehurek.com/gensim/models/doc2vec.html>

- WORD EMBEDDINGS STATE OF THE ART ALGO ELMO :

<https://towardsdatascience.com/elmo-contextual-language-embedding-335de2268604>

- CRF enrichissements:

<http://www.albertauyeung.com/post/python-sequence-labelling-with-crf/>

- NLP par transfert

<https://weave.eu/deep-transfer-learning-nlp-revolution/>

- SMOTE

https://en.wikipedia.org/wiki/Oversampling_and_undersampling_in_data_analysis

- ACPR sur les biais dans les données:

<https://acpr.banque-france.fr/sites/default/les/medias/documents/>

2018_12_20_intelligence_artificielle_fr_0.pdf

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

GLOSSAIRE

ACPR

Autorité de Contrôle Prudentiel et de Résolution de la banque de France

API

Interface de programmation applicative

Clustering

partitionnement de données

CRF

Champs Aléatoires conditionnels

Dataset

Jeu de données d'apprentissage, de test ou de validation

FAQ

Frequently Asked Questions

Features

Caractéristiques/ variables explicatives

F1-score

Mesure de performance d'un modèle de prédiction. Rapport entre le rappel et la précision

GCM

Générateur de corpus métier

Hyper-paramètres

Ensemble de paramètres d'un algorithme d'apprentissage, xés avant l'apprentissage, ne découlant pas de l'apprentissage sur les données

Lemmatisation

Action de ramener un lexème au lève, ie, renvoyer un mot dérivé à sa déinition du dictionnaire

N-grammes

Ensemble de mots consécutifs de taille n.

NLP

Natural Language Processing

NLU

Natural Language Understanding

Overtting

Sur apprentissage, un modèle ne généralise pas et n'offre pas de bonnes performances sur des données réelles.

POS

Part of speech, labels correspondant aux Verbes, Noms, Adjectifs, Locutions, Adverbes, Déterminants etc ...

Racinisation

Élimination des radicals et sufixes d'un mots, ramener à la racine

Sklearn

Module Python de machine Learning

SMOTE

Méthode utilisée pour l'équilibrage d'un jeu de données. Synthetic

Minority Overs-sampling Technique

Stop-words

De, la , le ... mots très fréquents n'apportant pas ou peu de sens à
un phrase

SVM

Machine à vecteurs de support, ou séparateurs à vaste marge

TAL/TALN

Traitement automatique du langage/Naturel

Web-scraping

Scripting permettant la récolte de données sur le web

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

30

ANNEXES

Toutes les annexes et scripts ayant généré les résultats des expériences présentées dans
ce document sont disponibles à cette adresse :

https://github.com/obre/memoire_2019

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

31

OPTIMISATION DES MODÈLES DE COMPREHENSION DU LANGAGE NATUREL

32
