Faculté des Sciences et Technique de Mohammedia

Mémoire de projet cadre

La réalisation d'un classificateur NLP hybride entre la machine learning et le deep learning .

Author:
Mr ELKAISSI SOUHAIL

Supervisor:
Mme K.DOUZI

Ce Rapport correspond au stage de la deuxième année du cycle d'ingénieur Ingénierie Logicielle et Intégration des Systèmes Informatiques

in the

La Faculté des Sciences et Technique de Mohammedia

22 juin 2018

REMERCIEMENTS

Je tiens à remercier les personnes qui ont développé lors de mon étape au sein de l'équipe du SANADTECH et notamment les trois personnes qui ont accompagné accompagné, structuré et aidé pendent ses six mois:

Mon encadrant en entreprise, M. ABDELHAKIM RHANIZAR, Fondateur et directeur technique de l'entreprise qui m'a accompagné tout au long de ce stage, en me faisant découvrir toutes les technologies que j'ai pu maitriser au cours des semaines .
Mon encadrant à la faculté des sciences et techniques de Mohammedia, Mme KHADIJA DOUZI, qui a travers ses nombreuses conseils, et grâce aussi à sa confiance j'ai pu m'accomplir totalement dans mes missions.
M. CHATRIA Zakaria, le second développeur avec lequel j'étais en contact direct, qui m'a aidé dans mes tâches et qui m'a fait sortir de nombreuses situations délicates en m'aidant à résoudre les problèmes rencontrés.

Et je tiens à laisser une pensée pour les autres membres de l'équipe que je n'ai pas eu l'occasion de côtoyer souvent, mais que j'ai trouvé sympathique.

Résumé

Dans le cadre de mon stage à la société SanadTech, j'ai eu la chance de participer au projet Nacharat. Nacharat, en fait c'est une application mobile qui permet à ses utilisateurs de consulter les nouveautés sous format vidéo.
Dans mon entreprise d'acceuil SanadTech, j'ai travaillé au sein d'une équipe dirigée par Mr RHANIZAR, le développeur front-end Mr CHATRIA, et moi-même comme Data Scientist.
La méthode de gestion de projets appliquée pour le processus est la méthode agile Scrum. Cette méthode privilégie les objetctifs à court terme et donne lieu à des interactions régulières avec les membres de l'équipe afin de discuter de l'avancement sur le projet.
Après une formation aux différents algorithmes, et théorie dans la machine learning et deep learning, notamment Les réseaux de neurones de convolution, et le support vector machine, ou encore le word to vector, j'ai participer de manière actif au développement de la plateforme.

Mots-Clés : application mobile, machine learning, deep learning, SVM, CNN, WORD2VEC, Bayes, méthodologie Agile/Scrum.

Abstract

As part of my internship at SANADTECH, I had the chance to participate in the NACHARAT project. NACHARAT is a mobile application that allows these users to see what's new in video form easily. My host company is the company SANADTECH, led by Mr RHANIZAR. So I joined the development team composed of Mr CHATRIA and myself.
The project management method applied for the process is the agile method Scrum. This method favors short-term objectives and gives rise to interactions members of the team to discuss progress on the project.

After training in differents algorithms, and theory in machine learning and deep learning, I was able to participate active in the development of the platform.

Keywords: mobile application, machine learning, Agile , Scrum methodology.

Table des matières

Table des figures

Abbréviations

LSTM :	short term momory
cnn :	convolutional neural network
rnn :	recurrent neural network
svm :	Support vector machine
word2vect :	word to vector
nlp :	neuro-languistic processing

Introduction

Depuis le début de l'informatique, l'homme cherche à communiquer avec les machines. Si les nombreux langages de programmation permettent une forme d'échange entre l'homme et la machine, on aimerait que cette communication se fasse de façon plus naturelle. Pour que cela soit possible, il faut d'abord que la machine "comprenne" ce que l'utilisateur lui dit ensuite qu'elle soit capable de répondre d'une manière compréhensible par l'homme. La discipline derrière ce processus s'appelle le Natural Langage Processing (NLP) ou traitement Automatique du Langage Naturel (TALN) en français. Elle étudie la compréhension, la manipulation et la génération du langage naturel par les machines. Par langage naturel, on entend le langage utilisé par les humains dans leur communication de tous les jours par opposition aux langages artificiels comme les langages de programmation ou les notations mathématiques.
C'est dans ce contexte que s'inscrit le projet cadre en sein de l'entreprise SANADTECH, qui envisage d'établir un classificateur de texte à étiquettes multiples pour analyser les titres de son application mobile NACHARAT.
NACHARAT est une application mobile qui permet à ses utilisateurs d'être au courant des nouveautés en temps réel, à travers un ensemble de panoplies de news (hespress, hibapress, ChoufTV, ya bila...etc.).
Éventuellement, ce document décrit les étapes de la mise en œuvre de ce projet structuré en six chapitres principales, à savoir:

- La première Chapitre présente la partie management de projet en passant par une présentation rapide des acteurs du projet, méthode agile utilisée, et la planification du projet.
- Le deuxième chapitre contient la conception globale du projet, en la présente par un ensemble de diagrammes, et descriptions.
- Le troisième chapitre présente les données traitées, et la façon dont ils ont été manipulées.

– Le quatrième,cinquième et sixième chapitre exposent de façon détaillée les différentes étapes de la mise en place du classificateur, de l'analyse globale du projet, vers les différents algorithmes utilisés pour le classificateur.

Chapitre 1

Contexte générale du Projet

1.1 Présentation de l'organisme d'accueil

1.1.1 SanadTech

SanadTech, Société d'ingénierie et de conseil à taille humaine, spécialisée dans les solutions Cloud et mobiles. Elle est située à Hay Agdal, Rabat. dans des environnements très contraints (sécurité, performance, disponibilité, volumétrie).

SanadTech, Partenaire de Fujitsu RunMyProcess depuis 2013, qui à travers sa plate-forme RunMyProcess aide les organisations à transformer leur approches de travail grâce à des expériences user interface de bout en bout, et à l'automatisation couvrant les systèmes cloud, sur sites web ou sur mobiles.

Le personnel de SanadTech forme une petite équipe de développeurs et d'architectes qualifiés. Ils croient que chaque membre de l'équipe est important et a son mot à dire dans le développement de produits. Ils travaillent en étroite collaboration pour résoudre les problèmes difficiles.

Ils croient au partage des connaissances, et organisent une «pause-café» hebdomadaire pour partager de nouvelles pratiques exemplaires en matière de développement de logiciels. Ils assistent et parlent aux rencontres locales (groupes de développeurs Google, ...).

1.1.2 L'équipe

L'entreprise se compose du personnel suivant :

ABDELHAKIM RHANIZAR Fondateur , et Directeur technique .

MOHAMMED ERRAYSY Développeur front-end.

HAMZA HAJJI Développeur front-end.

1.2 Présentation du sujet

Il y a de plus en plus de demandes à avoir l'accès à des nouveautés en temps réel, ceci est dû à la transformation digitale de la presse, qui a commencé dans les années 1994. Au Maroc, la vague de la digitalisation de la presse n'a commencé qu'à vers les années 2005, et avec le temps, le nombre de presses a augmenté, et n'a pas cessé de progresser avec la demande, et les exigences du secteur.

Il est aujourd'hui difficile de trouver un site web d'information qui ne permette pas aux internautes de visionner des contenus audiovisuels, sous la forme de vidéos, de sons, de diaporamas sonores, de « reportages web » ou de « récits multimédia » mêlant texte, audio et vidéo. l'importance est devenue éminent, par l'intégration et l'inclusion de la vidéo dans les grands réseaux sociaux, ce qui permet à la presse de partager son contenu, et d'avoir plus de portée .

La vidéo est un composant vif, elle permet de bien diminuer la distance entre l'internaute et le sujet, et elle le permet plus, si elle est accompagné avec un titre significatif, qui pourra donner à l'internaute, un préjugement de ce que va contenir la vidéo, elle va l'aider à choisir s'il va voir la vidéo ou non. Mais parfois, l'internaute souhaite voir qu'une seule catégorie de vidéo de nouveautés, mais malheureusement les catégories n'accompagnent pas la vidéo tout le temps.

L'entreprise SANADTECH souhaite réaliser une application mobile, qui va permettre à ses utilisateurs d'être au courant des nouveautés en temps réel, à travers un ensemble de panoplies de news (hespress ,hibapress ,ChoufTV ,yabiladi , elbotola ,hibasport , alyaoum24 , medi1tv ...etc). L'application va offrir à ses utilisateurs la possibilité de visualiser les différents contenus, par catégories .

Afin de classifier les vidéo en des catégories, on va utiliser un module intelligent à l'aide des algorithmes de machine Learning.

L'application va contenir les caractéristiques suivantes :

- L'application va permettre de visualiser les dernières nouvelles sous format vidéo du Maroc provenant de dizaines de sources d'informations nationales et régionales diverses.
- L'application va permettre de sélectionner les sources d'actualités utilisées, pour que par la suite, l'utilisateur verra que les nouveautés de celui-ci dans la file d'actualité.
- L'application va permettre au client de faire un classement automatique des nouvelles par sujets, ou il peut faire un classement manuelle, selon ces besoins.
- L'utilisateur pourra recevoir des notifications lorsque des événements importants se produisent, dans son smartphone, pour lui inviter à la voir en exclusivité.
- L'utilisateur va avoir la possibilité d'ajuster les paramètres de notification et les thèmes à son gout.
- L'application aussi permettra à ces clients d'enregistrer les vidéos à regarder plus tard.

1.3 Technologies

La réalisation de ce projet de est divisée en trois parties :

1. Front-end
2. Back-end
3. Développement du modèle

et dans chaque partie, on va utiliser un ensemble d'outils, qu' on va détailler par la suite.

1.3.1 la partie front-end

Le développement front-end ou web frontal correspond aux production HTML, CSS et JavaScript d'une page internet ou d'une application qu'un utilisateur peut voir et avec lesquelles il peut interagir directement.

Javascript : JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs avec l'utilisation (par exemple)

de Node.js.

ReactXP : ReactXP est une bibliothèque pour le développement d'applications multiplate-formes utilisant React et React Native.

Redux : Redux est un conteneur d'état prévisible pour les applications JavaScript.Il aide à écrire des applications qui se comportent de manière cohérente, s'exécutent dans des environnements différents (client, serveur et natif) et sont faciles à tester.

Progressive Web Apps PWA : est une application web qui consiste en des pages ou des sites web, et qui peuvent apparaître à l'utilisateur de la même manière que les applications natives ou les applications mobiles.

1.3.2 Pour la partie back-end

Le développeur back-end travaille sur les éléments invisibles aux utilisateurs depuis le navigateur, mais indispensables au bon fonctionnement du site/application web. C'est ici qu'intervient la notion de site dynamique, qui est en constante évolution et mis à jour en temps réel avec des informations.

JEE : Positionnement de Java EE vs Java SE.Java Platform, Enterprise Edition, Java EE ou Jakarta EE (anciennement Java 2 Platform, Enterprise Edition, ou J2EE), est une spécification pour la plate-forme Java d'Oracle, destinée aux applications d'entreprise.

1.3.3 Pour la partie Développement du modèle

Python : Python est un langage de programmation objet, multi-paradigme et multiplate-formes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

Tensorflow : TensorFlow est un outil open source d'apprentissage automatique développé par Google. Le code source a été ouvert le 9 novembre 2015 par Google et publié sous licence Apache.Il est basé sur l'infrastructure DistBelief, initiée par Google en 2011, et est doté d'une interface Python.TensorFlow est l'un des outils les plus utilisés en IA dans le domaine de l'apprentissage machine.

Ml-Engine : Google Cloud Machine Learning est un service géré qui vous permet de créer facilement des modèles de machine learning adaptés à tous les types et volumes de données. Concevez votre modèle avec le framework ultra-performant TensorFlow, qui est utilisé dans de nombreux produits Google comme Google Photos ou Google Cloud Speech.

1.4 Méthodologie Adoptée

Dans le cadre de ce stage, on a choisi comme méthodologie de pilotage pour notre projet la méthodologie SCRUM, car elle a plusieurs atouts, et on peut les résumés suit :

- Plus de souplesse et de réactivité
- La grande capacité d'adaptation au changement grâce à des itérations courtes
- La chose la plus importante, c'est que SCRUM rassemble les deux côtés théorique et pratique et se rapproche beaucoup de la réalité.

La méthodologie SCRUM est composée de quatre phases (on parle aussi de réunion):

- Planification du Sprint
- Revue de Sprint
- Rétrospective de Sprint
- Mêlée quotidienne

1.4.1 Planification SCRUM

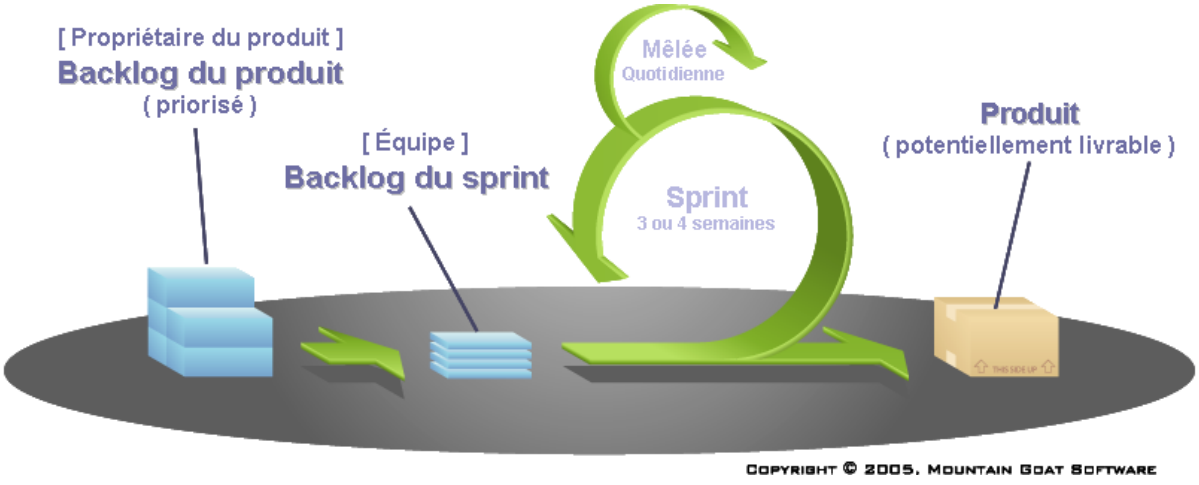

Fig. 1.1: Processus scrum

La planification du sprint correspond à lister les points prioritaires que l'équipe pense pouvoir réaliser au cours d'un sprint.

La revue du sprint a lieu en fin de sprint, l'équipe de développement présente les fonctionnalités terminées au cours du sprint et recueille les retours du représentant des utilisateurs finaux, c'est aussi à ce moment que la mise en place des prochains sprints peut être anticipée.

La Rétrospective de Sprint permet de faire un point sur le sprint en lui-même (productivité, efficacité, qualité...) afin de pouvoir s'améliorer pour les prochains sprints.

Enfin la mêlée quotidienne permet de faire un point sur les avancements de chacun, elle est courte et chacun réponds à trois questions principales : Qu'est-ce que j'ai terminé depuis la dernière mêlée ? Qu'est-ce que j'aurai terminé d'ici la prochaine mêlée ? Quels obstacles me retardent ?

1.4.2 L'équipe SCRUM

L'équipe a un rôle capital dans SCRUM : elle est constituée dans le but d'optimiser la flexibilité et la productivité; pour cela, elle s'organise elle-même et doit avoir toutes les compétences nécessaires au développement du produit. Elle est investie avec le pouvoir et l'autorité pour faire ce qu'elle a à faire.

Scrum definit les rôles suivants :

Le Product Owner (le propriétaire du produit) : c'est une personne qui porte la vision du produit à réaliser, généralement c'est un expert dans le domaine.

Le SCRUM Master (le directeur de produit) : c'est la personne qui doit assurer le bon déroulement des différents sprints du release, et qui doit impérativement maitriser SCRUM.

Le SCRUM Team (l'équipe de SCRUM) : constitué des personnes qui seront chargées d'implémenter les différents besoins du client. Bien évidemment, cette équipe sera constituée des développeurs, des testeurs, etc.

Dans le contexte de notre projet, Mr RHANIZAR serait le propriétaire du produit, et le directeur du produit puisqu'il satisfait les différents prés-requis de ces rôles cités précédemment et nous formons ELKAISSI SOUHAIL et CHATRIA ZAKARIA les deux membres de l'équipe SCRUM.

1.5 Langage de modélisation

Le langage de modélisation adopté durent ce stage sera le UML ,comme étant un outil comportant multiples points forts notamment sa standardisation ainsi que les divers diagrammes qu'il propose, et qui permettent la schématisation des systèmes complexes sous un format graphique et textuel simplifié et normalisé.
Généralement, le recours à la modélisation UML et plus particulièrement UML 2.0 procure de nombreux avantages qui agissent sur :

- La modularité.
- L'abstraction.
- La dissimulation.
- La structuration cohérente des fonctionnalités et des données.

1.6 TensorFlow : Librairie du machine learining et deep learning

TensorFlow est une bibliothèque de logiciels open source pour le calcul numérique utilisant des graphes de flux de données. Il a été initialement développé par l'équipe Google Brain au sein de l'organisation de recherche Machine Intelligence de Google pour l'apprentissage automatique et la recherche sur les réseaux neuronaux profonds, mais le système est assez général pour être applicable dans une grande variété d'autres domaines. Il a atteint la version 1.0 en février 2017 et a poursuivi son développement rapide, avec plus de 21 000 engagements à ce jour, dont un grand nombre de contributeurs externes.

TensorFlow est multi-plateforme. Il fonctionne sur presque tout : les processeurs graphiques et les processeurs normaux, y compris les plates-formes mobiles et embarquées, et même les unités de traitement tensoriel (TPU), qui sont des matériels spécialisés pour faire des calculs tensoriels. Ils ne sont pas encore largement disponibles, mais nous avons récemment lancé un programme alpha.

1.7 Planification prévisionnelle

Nom de la tâche	**Début de la tache**	**Date de la fin**	**Durée**
Début du stage cadre	23/04/2018	23/04/2018	1j
Phase de preparation	23/04/2018	01/05/2018	5j
Étude de la Solution actuelle	*22/04/2018*	*24/04/2018*	*2j*
Compréhension la totalité du projet	*25/04/2018*	*01/05/2018*	*3j*
Sprint 1: Réalisation d'un modèle de prédiction avec l'algorithme CNN	02/05/2018	23/05/2018	18j
Rédaction du Rapport	25/05/2018	25/05/2018	1j
Sprint 2: Réalisation d'un modèle de prédiction avec la méthode SVM	25/05/2018	04/06/2018	7j
Sprint 3: Amélioration du modèle CNN en ajoutant une couche de word2vec	05/06/2018	13/06/2018	8j
Sprint 4: Réalisation d'un modèle hybride qui regroupe CNN avec word2vec et le SVM	14/06/2018	06/07/2018	17j
Soutenance du Projet Cadre	22/06/2018	22/06/2018	1j
Sprint 5: Déploiement de la solution	09/07/2018	14/07/2018	5j

FIG. 1.2: planification prévisionnelle

Chapitre 2

Analyse et Conception

2.1 Le Backlog du Produit

Le backlog du produit est la liste des fonctionnalités attendues d'un produit. Plus exactement, au-delà de cet aspect fonctionnel, il contient tous les éléments qui vont nécessiter du travail pour l'équipe. Les éléments y sont classés par priorité ce qui permet de définir l'ordre de réalisation.

Le backlog est élaboré avant le lancement des sprints, dans la phase de préparation (ou sprint 0). Il est utilisé pour planifier la release, puis à chaque sprint, lors de la réunion de planification du sprint pour décider du sous-ensemble qui sera réalisé.

C'est donc un outil essentiel pour la planification. Mais il est aussi, par sa nature, un maillon de la gestion des exigences, puisqu'on y collecte ce que doit faire le produit.

A tout moment, le backlog est visible par tout le monde.

Story	**Priorité2**	**Effort ou chargen**
En tant qu'utilisateur, je souhaite pouvoir voir les news dans une fil d'actualité,et de sélectionner un contenu pour le visualiser.	M	22
En tant qu'utilisateur, je souhaite pouvoir ajouter un contenu au favoris pour que je puisse le visualiser plus tard.	C	11
En tant qu'utilisateur, je souhaite pouvoir visiter les différentes contenu par catégories,et de les sélectionner selon mon besoin.	M	60

FIG. 2.1: Le Backlog du Produit

2.1.1 La méthode de priorisation

Pour prioriser le besoin, on a utilisé la méthode **MoSCow** ; elle est une technique visant à prioriser des besoins ou des exigences en matière d'assistance à maîtrise d'ouvrage et de développement logiciel. L'objectif est que le maître d'œuvre et le maître d'ouvrage s'accordent sur l'importance des tâches à réaliser par rapport aux délais prévus.

Les lettres majuscules de l'acronyme MoSCoW signifient (en anglais):

M must have this, c'est-à-dire 'doit être fait' (vital).

S should have this if at all possible, c'est-à-dire devrait être fait dans la mesure du possible (essentiel).

C could have this if it does not affect anything else, pourrait être fait dans la mesure où cela n'a pas d'impact sur les autres tâches (confort).

W won't have this time but would like in the future, ne sera pas fait cette fois mais sera fait plus tard' (luxe, c'est votre zone d'optimisation budgétaire).

Chapitre 3

Sprint 0 : Preparation et étude de projet

3.1 Spécification

3.1.1 Le Backlog du Sprint

Story	**Priorité2**	**Effort ou chargen**
En tant qu'utilisateur, je souhaite pouvoir voir les news dans une fil d'actualité,et de sélectionner un contenu pour le visualiser.	M	22
En tant qu'utilisateur, je souhaite pouvoir ajouter un contenu au favoris pour que je puisse le visualiser plus tard.	C	11

FIG. 3.1: Le Backlog du Sprint 0

3.2 Conception

3.2.1 Diagramme de cas d'utilisation : services client

Comme on peut voir dans le digramme **FIG.3.2** , on a trois acteurs : le client, l'administrateur, et le modèle de machine learning de classification.

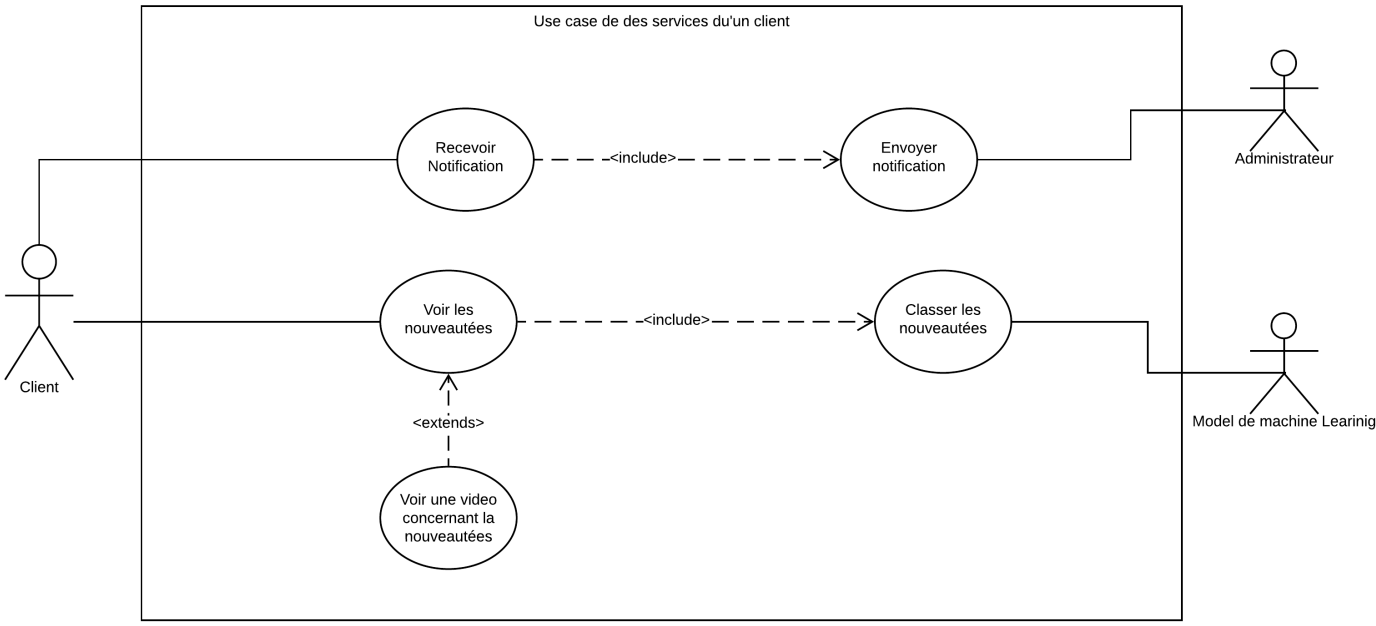

FIG. 3.2: Diagramme de cas d'utilisation

Pour que le client puisse visualiser les nouveautés , il faut que ces dernières soient classifier selon leurs catégories par le modèle, après cette manipulation, le client pourra naviguer dans la page d'accueil et voir les nouveautés .

Quand l'administrateur verra qu'une nouveauté est pertinente, il va lancer une notification, laquelle le client va recevoir, et pourra la visualiser en exclusivité.

3.2.2 Diagramme de classes

Comme on peut voir dans le digramme de la figure **FIG.3.3** , on a cinq classes : User, News, Catégorie; ApiAtacker et le modèle.

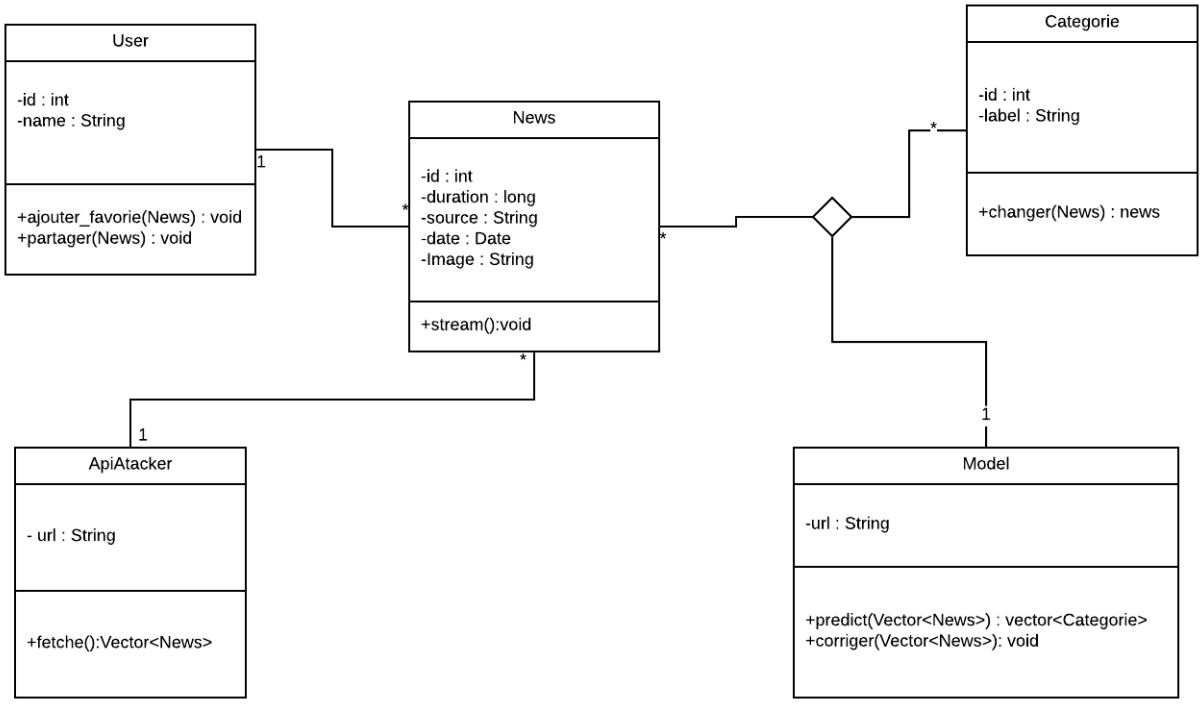

Fig. 3.3: Diagramme de classe

La classe **ApiAtacker**,c'est la classe responsable de l'acquisition des données depuis les différentes sources qu'on a, tels que : youtube, hesspress ...

La classe News représente une nouveauté dans notre système, et chaque nouveauté est lié par une et une seul catégorie.

Ici, la classe Model joue le rôle du classificateur, elle est celle qui transforme une nouveauté non étiqueté en une qui est qui est étiqueté.

Par ces différentes méthode, la classe User représente un client dans notre système, il décrit toutes ses actions possible.

3.2.3 Diagramme de séquence

Comme vous pouvez constater dans le diagramme de la figure **FIG.3.4**, nous avons quatre ligne de vie dont le Client, l'Application front-end, l'Application back-end,et le Model tensorFlow.

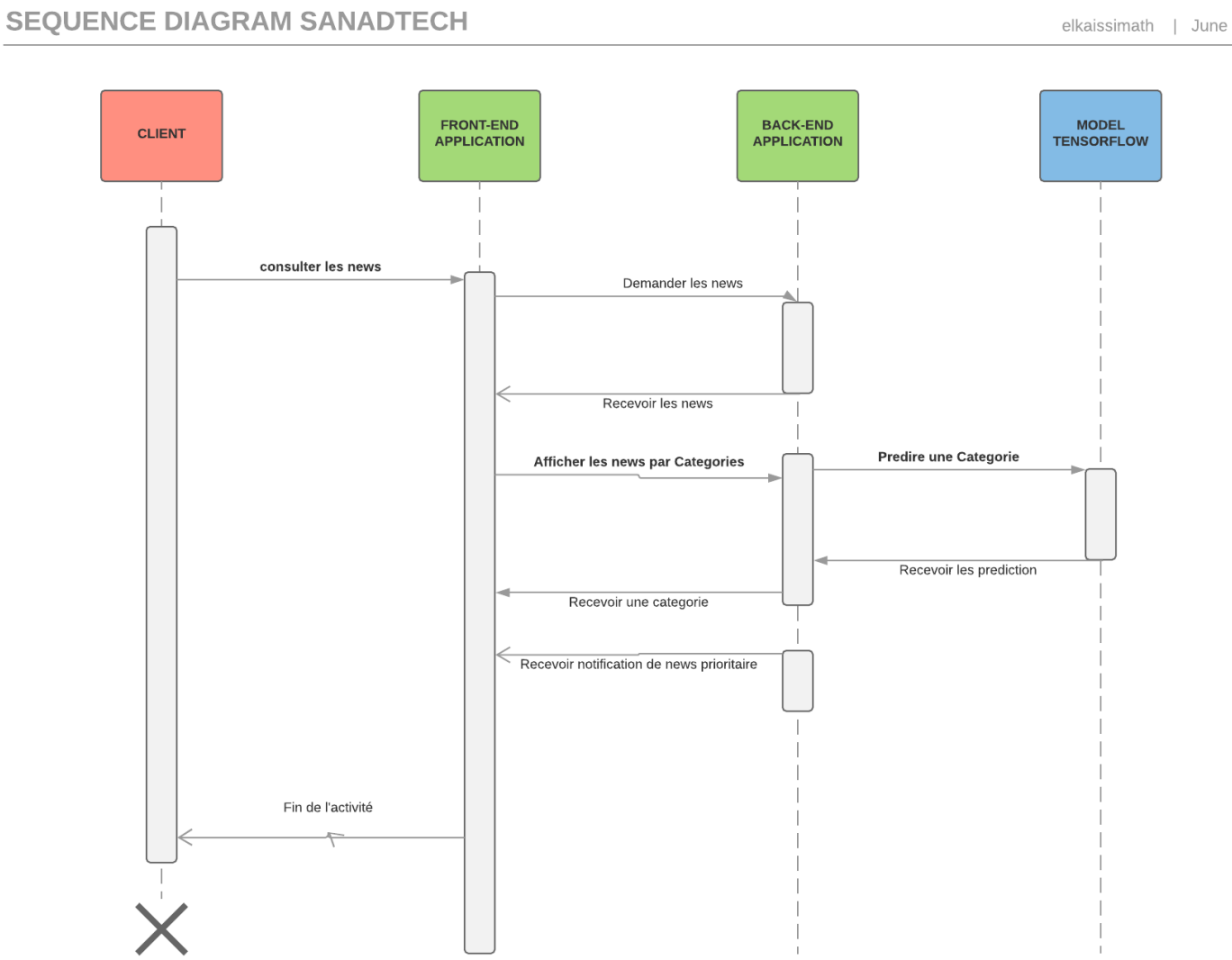

FIG. 3.4: Diagramme de séquence

L'activité de l'application au point de vue client commence, qu'on un utilisateur installe l'app dans son téléphone, il pourra ensuite recevoir les notifications des nouveautés les plus important.

Quand un utilisateur consulte les nouveautés , le front-end envoi une requête au back-end, et ce dernier lui envoi tout les news par ordre de date.

Si une news viens juste d'être intercepter par le back-end, elle envoi alors, une requête au modèle Tensorflow pour savoir de quelle catégorie cette news appartiens. Et puis après, le back-end l'enregistre avec ça nouvelle classe ou catégorie.

Si une nouveauté est jugé comme importante, elle sera envoyer vers toutes les utilisateurs sous la forme d'une notifications, depuis le back-end.

3.2.4 Diagramme de déploiement

Le diagramme de la figure **FIG.3.5** contient 4 nœuds principales, on site : front-end server, back-end server, database server , Model tensorflow server.

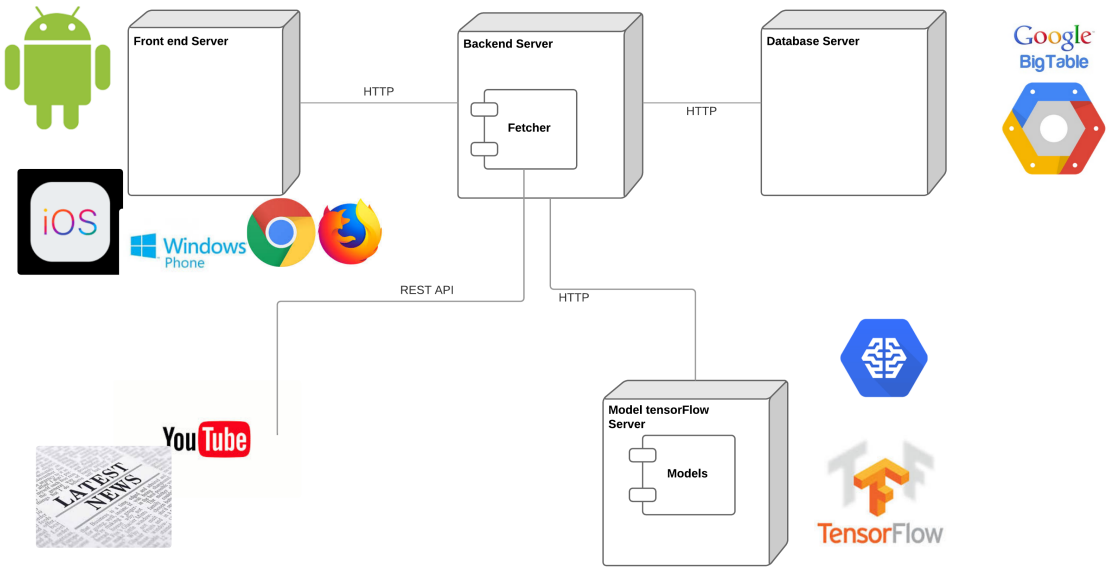

FIG. 3.5: Diagramme de déploiement

Front-end server , c'est ou l'application finale est déployé, d'ici les utilisateurs peuvent la télécharger, depuis les différentes Stores, comme google play, apple store ...,ou la consulter dans le cas d'un navigateur.

Back-end server, c'est ou le apiAtacker est situé , l'application envoi des requêtes vers ce nœud pour avoir les news.

Database server, ici on stocke nos données , on utilise la base de donnée de google Big Table.

Model Tensorflow Server, c'est ou notre model de prédiction est stocké , le modèle est écrit par tensorflow et déployer dans le service google ML-engine.

3.3 NLP : Natural Language Processing

Le modèle de classification qu'on va réaliser entre dans la catégorie de la NLP, avant de le définir et l'expliquer, il faut d'abord rappeler qu'est ce que le machine learning et le deep learning.

3.3.1 Machine learning

L'apprentissage automatique (en anglais machine learning, littéralement « l'apprentissage machine ») ou apprentissage statistique, champ d'étude de l'intelligence artificielle, concerne la conception, l'analyse, le développement et l'implémentation de méthodes permettant à une machine (au sens large) d'évoluer par un processus systématique, et ainsi de remplir des tâches difficiles ou problématiques par des moyens algorithmiques plus classiques.

L'analyse peut concerner des graphes, arbres, ou courbes (par exemple, la courbe d'évolution temporelle d'une mesure ; on parle alors de données continues, par opposition aux données discrètes associées à des attributs-valeurs classiques) au même titre que de simples nombres. Voir l'article Inférence bayésienne.

Les algorithmes utilisés permettent, dans une certaine mesure, à un système piloté par ordinateur (un robot éventuellement), ou assisté par ordinateur, d'adapter ses analyses et ses comportements en réponse, en se fondant sur l'analyse de données empiriques provenant d'une base de données ou de capteurs.
La difficulté réside dans le fait que l'ensemble de tous les comportements possibles compte tenu de toutes les entrées possibles devient rapidement trop complexe à décrire (on parle d'explosion combinatoire). On confie donc à des programmes le soin d'ajuster un modèle pour simplifier cette complexité et de l'utiliser de manière opérationnelle. Idéalement, l'apprentissage visera à être non supervisé, c'est-à-dire que la nature des données d'entrainement n'est pas connue.

Ces programmes, selon leur degré de perfectionnement, intègrent éventuellement des capacités de traitement probabiliste des données, d'analyse de données issues de capteurs, de reconnaissance (reconnaissance vocale, reconnaissance de forme, d'écriture...), de data-mining, d'informatique théorique...

3.3.2 Deep Learning

L'apprentissage profond1 (en anglais deep learning, deep structured learning, hierarchical learning) est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires[réf. souhaitée]. Ces techniques ont permis des progrès importants et rapides dans les domaines de l'analyse du signal sonore ou visuel et notamment de la reconnaissance faciale, de la reconnaissance vocale, de la vision par ordinateur, du traitement automatisé du langage. Dans les années 2000, ces progrès ont suscité des investissements privés, universitaires et publics importants, notamment de la part des GAFA (Google, Apple, Facebook, Amazon).

Les recherches dans ce domaine s'efforcent de construire de meilleures représentations du réel et de créer des modèles capables d'apprendre ces représentations à partir de données non labellisées à grande échelle. Certaines de ces représentations s'inspirent des dernières avancées en neuroscience. Il s'agit, grosso modo, d'interprétations du traitement de l'information et des modèles de communication du système nerveux, à l'image de la façon dont le système nerveux établit des connexions en fonction des messages reçus, de la réponse neuronale et du poids des connexions entre les neurones du cerveau.
Les différentes architectures de « deep learning » telles que les « deep neural networks », les « convolutional deep neural networks », et les « deep belief network » ont plusieurs champs d'application :

- la vision par ordinateur (reconnaissance de formes) ;
- la reconnaissance automatique de la parole ;
- le traitement automatique du langage naturel ;
- la reconnaissance audio et la bio-informatique.

Dans ces deux derniers domaines, notamment, elles ont obtenu des résultats très prometteurs.

3.3.3 NLP

Le traitement automatique du langage naturel (abr. TALN), ou traitement automatique de la langue naturelle1, est un domaine multidisciplinaire impliquant la linguistique, l'informatique et l'intelligence artificielle. Il vise à créer des outils de traitement de la langue naturelle pour

diverses applications. Il ne doit pas être confondu avec la linguistique informatique, qui vise à comprendre les langues au moyen d'outils informatiques.

Le TALN est sorti des laboratoires de recherche pour être progressivement mis en œuvre dans des applications informatiques nécessitant l'intégration du langage humain à la machine2. Aussi le TALN est-il parfois appelé ingénierie linguistique3. En France, le traitement automatique de la langue naturelle a sa revue, ATALA, publiée par l'Association pour le traitement automatique des langues.

3.4 Préparation des données

Dans un projet de machine learning, il est important de bien encadrer les données, parce que en sens large, il constitue l'enjeu majeur. Pour ce fait, on a suivi ces principaux étapes :

3.4.1 Identification des sources de données

Les données qu'on traite sont des données de type chaine de caractère qui comporte deux champs, le première champs et celui de la catégorie ou la classe, et le deuxième champs et celui de la description ou le titre de la vidéo.

Category	Descript
sport	انتظارات من أسود الأطلس
politics	الجزائر.. وضعية بوتفليقة تطغى على افتتاح دورة البرلمان
society	روبورتاج.. أعباء الدخول المدرسي ثقل كاهل أولياء التلاميذ
society	توقعات المغاربة لمباراة الإياب بين المنخب المغربي والمنتخب المالي
society	أطباء عاطلون يحتجون بالرباط
sport	البطل محمد الصنهايي يتألق في الكيك بوكسينغ
society	خنيفرة - مشروع التكتل من اجل العدالة للنساء
society	الواجهة: العثور على جثة سبعيني ميت في بيته
society	جهة الدار البيضاء-سطات ومركز تحاقن الدم يدقان ناقوس الخطر: تبرعوا بالدم لإنقاذ إخوانكم
business	ملف.. الشراكة المغرب الإسبانية في المجال التجاري
various	سوق الصباط بالرباط
sport	أصغر بطلة فول كونتاكت
various	تافوغالت .. أريج الفجاج
culture	دردشة مع بدر سلطان
various	أبراج | أشنو قال زهرك اليوم : 03 يوليوز 2017 | شوف تيفي
politics	زوجة المهدوي تناقش حراك الريف
various	أحوال الطقس : 03 يوليوز 2017
politics	غضبة ملكية تطيح بمسؤولي الدرك بتطوان
politics	المغرب يوجه صفعة قوية للبوليساريو والجزائر بأثيوبيا

FIG. 3.6: Exemple de données

Le format des données actuellement, avec lequelle on va faire l'entrainement et les testes, sont en format **CSV**, l'avantage de ce format c'est qu'il est facilement manipulable par le langage de programmation python même si la taille de données peut être grande.

On récupère ses données là depuis plusieurs source, et après on extrait les données juger utiles pour les prédiction, dans ce cas les description des vidéo.

3.4.2 Définition du lieu d'entreposage

Les données transformé doivent etre stockés, pour le fait de ne pas les predire une deuxieme fois, on utilise dans notre projet la base données google, Big Table, il est un système de gestion de base de données compressées, haute performance, et propriétaire, C'est une base de données orientée colonnes, dont se sont inspirés plusieurs projets libres, comme HBase, Cassandra ou Hypertable.

FIG. 3.7: google cloud big table

Google ne distribue pas sa base de données mais propose une utilisation publique de BigTable via sa plateforme d'application Google App Engine.

3.4.3 Nettoyage des données

Les données utilisées proviennent de diverses sources et processus et peuvent contenir des irrégularités ou des données corrompues compromettant la qualité de l'ensemble de données. Les problèmes typiques de qualité des données qui se posent sont les suivants:

Incomplet : Les données manquent d'attributs ou contiennent des valeurs manquantes.

Noisy : les données contiennent des enregistrements erronés ou des valeurs aberrantes.

Inconsistant : les données contiennent des enregistrements ou des divergences conflictuels.

Les données de qualité sont une condition préalable à la qualité des modèles prédictifs. Pour éviter les «déchets» et améliorer la qualité des données et donc la performance du modèle, nous avons trouvé qu'il est impératif de réaliser un écran d'intégrité des données afin de repérer rapidement les problèmes de données et de décider des étapes de traitement et de nettoyage correspondantes.

3.4.4 Features generation

C'est un processus consistant à prendre des données brutes non structurées et à définir des caractéristiques (c'est-à-dire des variables) pour une utilisation potentielle dans l'analyse statistique.
Par exemple, dans le cas de l'exploration de texte *(notre cas)*, on a commencé avec un journal brut de milliers de descriptions *(celles des vidéos)* et générer des fonctionnalités en supprimant des mots de faible valeur. des blocs de mots *(n-grammes)* ou en appliquant d'autres règles.

pour notre cas, on prend le corpus de données et on fait une relation symétrique entre les données de types chaine de caractère,et des entiers unique, qu'on le store par la suite dans un dictionnaire de données.

3.4.5 Feature extration

Une fois les fonctionnalités générées, il est souvent nécessaire de tester les transformations des entités d'origine et de sélectionner un sous-ensemble de ce groupe de caractéristiques potentielles originales et dérivées à utiliser dans notre modèle (extraction et sélection de caractéristiques).

Tester des valeurs dérivées est une étape courante car les données peuvent contenir des informations importantes qui ont un modèle ou une relation non linéaire avec notre résultat, ainsi l'importance de l'élément de données peut seulement apparaître dans son état transformé (par exemple des dérivés d'ordre supérieur).

L'utilisation d'un trop grand nombre de caractéristiques peut entraîner une colinéarité multiple ou confondre des modèles statistiques, alors qu'extraire le nombre minimum de caractéristiques correspondant au but de votre analyse suit le principe de la parcimonie.

On a divisé les données generées en deux parties, les données de d'entrainement qu'on lui a attribué 60% des données du corpus, et les données de test qui formes un pourcentage de 40%.

Chapitre 4

Sprint 1 : Réalisation d'un modèle de prédiction avec l'algorithme CNN

4.1 Spécification

4.1.1 Le Backlog du Sprint

Story	**Priorité2**	**Effort ou chargen**
En tant qu'utilisateur, je souhaite pouvoir visiter les différentes contenu par catégories,et de les sélectionner selon mon besoin.	M	22

FIG. 4.1: Le Backlog du Sprint 1

4.2 Conception

4.2.1 Qu'est ce qu'un CNN/RNN ?

Réseau de neurones convolutifs

En apprentissage automatique, un réseau de neurones convolutifs ou réseau de neurones à convolution (en anglais CNN ou ConvNet pour Convolutional Neural Networks) est un type de réseau

de neurones artificiels acycliques (feed-forward), dans lequel le motif de connexion entre les neurones est inspiré par le cortex visuel des animaux. Les neurones de cette région du cerveau sont arrangés de sorte qu'ils correspondent à des régions qui se chevauchent lors du pavage du champ visuelle.

Leur fonctionnement est inspiré par les processus biologiques, ils consistent en un empilage multicouche de perceptrons, dont le but est de pré-traiter de petites quantités d'informations. Les réseaux neuronaux convolutifs ont de larges applications dans la reconnaissance d'image et vidéo, les systèmes de recommandation et le traitement du langage naturel.

Réseau de neurones récurrents

Un réseau de neurones récurrents est un réseau de neurones artificiels présentant des connexions récurrentes. Un réseau de neurones récurrents est constitué d'unités (neurones) inter-connectés interagissant non-linéairement et pour lequel il existe au moins un cycle dans la structure. Les unités sont reliées par des arcs (synapses) qui possèdent un poids. La sortie d'un neurone est une combinaison non linéaire de ses entrées.

Les réseaux de neurones récurrents sont adaptés pour des données d'entrée de taille variable. Ils conviennent en particulier pour l'analyse de séries temporelles. Ils sont utilisés en reconnaissance automatique de la parole ou de l'écriture manuscrite - plus en général en reconnaissance de formes - ou encore en traduction automatique. « Dépliés », ils sont comparables à des réseaux de neurones classiques avec des contraintes d'égalité entre les poids du réseau (voir schéma ci-dessous).

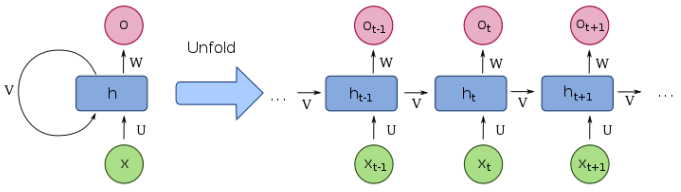

FIG. 4.2: Schéma d'un réseau de neurones récurrents à une unité reliant l'entrée et la sortie du réseau. A droite la version « dépliée » de la structure.

4.2.2 L'architecture de notre réseaux neuronal

Le réseaux neuronal qu'on a adopté est comme suite :

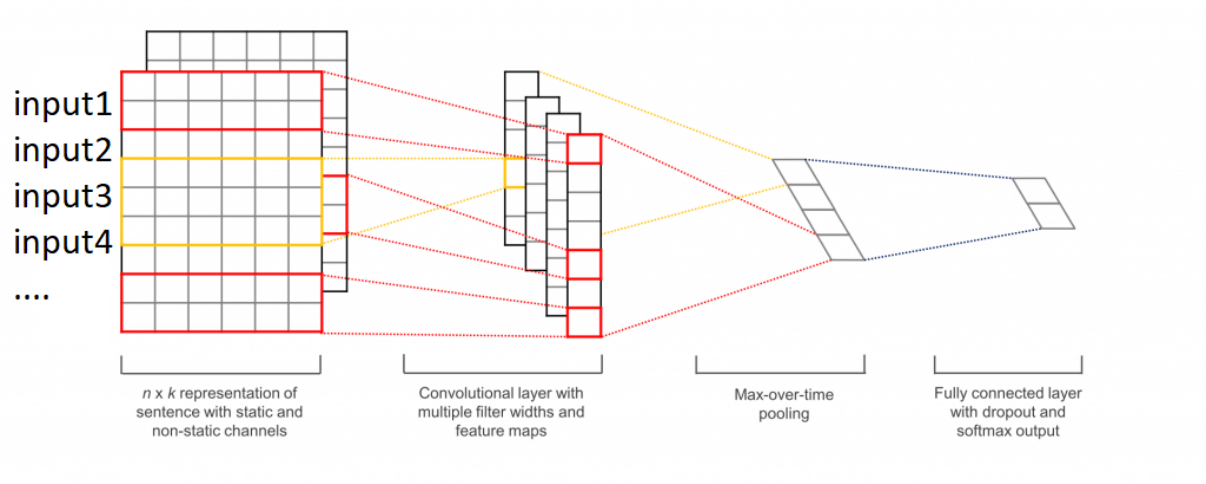

FIG. 4.3: La structure du reseaux neuronal ,réaliser par cnn.

Les premières couches incorpore des mots dans des vecteurs de basse dimension. La couche suivante effectue des convolutions sur les vecteurs de mots incorporés, en utilisant des tailles de filtres multiples.

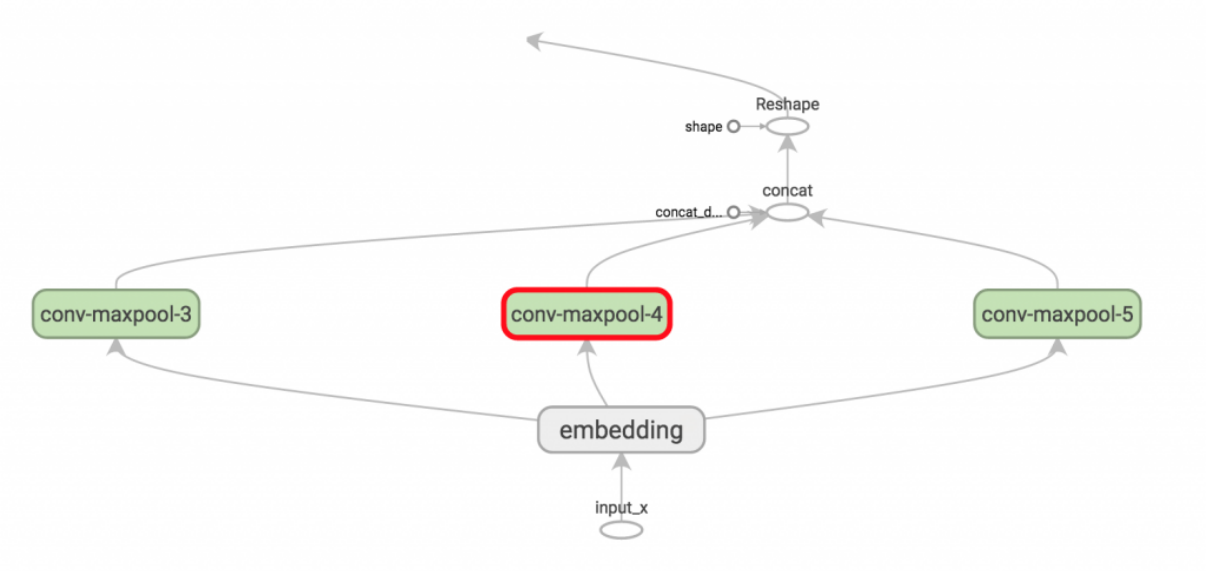

FIG. 4.4: La structure du réseaux neuronal, structure des couches de convolutions.

Par exemple, glisser sur 3, 4 ou 5 mots à la fois. Ensuite, nous max-poolons le résultat de la couche convolutionnelle dans un long vecteur de caractéristiques,et on ajoute une régularisation de décrochage, et en fin on classe le résultat en utilisant une couche softmax.

Puisque on ne peut pas traiter les chaînes de caractères dans un réseaux neuronal, on doit transformer ces données là, en des données numérique, c'est pour cela, j'ai utilisé une transformation classique « Embeding» qui mappe chaque mot qui existe dans mon dictionnaire de mot ,en des numéro significatif.

Pour bien comprendre le déroulement ,et le processus de fonctionnement de ce réseaux neuronal , voici une figure descriptive :

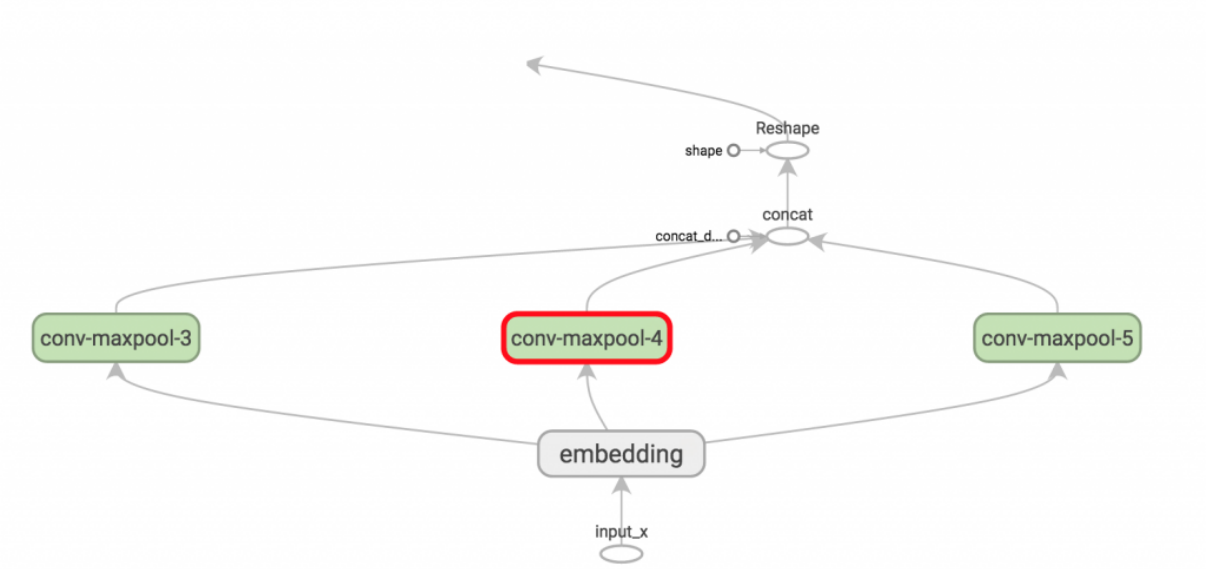

FIG. 4.5: Architecture des couches du réseaux neuronal.

Ici, **W** est notre matrice de filtre et h est le résultat de l'application de la non-linéarité à la sortie de la convolution. Chaque filtre glisse sur l'ensemble de l'incorporation, mais varie en nombre de mots. Le remplissage *VALIDE* signifie que nous glissons le filtre sur notre phrase sans capitonner les bords, en effectuant une convolution étroite qui nous donne une sortie de forme [1,$sequenceLengthFilterSize + 1$,1,1].

L'exécution de **max-pooling** sur la sortie d'une taille de filtre spécifique nous laisse avec un tenseur de forme [$batchSize$,1,1,$numFilters$]. C'est essentiellement un vecteur de caractéristiques,

où la dernière dimension correspond à nos caractéristiques.

Une fois que nous avons tous les tenseurs de sortie regroupés de chaque taille de filtre, nous les combinons en un long vecteur de forme de forme [$tailleBatch$,$numFiltersTotal$].

L'utilisation de -1 dans tf.reshape indique à TensorFlow d'aplatir la cote lorsque cela est possible.

Pour régulariser notre CNN, j'ai utilisé une couche **Dropout** ,l'idée derrière est simple. Une couche de décrochement stochastique **désactive** une fraction de ses neurones. Cela empêche les neurones de co-adapter, et les oblige à apprendre des fonctionnalités utiles individuellement.

La fraction de neurones que nous gardons activée est définie sur notre réseau. Nous avons réglé ceci à quelque chose comme *0.5* pendant l'entraînement, et à *1* (désactiver la couche) pendant l'évaluation.

En utilisant le vecteur de caractéristiques de **max-pooling** (avec **dropout** appliqué), nous pouvons générer des prédictions en faisant une multiplication matricielle et en choisissant la classe ayant le score le plus élevé. Nous pourrions également appliquer une fonction **softmax** pour convertir les scores bruts en probabilités normalisées, mais cela ne changerait pas nos prédictions finales.

En utilisant nos *scores*, nous pouvons définir la fonction de perte. La perte est une mesure de l'erreur que notre réseau fait, et notre objectif est de le minimiser. La fonction de perte standard pour la catégorisation pose des problèmes de perte d'*entropie* croisée.

Enfin le graphe final résultant ressemble à ça :

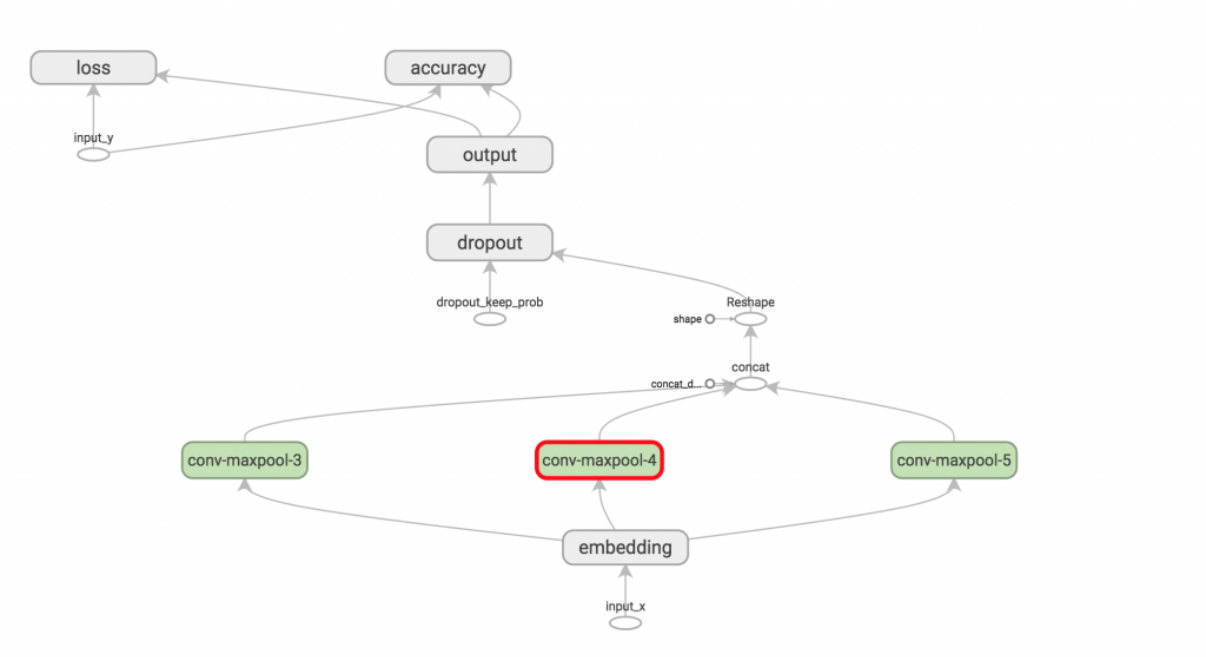

FIG. 4.6: Graphe qui représente les étapes du traitement du modèle.

L'état des pertes sur les variables de décision :

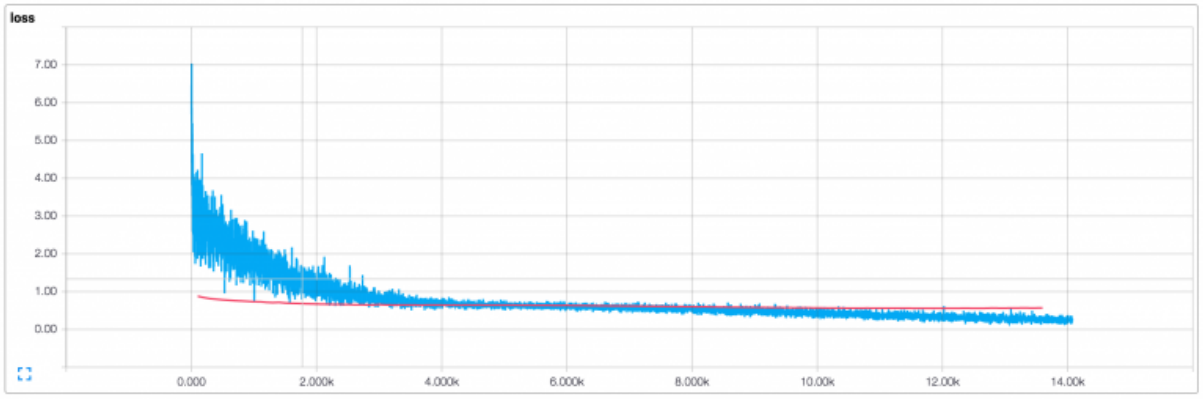

FIG. 4.7: Graphe qui représente les données perdues, du reseaux CNN/RNN.

l'état des précision qui est égale à 0,76:

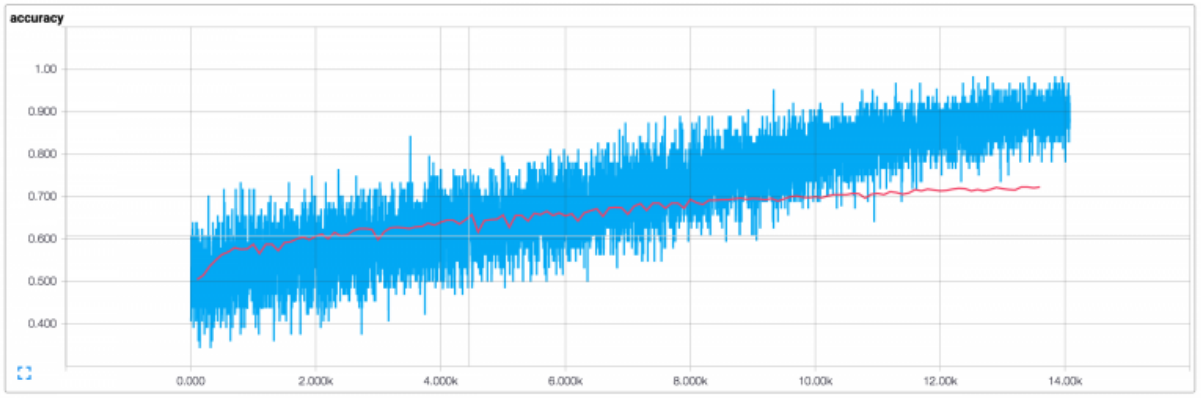

FIG. 4.8: Graphe qui représente la précision, du reseaux CNN/RNN

Chapitre 5

Sprint 2 : Réalisation d'un modèle de prédiction avec l'algorithme SVM

5.1 Spécification

5.1.1 Le Backlog du Sprint

Story	**Priorité2**	**Effort ou chargen**
En tant qu'utilisateur, je souhaite pouvoir visiter les différentes contenu par catégories,et de les sélectionner selon mon besoin.	M	22

FIG. 5.1: Le Backlog du Sprint 2

5.2 Conception

5.2.1 Qu'est ce que le SVM?

Les machines à vecteurs de support ou séparateurs à vaste marge (en anglais support vector machine, SVM) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination 1 et de régression. Les SVM sont une généralisation des classifieurs linéaires.

Les séparateurs à vaste marge ont été développés dans les années 1990 à partir des considérations théoriques de Vladimir Vapnik sur le développement d'une théorie statistique de l'apprentissage : la théorie de Vapnik-Chervonenkis. Ils ont rapidement été adoptés pour leur capacité à travailler avec des données de grandes dimensions, le faible nombre d'**hyperparamètres**, leurs garanties théoriques, et leurs bons résultats en pratique.

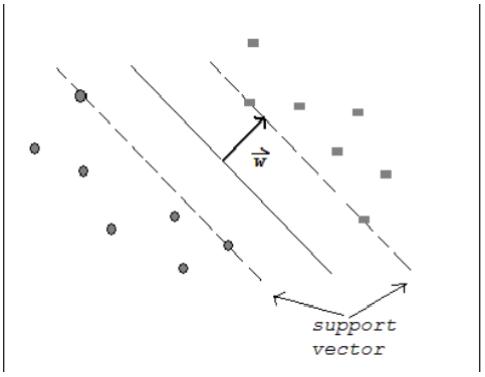

FIG. 5.2: Exemple d'un hyperplan pattern de SVM

Les SVM ont été appliqués à de très nombreux domaines (bio-informatique, recherche d'information, vision par ordinateur, finance...). Selon les données, la performance des machines à vecteurs de support est de même ordre, ou même supérieure, à celle d'un réseau de neurones ou d'un modèle de mélanges gaussiens.

Les machines à vecteurs de support sont basées sur le principe de la structure de minimisation de risque, dans la théorie de l'apprentissage aléatoire.L'idée de la structure de minimisation de risque est de trouver une hypothèse $\boldsymbol{h}$ avec laquelle on peut garantir, d'avoir le plus petit cout d'erreur.
La vrai erreur de $\boldsymbol{h}$, c'est la probabilité que $\boldsymbol{h}$ va faire une erreur en une données non traiter,et choisit aléatoirement. Un extremum haut, peut être utiliser, pour connecter, la vrai erreur de $\boldsymbol{h}$, avec l'erreur de $\boldsymbol{h}$ et avec les données de l'entrainement, est une complexité $\boldsymbol{H}$ (mesurer par VC-Dimension), l'hypothèse $\boldsymbol{h}$, laquelle approximativement, minimise l'extremum pour la vrai valeur de $\boldsymbol{h}$ en controllant the VC-Dimension de $\boldsymbol{H}$.

Les SVMs, sont des apprentis universelles. Dans leur frome de base, ils apprennent par des fonctions seuil linéaire. Sauf que par une utilisation des fonction de kernel à la place des variables

unitaires, ils peuvent être utiliser pour un apprentissage polynomiale, et produire de classificateurs polynomiale.

Autre chose qui est magnifique chez le SVM,c'est que son pouvoir d'apprentissage est indépendante du dimension de l'espace du features, il mesure la complexité de l'hypothèse basé sur la marge qui sépare les données, non le nombre des features. Ceci signifie qu'on peut généraliser, si notre données est séparable avec une grande fonction de marge, même en présence de plusieurs features.

La marge suggère aussi d'avoir une heuristique pour sélectionner des valeurs exactes de traitement pour l'apprenti (comme le kernel avec le réseaux RBF). le meilleurs paramétrage, est celui qui produit l'hypothèse avec le plus petit VC-Dimension. Ceci permet d'automatiser les paramètres sans une perte de la cross-valisation.

5.2.2 implémentation et optimisation

On utilise le SVM dans les problèmes de classification binaires, or dans notre cas on 7 catégories. Pour ce fait , il y a deux approche pour remédier à cette problématique :

Un contre tout le monde

Dans cette approche, on construit k classificateurs binaires séparés pour la classification k-class. Le classificateur binaire m-th est formé en utilisant les données de la m-ième classe comme exemples positifs et les classes k-1 restantes comme négatives exemples.

Pendant le test, l'étiquette de classe est déterminée par le classificateur binaire donne la valeur de sortie maximale. Un problème majeur de l'approche un versus repos est l'ensemble d'entraînement déséquilibré. Supposons que toutes les classes ont une taille égale d'entraînement exemples, le ratio d'exemples positifs à négatifs dans chaque classificateur est $1/(1-k)$. Dans ce cas, la symétrie du problème original est perdue.

Un contre Un

Une autre approche classique pour la classification multi-classe est le Un contre Un (1V1) ou

une décomposition par paires . Il évalue tous les classificateurs par paires possibles et ainsi induit $k * (k-1)/2$ classificateurs binaires individuels. Appliquer chaque classificateur à un test exemple donnerait un vote à la classe gagnante.

Un exemple de test est libellé pour la classe avec le plus de votes. La taille des classificateurs créés par le un contre un approche est beaucoup plus grande que celle de l'approche un versus repos.

Cependant, la taille de QP dans chaque classificateur est plus petit, ce qui permet de s'entraîner rapidement. De plus, par rapport à l'approche un contre le repos, la méthode un-contre-un est plus symétrique. Platt et al. a amélioré l'approche un-contre-un et proposé une méthode appelée SVM Directed Acyclic Graph SVM (DAGSVM) qui forme un arbre structure pour faciliter la phase de test. En conséquence, il ne faut que k - 1 individu évaluations pour décider de l'étiquette d'un exemple de test.

J'ai essayé les deux méthode, avec l'utilisation du kernel gaussiens pour les deux cas, et aussi en variant la valeur de la variables gamma, et j'ai conclue après les testes sur les deux modèles que le (Un contre Un) et mieux adapté pour notre cas.
Puisque qu'on a 7 catégorie, le nombre de modèles utiliser est 21, chaque modèle sert à classifier la description entre deux classes, et c'est sur qu'il va y avoir un résultat .

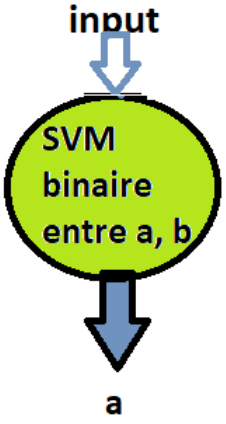

FIG. 5.3: Représentation d'un modèle binaire SVM

L'algorithme utilisé consiste à faire les prédiction dans chaque modèle qu'on a , et puis après en regroupe le résultat finale dans un vecteur, qui contient lui aussi les valeurs d'apparition d'une classe dans une prédiction, on le résultat final, est la classe qui a apparu le plus, dans le vecteur .

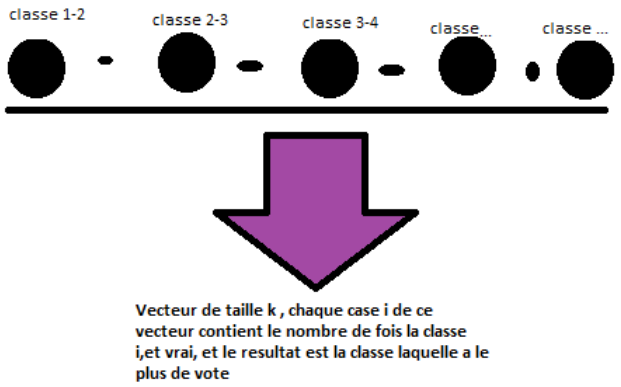

FIG. 5.4: Représentation globale du modèle

L'état des pertes sur les variables de décision :

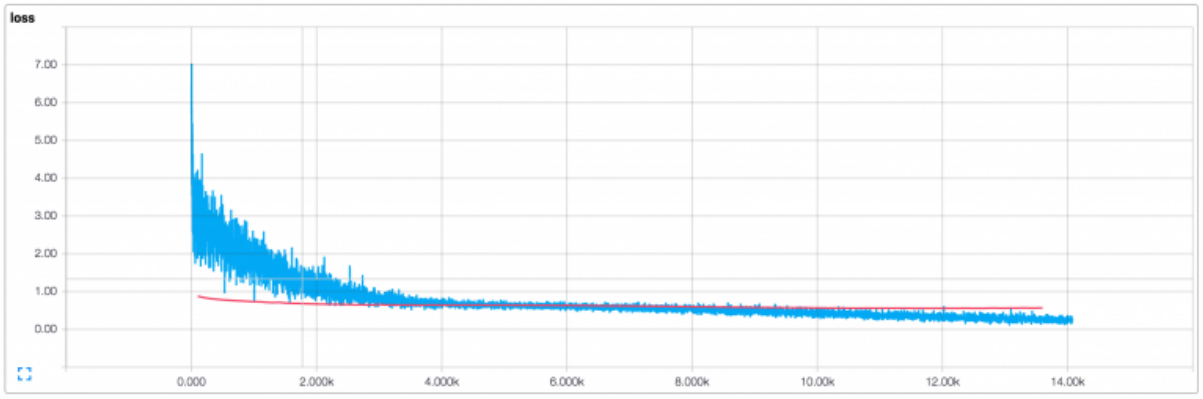

FIG. 5.5: Graphe qui représente les données perdues, du multiClass-SVM.

l'état des précision qui est égale à 0,65:

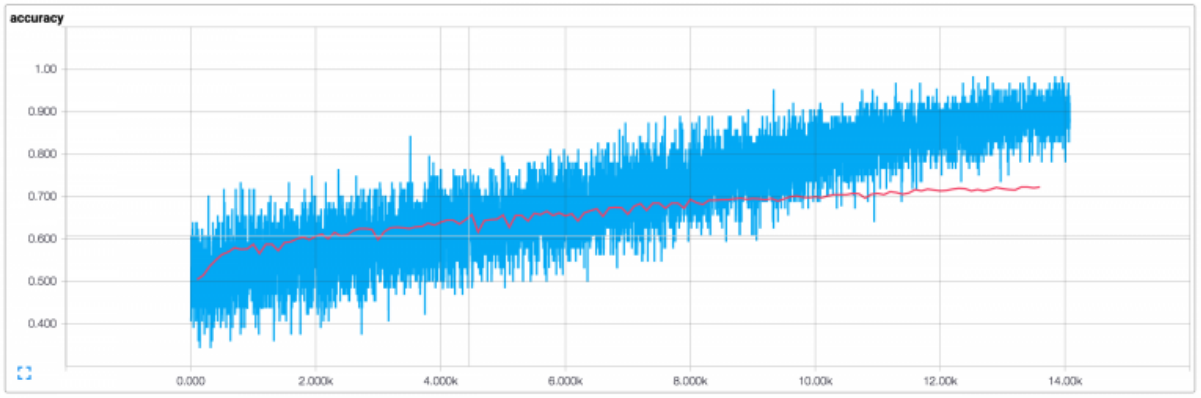

FIG. 5.6: Graphe qui représente la précision, du multiClass-SVM.

Chapitre 6

Sprint 3 : Amélioration du CNN avec word2vec

6.1 Spécification

6.1.1 Le Backlog du Sprint

Story	**Priorité2**	**Effort ou chargen**
En tant qu'utilisateur, je souhaite pouvoir visiter les différentes contenu par catégories,et de les sélectionner selon mon besoin.	M	22

Fig. 6.1: Le Backlog du Sprint 3

6.2 Conception

6.2.1 Qu'est ce qu'un word2vec ?

Les systèmes de traitement de l'image et du son fonctionnent avec des jeux de données riches et de grande dimension codés comme vecteurs des intensités de pixels brutes individuelles pour les données d'image, ou par exemple des coefficients de densité spectrale de puissance pour les données audio. Pour les tâches telles que la reconnaissance d'objet ou de reconnaissance vocale,

nous savons que toutes les informations nécessaires à l'exécution de la tâche sont codées dans les données (car les humains peuvent effectuer ces tâches à partir des données brutes).

Cependant, les systèmes de traitement du langage naturel traitent traditionnellement les mots comme des symboles atomiques discrets, et par conséquent, le terme 'chat' peut être représenté par Id537 'chien' Id143. Ces codages sont arbitraires et ne fournissent aucune information utile au système concernant les relations qui peuvent exister entre les symboles individuels. Cela signifie que le modèle peut tirer parti très peu de ce qu'il a appris sur les 'chats' lorsqu'il traite des données sur les 'chiens' (tels qu'ils sont à la fois des animaux, des quadrupèdes, des animaux domestiques, etc...). Le fait de représenter les mots comme des identificateurs uniques et discrets entraîne en outre un manque de données, et signifie généralement que nous aurons besoin de plus de données pour réussir la formation de modèles statistiques. L'utilisation de représentations vectorielles peut surmonter certains de ces obstacles.

Les modèles d'espace vectoriel (VSM) représentent (incorporent) des mots dans un espace vectoriel continu où des mots sémantiquement similaires sont mappés à des points voisins ('sont imbriqués les uns près des autres'). Les VSM ont une histoire longue et riche en PNL, mais toutes les méthodes dépendent d'une manière ou d'une autre de l' hypothèse distributionnelle , qui stipule que les mots qui apparaissent dans les mêmes contextes partagent une signification sémantique. Les différentes approches qui tirent parti de ce principe peuvent être divisées en deux catégories: les méthodes fondées sur le nombre (par exemple, l' analyse sémantique latente) et les méthodes prédictives (par exemple, les modèles de langage probabiliste neuronal).

Word2vec est un modèle prédictif particulièrement efficace sur le plan informatique pour l'apprentissage des plongées de mots à partir de texte brut. Il se présente sous deux formes: le modèle du sac continu de mots (CBOW) et le modèle de Skip-Gram (sections 3.1 et 3.2 de Mikolov et al.). Algorithmiquement, ces modèles sont similaires, sauf que CBOW prédit les mots cibles (par exemple 'mat') à partir des mots de contexte source ('le chat s'assoit sur le'), tandis que le skip-gram fait l'inverse et prédit les mots de contexte source de la cible mots. Cette inversion peut sembler un choix arbitraire, mais statistiquement elle a pour effet que le CBOW lisse une grande partie de l'information distributive (en traitant un contexte entier comme une observation). Pour la plupart, cela s'avère être une chose utile pour les jeux de données plus petits. Cependant, skip-gram traite chaque paire contexte-cible comme une nouvelle observation, ce qui a tendance à être meilleur lorsque nous avons des ensembles de données plus volumineux.

6.2.2 Utilisation

Ce qui m'a motivé d'utiliser le word2vect est le problème de pertes données au niveau de l'entrer du Convolutinal Neural Network du premier algorithme, et aussi le plus qui va ajouter l'approche sémantique à la precision de l'algorithme.
Avant l'algorithme CNN traité en input un taille de données qui est égal à 30, cette taille constante ne permettait pas de représenter la vrai valeur d'entrée, ce qui parfois déclenche une déviation sur la vrai identité de la description.

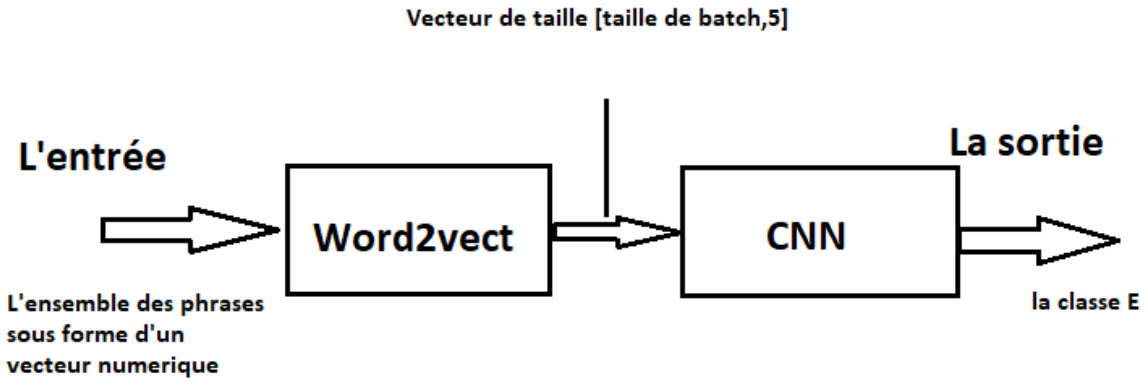

FIG. 6.2: Modélisation du modèle avec word2vect et cnn.

J'ai utilisé une taille de vecteur de word2vect égal à 5, parce que premièrement, mon corpus de données n'est pas assez grand, alors je vais économiser la mémoire, et deuxièmement pour minimiser la pertes de données au niveau de la fonction softmax.
La precision finale est égale à 80%, c'est bien meilleur que les deux approches discutées au paravent.

Conclusion & Perspective

Pour conclure, j'ai effectué mon stage de deuxième année de cycle d'ingénierie logiciel et intégration des systèmes informatique en temps que Data Scientist pour l'application mobile Nacharat au sein de l'entreprise SanadTech. Lors de ce stage de 3 mois, j'ai pu mettre en pratique mes connaissances théoriques acquises durant ma formation et aussi ça m'a permis de m'introduire dans une nouvelle route, celle du big data et machine learning.

Pendant cette expérience, j'ai tous d'abord traiter les données,en travailler à bien saisir son architecture, ce qui m'a permis de les transformer en des matrices numériques, puis après j'ai commencé par aborder le problème de classification depuis le coté de deep learning en utilisant le Réseau neuronal convolutif, ce qui m'a donné une precision en dessus de la moyenne, après j'ai voulu changer l'approche de deep learning, c'est pour cela que cette fois j'ai abordé le problème par l'une des approches du machine learning, en utilisant le Support vector machine avec un kernel gaussien, il m'a donné un résultat satisfaisant mais pas assez convaincant, et à la fin j'ai repris le premier algorithme développé, et j'ai travaillé à augmenter sa precision en utilisant le réseaux neuronal word2vec dans la phase d'entrer pour minimiser la perte des données et aussi pour données à l'algorithme d'apprendre la sémantique des mots,et tout cela pour avoir une precision bien meilleur.

Pour le Sprint finale, j' y compte rassembler les deux approches en un seul algorithme de classification, l'algorithme va contenir une seule entrée, la première couche va contenir deux parties indépendantes le CNN plus le Word2vec et le SVM , la deuxième couche va ressembler les résultats obtenus par la première couche on un vecteur qu'on va l'analyser par la suite par la méthode de Bayes, pour enfin avoir un résultat avec un taux de precision très élevé.