No file main.ind.

Package minitoc(hints) Warning : W0024 (minitoc(hints)) Some hints have been written (minitoc(hints)) in the main.log file.

LaTeX Warning : There were undefined references.

LaTeX Warning : Label(s) may have changed. Rerun to get cross-references right.

Package biblatex Warning : Please (re)run Biber on the file : (biblatex) main (biblatex) and rerun LaTeX afterwards.

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Faciliter la conception d'un assistant conversationnel avec le clustering interactif

THÈSE

présentée et soutenue publiquement le 01 mai 2023

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Erwan SCHILD

Composition du jury

Présidents :	Dr. Pascal CUXAC (à demander)
Rapporteurs :	Dr. Thomas LAMPERT (attendre réponse) Dr. (avoir une femme !)
Examinateurs :	Dr. Adrien COULET (à demander) Dr. (avoir une femme !)
Invités :	Dr. Gautier DURANTIN (à demander) Dr. Mathieu POWALKA (à demander)
Encadrants :	Dr. Jean-Charles LAMIREL Dr. Florian MICONI

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

M i s e n p a g e a v e c l a c l a s s e t h e s u l .

Résumé

Le résumé à faire.

Mots-clés: chat, chien, puces.

Abstract

The abstract to do

Keywords: cat, dog, flees.

Remerciements

Par la présente, je souhaites remercier :

☐ ma femme

☐ ma tortue

☐ ma famille

☐ mes amis

☐ mes collègies

☐ mes encadrants

☐ chatGPT

☐ ...

Je dédie cette thèse
à quelqu'un de bien.

Table des matières

Introduction

CHAPITRE À REFORMULER FAÇON SWALES

1 "*Asset centrality*"

SECTION À RÉDIGER

- ☐ Des enjeux ou problèmes actuels
  - Accessibilité à l'information : o Grosses bases documentaires, pas toujours ordonnées ;
  - Relations client à distance o Besoin d'un accessibilité h24 ;
- ☐ Utilisation de plus en plus fréquente des chatbots
  - Description succinte ;
  - Cas d'usage usuels ;
  - Tous les canaux d'utilisation ;
  - Avantages et Dérives potentiels de l'utilisation (emploi, biais, pertinence, ergonomie, ...) ;
- ☐ Révolution techniques fréquentes (règles, classification, modèles)
  - Moteurs de règles : o Basé sur la détecté de mots clés, o (+) facile à mettre en œuvre, o (-) peu robuste au langage naturel, o Paramétrage des réponses ;
  - Paramétrage intentions-entités : o Classification d'intention et/ou détection d'entités, o (+) plus robuste au langage naturel, facile à paramétrer, réponses contrôlées, o (-) demande de l'entrainement, des données, ..., o Paramétrage des réponses ;
  - Génération de réponse : o Réseau de neurones avec attention, o Transformers, o (+) plus robuste, o (-) plus complexe à mettre en œuvre, réponses non contrôlées, o Réponses non paramétrées ;
  - Approche hybride : o Cumul des trois approches pour cumuler certains avantages suivant les besoins ;

2 "*Estabilishing a Niche*"

SECTION À RÉDIGER

- ☐ Cadre industriel
  - Algorithme fixe
  - Données spécifiques

☐ GAP : Besoins de données

• Collecte de données spécifiqus au domaine traité : o extraction de base de données (solution simple), o collecte manuelle (organisation complexe, biais de collecte), o scraping (pas toujours fiable) ;

• Nombreux biais : o Biais„ o Réglementation, o Compétences (NOTRE COEUR DU SUJET), o ...

3 "*Occupying the Niche*"

SECTION À RÉDIGER

☐ Etude de l'organisation d'une entreprise pour concevoir ses jeux de données

☐ Etude de l'état de l'art pour concevoir des jeux de données

☐ proposition/contribution : une méthode adaptée pour un cadre industriel

Chapitre 1

État de l'art : concevons un jeu de données

Dans cette partie, nous allons faire un état des lieux des méthodes pour créer le premier jeu de données nécessaire à l'entrainement d'un assistant conversationnel. Cela comprend une description des acteurs du projet, un rappel de l'organisation usuelle en fonction de leur compétence, et une énumération des problèmes et solutions les plus communs.

TRANSITIOn À COMPLÉTERR

Rappel des contrainte indus-trielles

Sommaire

1.1 Rappel sur le fonctionnement usuel d'un chatbot

SECTION À RÉDIGER

- Description du cas d'un chatbot "classique" modélisé à base d'intention et d'entités o On se concentre sur ces implémentations car on peut y controller les réponses (image de marque en jeu)
- Classification d'intention (règles, classification supervisée, ...)
- Extraction d'entités (règles, ner, ...)

- Mapping des réponses sur la base du couple $(intention, enites)$
- **CITATION**

1.2 Les étapes usuelles de conception d'un chatbot

SECTION À RÉDIGER

Préambule : l'organisation peut bien entendu varier suivant les contextes, mais la description qui suit est représentative des organisation principales

1.2.1 Définition des acteurs

• Data scientistes : o Experts en IA o Peu de connaissance métier, i.e. peu de regarde critique sur la pertinence des résultats (autre que statistique)

• Expert métier : o Pas de connaissance en IA, i.e. nécessitent des formations o Connaissance métier forte, i.e. peuvent décrire la pertinence d'un résultat

• Chef de projet o Pas de connaissance en IA o Pas de connaissance métier o Connaissance du besoin (hypothèse non vérifiée car parfois ils ne savent pas ce qu'ils veulent dû à la méconnaissance des capacités de l'IA)

1.2.2 Cadrage du projet

• Objectifs : o Clarification du besoin, o Définition du périmètre couvert (i.e. les fonctionnalités et réponses à proposer),

• Livrable : un cahier des charges

1.2.3 Collecte des données

• Souvent pas de données à disposition : o En R&D, "80%" sur la recherche d'algo sur des données publiques, d'où le besoin de datascientists, o En entreprise, "80%" sur la gestion des données privées/spécifiques sur des algo connus, d'où le besoin d'experts métiers ;

• Risque de biais dans les données : o Biais d'échantillon : la collecte ne représente pas la réalité, o Biais de sélection : le trie de la collecte ne représente plus la réalité, o Biais de confirmation : on garde les données qui nous arrangent, o Biais de valeur : les données ne sont pas éthiquement représentatives, o Biais de contexte : les données d'un cas d'usage ne sont pas toujours réutilisables pour un autre cas d'usage (ex : différence entre les jargons des AV clients et celui des AV conseillers) ; o **A COMPLETER**

• Livrable : une collecte de données brutes

1.2.4 Modélisation d'une structure et Labellisation des données

• Le coeur "métier" de la création du projet ;

• Objectif : Définition d'une modélisation sur la base des besoins attendus restreints au périmètre à couvrir ;

• En théorie : o Intention : verbe d'actions, o Entités : informations complémentaires, personnes, date, lieux, montants, noms de produits, ... ;

• Complexité de la tâche : o Intention abstraite : définition difficile voir subjective, ... o Annotation difficile : différence entre théorie et pratique, données ambigues, ... o Plusieurs itérations

car modélisation trop théorique / pas pratique o Besoins de beaucoup de formation (pour donner la compétence aux experts) et d'atelier (pour se mettre d'accord)

- Livrable : un jeu de données annotées

1.2.5 Entrainement et tests

- Le coeur "technique" de la création du projet ;
- Objectif : avoir un modèle qui soit adapté à son utilisation en production
- En théorie : o Split en train et tests o Entrainement et tests o Association des réponses
- Complexité de la tâche : o Modélisation précédente pas toujours adaptée : OK pour un métier, mais pas possible à entrainer à cause de déséquilibre, de manque de données, ... o Algorithme fixe mais données variables : savoir quelle modélisation est la plus adaptée est compliqué à deviner o Réponses pas toujours adaptées aux questions : décalage entre entrainement (modélisation théorique) et réponse (modélisation pratique)

1.2.6 Déploiement de la première version

- RAS
- Parfois la modélisation est décalée par rapport à l'utilisation en production o Comportement en moteur de recherche avec des questions courtes o Vocabulaire non maitrisé par les utilisateurs o problème d'ergo ou d'expérience utilisateur

1.2.7 Amélioration continue

- Vérification du comportement ;
- Ajustement du modèle ;
- Déploiement des versions suivantes.

1.3 Zoom sur la partie Modélisation et Labellisation de la base d'apprentissage

SECTION À RÉDIGER

1.3.1 Création « manuelle »

- Enchainement de plusieurs ateliers/cycles : o Définition d'une structure en atelier et Annotation des données o Premier conflit : La structure est trop théorique o Redéfinition et Ré-annotation o Second conflit : Les structure ou les données ne sont pas adaptées o Collecte complémentaire, Redéfinition et Ré-annotation
- Avantages : o Transmission progressive du savoir aux datascientist o Test des modélisations potentielles
- Inconvénients : o Nombreux ateliers o Nombreuses remises en questions / aller-retour de conception o L'avis initiale sur le périmètre à couvrir est flou quand cela concerne une centaine de demandes clients o Se base sur de la connaissance que les experts métiers n'ont pas o Comment les aider dans ce problème d'organisation ?

1.3.2 Création assistée par des regroupements non-supervisés

• Constat : o Pour des jeux de données à taille humaine (moins de 20.000 données), le premier tri est parfois "optimisé" manuellement sur la base des patterns commun (ordonnancement alphabétique)

• Solution : o Un clustering pourrait simplifier cette tâche ! o Rappel : grandes lignes du fonctionnement d'un algorithme de clustering ? o NB : une section ou une annexe détaillera les algorithmes de les plus utilisés

• Avantages : o Regroupement automatique o Découverte de la structure

• Inconvénients : o Les résultats sont souvent peu pertinents o Similarité par entités, et pas par intentions o Nuances métiers non comprises o Plusieurs soucis si le jeu de données est déséquilibré ou spécifique o Absence d'un modèle de langue spécifique au contexte... o parfois besoin d'hyperparamètres complexes à déterminer

1.3.3 Conception assistée par des regroupements semi-supervisés

• Solution : o On peut envisager ainsi de corriger le clustering en y insérant des contraintes métiers LAMPERT et al., 2018 o Méthodes semi-supervisée o NB : une section ou une annexe détaillera les algorithmes de clustering sous contraintes

• Interactions possibles avec le clustering (sur la base de proposition de l'humain) o Sur les données / sur le résultat : ajouts de contraintes sur les données, suppressions ou modifications manuelles de données, réorganisation manuelles des clusters, ... o Sur les paramètres : modifier les hyper-paramètres, modifier le nombre de clusters, modifier les embeddings, utiliser d'autres algorithmes, ... o Besoin de visualisation : vue des contraintes, de la représentation vectorielle, ...

• Avantage : o On a réglé les problèmes de pertinence en ajoutant des contraintes

• Inconvénients : o Choisir comment modéliser ces contraintes peut être complexe o Surtout énorme en ajoutant des contraintes o Choisir les contraintes pertinentes est une tâche difficile

1.3.4 Conception basée sur des méthodes d'apprentissage actif

• Solution : o On peut demander à la machine de définir les contraintes dont elle a besoin pour s'améliorer / confirmer son comportement o On peut séparer et cibler les tâches pour que le clustering se nourrissent des commentaires de l'expert et que l'expert corrige ce qui semble utile au clustering o Sous-entendu : Préférer la collaboration à la supériorité (que ce soit celle de la machine ou celle de l'expert) o NB : une section ou une annexe détaillera les interactions possibles entre homme et machine

• Interactions possibles avec le clustering (sur la base de propositions de la machine) o Sur les données / sur le résultat : proposition de suppression de données abhérrantes, proposition d'ajout de contraintes à des endroits stratégiques, ... o Sur les paramètres : réévaluation des paramètres, combiner plusieurs algorithmes et synthétiser le résultat, ...

• Avantage : o On a réglé les problèmes de pertinence et de coûts en ajoutant des contraintes

• Inconvénients / problème à résoudre : o Accepter de collaborer avec la machine (problème UX, ergo, accompagnement au changement) o Il faut prouver cette méthode

Chapitre 2

Proposition d'un Clustering Interactif

Dans le chapitre précédent, nous avons vu les points essentiels suivants :

- ☑ Dans un cadre industriel, le choix de l'algorithme utilisé pour l'entrainement d'un modèle est déterminé à l'avance, donc la qualité de l'assistant repose principalement sur la fiabilité et la pertinence de son jeu de données ;
- ☑ Pour concevoir ce jeu de données, il est nécessaire de faire appel à des experts maîtrisant le domaine à couvrir par l'assistant car les données sont en général spécifiques ou privées ;
- ☑ L'intervention de ces experts métiers au sein du projet est en général laborieuse : d'une part à cause de leur manque de connaissances en datascience (ce n'est pas leur domaine d'expertise), d'autre part à cause de la complexité inhérente des tâches de modélisation et d'annotation des données.
- ☑ Par manque de compétences, de connaissances ou d'ergonomie, la tâche de conception d'un jeu de données reste manuelle et est encore mal assistée par ordinateur par manque d'ergonomie ou de faisabilité.

Dans cette partie, nous proposons une alternative à l'organisation manuelle destinée à la conception d'un jeu de données. Notre proposition vise à remplir un double objectif :

- ☐ Proposer une méthode permettant d'assister la modélisation et l'annotation des données pour créer plus efficacement une base d'apprentissage pour la classification d'intention d'un assistant conversationnel ;
- ☐ Redéfinir les tâches et les objectifs des différents acteurs afin de rester au plus proche de leurs compétences réelles, particulièrement en ce qui concerne les experts métiers intervenants dans le projet.

Sommaire

2.1 Intuitions à l'origine de notre méthode

La pierre angulaire de notre méthode repose sur le fait qu'il est difficile pour un expert métier de classer une question suivant une modélisation abstraite prédéfinie : cela l'éloigne de ses compétences initiales, nécessite en contre-partie de nombreuses formations, et introduit de nombreuses erreurs d'annotations. De fait, il semble plus adéquat de demander à l'expert métier de discriminer deux questions sur la base de leurs réponses : une telle approche demande une charge de travail plus faible et est plus intuitive car elle est plus proche des compétences réelles de l'annotateur. Ainsi, nous basons notre méthode sur l'annotation de contraintes sur les données.

Référence

Référence

Toutefois, l'annotation de contraintes semble elle aussi fastidieuse. En effet, pour faire émerger une base d'apprentissage, il faut annoter un grand nombre de contraintes et être attentifs aux éventuelles incohérences pour ne pas introduire de contraintes contradictoires. Pour assister l'expert dans cette tâche, nous avons donc décidé de l'intégrer dans une stratégie d'apprentissage actif en essayant de tirer parti des interactions possibles avec la machine. Ce choix est motivé entre autre par l'intuition qu'il est possible de coopérer avec la machine pour obtenir plus efficacement un résultat pertinent.

Référence

à reformuler plus tard.

C'est sur la combinaison de ces deux éléments que repose notre méthode d'annotation pour concevoir le jeu d'entrainement de notre assistant conversationnel.

2.2 Description théorique de notre clustering interactif

Nous proposons la méthode suivante pour transformer une collecte de données bruts en une base d'apprentissage nécessaire à l'entrainement d'un assistant conversationnel. Cette méthode, que nous appelons "*clustering interactif*", est décrite formellement à l'aide du pseudo-code figurant dans Alg. 2.1.

Algorithme 2.1 Description en pseudo-code de la méthode d'annotation proposée employant le clustering interactif

Entrée(s): données non segmentées ; budget à disposition
initialisation : créer une liste vide de contraintes
optionnel : évaluer les hyper-paramètres de la segmentation automatique
segmentation initial : regrouper les données par similarité
répéter
    optionnel : évaluer les hyper-paramètres de l'échantillonnage
    échantillonnage : sélectionner une partie de la segmentation à corriger
    annotation : corriger la segmentation en ajoutant des contraintes sur l'échantillon
    optionnel : ré-évaluer les hyper-paramètres de la segmentation automatique
    segmentation : regrouper les données par similarité et avec les contraintes
    évaluation (1) : estimer la pertinence et la stabilité de la segmentation
    évaluation (2) : estimer le budget restant et les coûts restant à investir
jusqu'à segmentation satisfaisante OU budget épuisé
Sortie(s): données segmentées (i.e. base d'apprentissage)

Comme vous pouvez le constater, la méthode repose principalement sur l'alternance successive entre deux phases clefs :

— une phase d'**annotation de contraintes** par un expert sur la base des connaissances qu'il détient ;

— une phase de **segmentation automatique** des données par une machine sur la base de la proximité sémantique des données et des contraintes précédemment annotées.

L'objectif recherché en associant ces deux phases est la création d'un cercle vertueux pour améliorer itérativement la qualité de la base d'apprentissage en cours de construction. En effet, à chaque itération, l'expert métier obtiendra une proposition de segmentation des données qu'il pourra raffiner pour corriger le fonctionnement de la machine et ainsi obtenir une segmentation plus pertinente à l'itération suivante.

Pour l'**initialisation** la méthode (cf. Alg. 2.1, *lignes 1 à 3*), nous définissons une liste vide de contraintes : tout au long du processus, cette liste contiendra l'ensemble de la connaissance que l'expert transmettra au système sous la forme de contraintes simple sur les données (nous entrerons en détails en décrivant la phase d'annotation). De plus, il faut une première segmentation des données par la machine : celle-ci se réalise par l'exécution d'un algorithme de clustering. Nous estimons qu'il n'est pas du ressort de l'expert métier de choisir de l'algorithme de clustering et ses hyper-paramètres. Ces derniers pourront être déterminés par un data scientist en fonction du problème à traiter ou laissés par défaut. Il est à noter que cette segmentation des données est réalisée sans bénéficier de la connaissance de l'expert, il est donc peu probable que le résultat soit pertinent à ce stade.

Nous entrons dans le coeur de la boucle itérative par la phase d'**échantillonnage** (cf. Alg. 2.1, *lignes 5 et 6*). Comme mentionné au préalable, savoir quelles contraintes ajouter pour corriger efficacement le clustering est un problème NP-difficile (le nombre de possibilité croît proportionnellement au carré du nombre de données). De plus, l'intervention d'expert est chiffrée et représente en général la majeure partie des coûts à investir dans un projet. Il est donc inconcevable de laisser un expert métier annoter des contraintes "seul" et "au hasard". Ainsi, pour optimiser ses interventions, il convient de déterminer là où l'expert aura le plus d'impact lors de sa transmission de connaissance. C'est pourquoi la phase d'échantillonnage est primordiale dans la méthode proposée : Nous proposons d'y sélectionner des couples de données sur la base de leur similarité, de leur segmentation ou encore de leur relations avec d'autres données déjà liées par d'autres contraintes.

Sur la base de cet échantillon, l'expert peut entamer son étape d'**annotation de contraintes** (cf. Alg. 2.1, *ligne 7*). Pour alléger la charge d'annotation, nous avons décidé de discriminer les données de l'échantillon par des contraintes binaires simples : *MUST_LINK* et *CANNOT_LINK*. Ces contraintes représentent respectivement la similitude ou la différence entre deux données, et seront utilisées pour regrouper ou séparer certaines données dans la prochain segmentation. En fonction de l'orientation du projet et afin de rester au plus proche des compétences réelles de l'expert, la formulation de l'énoncer d'annotation doit être judicieusement définie : par exemple, les contraintes peuvent représenter une similitude sur la thématique concernée [1], sur l'action désirée [2], ou encore sur le besoin de l'utilisateur [3]. On notera que ces contraintes respectent des règles de transitivité (cf. figure), ce qui peuvent introduire des incohérences dans les contraintes si elles ne sont pas vérifiées [4].

Pour finir, la dernière phase de cette boucle est composée d'une nouvelle **segmentation** des données (cf. Alg. 2.1, *lignes 8 et 9*). Cette devra respecter les contraintes préalablement définies par l'expert, nous nous tournons donc vers l'utilisation d'un clustering sous contraintes. Au fur et à mesure des itérations, de plus en plus de contraintes seront ajoutées pour corriger le clustering.

utiliser l'appellation clustering ou segmentation ?

cf. partie étude

référence

descriptio technique plus tard ?

figure

1. thématique : *crédit* vs. *assurance*
2. action : *souscrire* vs. *résilier*
3. besoin : *souscrire un crédit* vs. *souscrire une assurance*
4. Si (d_1, d_2) sont liées par un *MUST_LINK* et (d_2, d_3) sont liées par un *CANNOT_LINK*, alors (d_1, d_3) devrait en toute cohérence être liées par un *CANNOT_LINK*

ainsi, au bout d'un certain nombre d'itérations, la segmentation des données reflétera la vision que l'expert aura voulu transmettre. Comme précédemment, nous estimons qu'il n'est pas du ressort de l'expert métier de choisir de l'algorithme de clustering et ses hyper-paramètres. Ces derniers pourront être déterminés par un data scientist en fonction du problème à traiter, estimés en fonction de l'itération et des contraintes disponibles, ou laissés par défaut.

cf. partie étude

description technique plus tard ?

Comme la méthode est itérative, il faut pouvoir estimer des **cas d'arrêt** (cf. Alg. 2.1, *lignes 10 à 12*). Le cas d'arrêt le plus évident n'est pas technique mais relatif aux coûts investis dans l'opération : si le projet n'a plus de budget dédié à l'annotation, il faudra créer la base d'apprentissage avec le résultat à disposition, quel que soit la pertinence de la segmentation obtenue sur les données. Ce cas d'arrêt par défaut peut malheureusement être synonyme d'échec pour le projet si les résultats sont inexploitables. D'autres cas d'arrêts plus techniques peuvent être envisager en fonction de la qualité de la segmentation. D'une part, nous pouvons comparer l'évolution de la segmentation des données : si les segmentations sont similaires sur plusieurs itérations, il est possible que la modélisation atteint un optimum local ou un palier de performance. D'autre part, nous pouvons aussi comparer l'évolution de l'accord entre la segmentation obtenue et l'annotation de l'expert : en effet, si l'expert ne contredit plus la répartition proposée des données, il est probable sa vision et la vision de la machine aient convergé. Dans les deux cas, l'analyse de l'expert métier reste nécessaire pour valider si la modélisation des données est pertinente ou si elle comporte encore des incohérences à corriger.

description technique plus tard ?

2.3 Description technique et implémentation

SECTION À RÉDIGER ? EN ANNEXE ? DANS LA PARTIE PRÉCÉDENTE ?

- cognitivefactory.interactive-clustering : Gestion des données
- cognitivefactory.interactive-clustering : Gestion des contraintes + conflits
- cognitivefactory.interactive-clustering : Algorithmes de clustering o Kmeans : Classique, Incontournable, Rapide, Efficace o Hiérarchique : Lent mais facile à implémenter o Spectral : Permet des topologies complexes o DBScan : Classique, Incontournable, Rapide, Efficace, Peu d'hyperparamètres o Affinity propagation : o Metric learning : Lent mais plus adapté au corpus o ...
- cognitivefactory.interactive-clustering : Algorithmes de sampling o Random ou Pseudo-random o Farhtest : Scinder les gros clusters o Closest : Redéfinir la position des frontières de clusters o ...
- cognitivefactory.interactive-clustering-gui : Diagramme d'état ? o Boucle itérative entre clustering, échantillonnage et annotation o Améliorer le résultat précédent o Autant de boucle que « nécessaire » o Avoir le clustering le plus efficace pour avoir de bon résultats o Avoir l'échantillonnage le plus efficace pour améliorer le plus efficacement o Avoir une annotation sans ambiguïté pour ne pas biaiser la construction itérative
- cognitivefactory.interactive-clustering-gui : Interface d'annotation o MUST-LINK / CANNOT-LINK / SKIP o « Répondriez-vous de la même manière à ces deux demandes ? » o Formulation de la question
- cognitivefactory.interactive-clustering-gui : Interface d'analyse o Analyse de l'évolution de l'accord clustering->annotation o Analyse des patterns linguistiques pertinents o Analyse de la formation de clusters (taille, répartition, ...)
- NB : captures d'écrans pour donner un aperçu, puis redirection vers les annexes

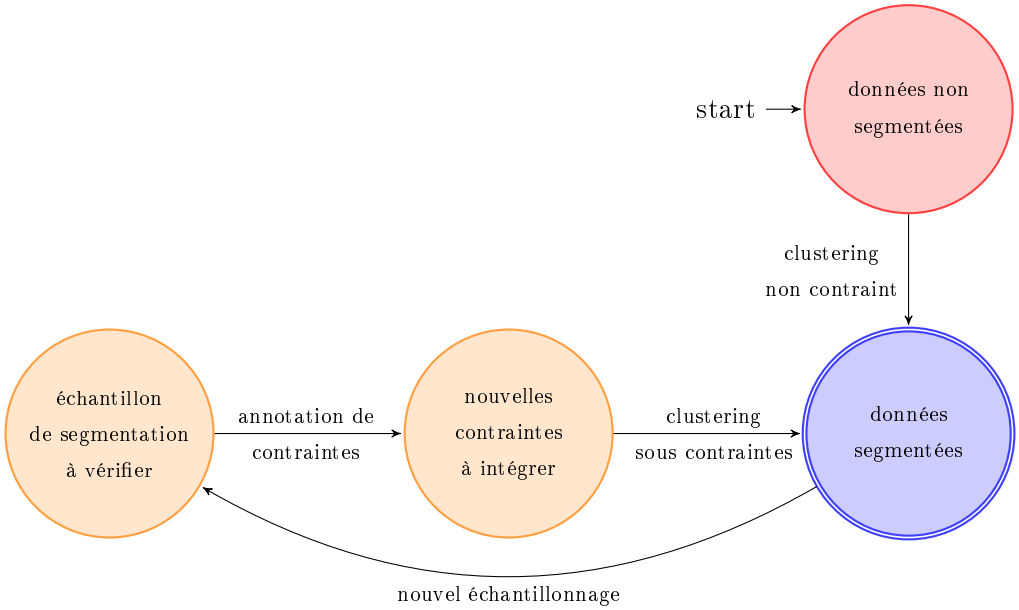

FIGURE 2.1 – Diagramme d'état représentant les grandes étapes du clustering interactif.

2.4 Espoirs de la méthode proposée

SECTION À RÉDIGER

• Moins de formations, d'ateliers, ... • Se concentrer sur son domaine de compétence (i.e. pas de datascience pour les experts métiers) • Permettre de trouver la base d'apprentissage • Méthode réaliste / pas trop coûteuse • ...

2.5 Protocole d'utilisation : Mode d'emploi associé (? ?CONCLUSION ? ?)

SECTION À RÉDIGER

- Collecte des données
- Itération de clustering > échantillonnage > annotation
- A chaque conflit : correction nécessaire
- A la fin d'un clustering : caractériser la pertinence métier avec FMC
- A chaque itération : voir l'évolution par rapport à la précédente

NB : la démonstration de cette proposition de protocole sera démontrée dans la partie 3.

Chapitre 3

Etude de la méthode

Sommaire

Transition / Récap : Cette méthode étant nouvelle, plusieurs études sont nécessaires :

1. Est-ce que la méthode marche/converge ?
2. Quelle est la meilleure implémentation ?
3. Quels sont les coûts à investir ?
4. Quel est l'impact d'une différence / erreur d'annotation ?
5. Comment interpréter les résultats et leur donner du sens ?
6. Quand s'arrêter ?

(à formuler sous la forme d'hypothèse de travail à vérifier ? et/ou formuler/rappeler ces hypothèse en début de chaque sous-partie)

==> Remarque en préambule : C'est compliqué de remettre en cause l'annotation manuelle avec des vérité terrain conçues manuellement... Certaines études ne sont donc pas faisables

3.1 Etude de viabilité : « est-ce que la méthode marche ? »

3.1.1 Etude de convergence vers une vérité terrain établie

• Expérience : Faire des itérations de « clustering > échantillonnage > annotation » jusqu'à convergence vers la vérité terrain o Jeu de données : Carte bancaire (500 questions) o Cf. EGC/IJDWM

• Résultats : o Les contraintes sont respectées o La vérité terrain est atteinte (La vmeasure atteint 100%) o L'ajout de contraintes permet vite d'être meilleur qu'un clustering simple

• Conclusion : la méthode est viable : on trouve bien vérité terrain o Avantage : Pas besoin de définir une structure, elle se découvre toute seule o Avantage : Besoin de peu de connaissance en IA (same ? oui ou non) o Inconvénient : Besoin de « beaucoup » de contraintes o ==> IL FAUT UNE OPTIMISATION (IV.B.)

• Discussion : balance entre annotations de contraintes et clustering o Si pas assez de contraintes : alors clustering pas assez pertinent ! o Si 100% de contraintes : alors résultat trop subjectif ! o Trouvons une juste milieu : supposons 80% pour la suite (annotation partielle)

3.2 Etude technique : « quelle est la meilleure implémentation ? »

3.2.1 Etude des paramètres optimaux

• Expérience : Faire des itérations de « clustering > échantillonnage > annotation » jusqu'à convergence vers la vérité terrain et étudier des itérations clés pour déterminer quelle implémentation est la plus efficace. o Jeu de données : Carte bancaire (500 questions) o Paramètres étudiés : prétraitement, vectorisation, sampling, clustering o Métrique : VMeasure du clustering obtenue avec la vérité terrain en fonction du nombre de contraintes annotées o Cf. EGC/IJDWM

• Résultats : o Tous les paramètres sont importants, surtout le sampling et le clustering o Meilleure implémentation trouvée

• Conclusion : o Il y a donc moyen d'optimiser la méthode pour annoter moins de contraintes

• Remarques : o Expliquer le choix du kmeans : rapide, efficace, itération rapide o Expliquer le choix du closest : favorise les MUST-LINK o Expliquer le non-choix du farthest : favorise CANNOT-LINK, mais n'aide pas

3.3 Etude des coûts : « quels sont les coûts à investir ? »

3.3.1 Etude du temps d'annotation

• Expérience : Faire annoter des contraintes par plusieurs annotateurs o Jeu de données : Carte bancaire (500 questions) ou Titre de journaux (? ? ?)

• Résultats : (**! !TODO ! !**)

• Conclusions : (**! !TODO ! !**)

• Discussions : Définition d'un batch moyen

3.3.2 Etude du temps de calcul

• Expérience : Estimer le temps de calcul de chaque algorithme o Jeu de données : Carte bancaire (1000 questions, artificiellement augmenté jusqu'à 1000 données) o Estimation des facteurs influents : nombre de données, nombre de contraintes, nombre de clusters, hyper-paramètres, effets aléatoires, . . .

• Résultats : o Estimation des temps de calcul et des facteurs influents

• Conclusions : o Certains algorithmes sont longs. . . o Définition d'une fonction d'estimation du temps de calcul pour chaque algo

• Discussion : balance entre performance et coût temporel o Les itérations doivent être fluides o On a plus à gagner à ajouter des contraintes qu'attendre un algo trop long o Un optimal serait d'annoter des batch d'une durée d'un algo pour ne pas avoir trop à attendre entre deux itérations

3.4 Etude des erreurs : « quel est l'impact d'une différence d'annotation ? »

Préambules

• Source potentielle des différences : o Différence de points de vue, o Erreurs d'inattention, o Données ambigües ;

• Identification des différences : o Relectures / revue d'annotation, o Détection des incohérences entre contraintes, o Adjudication / ajout de redondance ;

3.4.1 Etude de l'impact d'une erreur par simulation

• Expérience : Faire des itérations de « clustering > échantillonnage > annotation » en simulant un pourcentage d'erreur d'annotation o Jeu de données : Carte bancaire (500 questions), o Pourcentage d'erreur variable, o Prise en compte ou non de la résolution de conflits ;

• Résultats : o Peut gravement impacter le résultat, o N'est pas détectable sans redondance ;

• Conclusions : (**! !TODO ! !**)

• Discussions : importance de la fiabilité des annotations : o Besoin de savoir ce que l'on veut : Ajouter de l'adjudication en début de projet pour confronter rapidement les visions, o Besoin de limiter les erreurs : Ajout de redondance ou de stratégie de vérification (de nouvelles méthode de sélection) mais cela entraîne un surcoût, o Préférer passer l'annotation que de forcer une annotation ambiguë ;

3.4.2 Etude de ré-annotation d'un chatbot existant

• Expérience : Ré-annoter un AV existant pour identifier les inconvénients pratiques : o Jeu de données : Moyens de paiements (8000 questions de production), o Deux annotateurs avec des

règles « floues » (comme c'est le cas en initialisation) mais avec un « objectif » commun (arbre de dialogues & réponses « connues »), o Revues d'annotations à mi-projet ;

• Résultats : o 23% de désaccord, mais au moins 77% d'accord sans concertation ! o Après revue d'annotation : (???), o Expériences interrompue ;

• Conclusions : (**!!TODO!!**)

• Discussions : Quelques conseils : o En début de projet : Adjudication pour confirmer les visions, o Correction des incohérences le plus rapidement possible, o D'autres sélection pour ajouter de la redondance, o Mais cela représente un cout supplémentaire,

3.5 Etude métier : « comment interpréter les résultats et leur donner du sens ? »

Préambule de définition de la FMC ou référence à l'état de l'art / annexe :

3.5.1 Etude de la caractérisation du clustering par FMC

• Expérience : Faire des itérations de « clustering > échantillonnage > annotation » jusqu'à convergence vers la vérité terrain et étudier la FMC du clustering en cours : o Jeu de données : Carte bancaire (500 questions), o Métrique : VMeasure adaptée pour comparer la FMC d'un clustering et la FMC de la vérité terrain ;

• Résultats : (**!!TODO!!**)

• Conclusions : (**!!TODO!!**)

• Discussions : (**!!TODO!!**)

3.6 Etude d'arrêt : « quand le résultat est-il satisfaisant ? »

3.6.1 Etude d'un prérequis d'arrêt : la cohérence du clustering obtenu

• Expérience : Faire des itérations de « clustering > échantillonnage > annotation » jusqu'à convergence vers la vérité terrain et étudier si un classifier entrainé sur cette base est stable, i.e. s'il arrive à retrouver sa base d'apprentissage o Jeu de données : Carte bancaire (500 questions), o Remarque : si la base d'apprentissage n'est pas stable, le clustering doit encore être modifié. . .,

• Résultats : (**!!TODO!!**)

• Conclusions : (**!!TODO!!**)

• Discussions : (**!!TODO!!**)

3.6.2 Etude d'un critère d'arrêt : l'accord entre un batch d'annotation et le clustering précédent

• Expérience : Faire des itérations de « clustering > échantillonnage > annotation » jusqu'à convergence vers la vérité terrain et étudier l'évolution des accords/désaccords entre l'annotateur et le clustering o Jeu de données : Carte bancaire (500 questions), o Remarque : si l'accord est maximal, l'annotateur n'a plus de valeur ajoutée car le clustering n'est jamais modifiée,

• Résultats : (**!!TODO!!**)

• Conclusions : (**!!TODO!!**)

• Discussions : (**!!TODO!!**)

3.6.3 Etude d'un critère d'arrêt : la similitude entre deux clustering consécutifs

• Expérience : Faire des itérations de « clustering > échantillonnage > annotation » jusqu'à convergence vers la vérité terrain et étudier l'évolution des similitudes entre le clustering courant et le clustering précédent o Jeu de données : Carte bancaire (500 questions), o Remarque : si la similitude est forte, c'est que le clustering devient stable,

• Résultats : (**!!TODO!!**)
• Conclusions : (**!!TODO!!**)
• Discussions : (**!!TODO!!**)

3.6.4 Etude d'un critère d'arrêt : la similitude entre deux FMC de clustering consécutifs

• Expérience : Faire des itérations de « clustering > échantillonnage > annotation » jusqu'à convergence vers la vérité terrain et étudier l'évolution des similitudes entre la FMC du clustering courant et la FMC du clustering précédent o Jeu de données : Carte bancaire (500 questions) o Remarque : si la similitude est forte, c'est que le clustering devient stable

• Résultats : (**!!TODO!!**)
• Conclusions : (**!!TODO!!**)
• Discussions : (**!!TODO!!**)

3.7 Autres études à réaliser (!!PAS FAIT PAR MANQUE DE TEMPS!!)

3.7.1 Choix du nombre de clusters ==> problème de recherche complexe

o Piste de résolution : plusieurs clusterings + vote collaboratif ? algorithmes sans le nombre de clusters en hyper-paramètres

3.7.2 Impact d'un modèle de langage ==> nécessite de nombreuses données spécifiques au domaine

o Piste de résolution : script d'étude comparative déjà prêt, mais il manque les données opensources...

3.7.3 Paradigme d'annotation (intention vs dialogue) ==> problème d'UX + objectif métier

o Etude Ergo, sort de mon domaine d'expertise

3.7.4 (et plein d'autres que j'ajouterai au fur et à mesure de ma rédaction)

o

Conclusion

1 Rappel de la problématique ? ?

TODO

2 Avantage et limites de la méthodes ? ?

TODO

3 Ouverture ? ?

TODO

Annexe A

Annexe théorique

Sommaire

A.1 Les algorithmes de clustering

A.1.1 Kmeans

kmeans

A.1.2 Hierarchique

hierarchique

A.1.3 Spectral

spectral

A.1.4 DBScan

dbscan

A.1.5 Affinity Propagation

affinity propagation

A.2 Evaluation d'une clustering

A.2.1 Homogénéité – Complétude – Vmeasure

la VMeasure est la moyenne harmonique entre l'homogénéité et la complétude.

A.2.2 FMC

Annexe B

Annexe technique

Sommaire

B.1 package pypi interactive-clustering

B.2 package pypi interactive-clustering-gui

B.3 package pypi features-maximization-metrics

B.4 experimentations jupyter notebook

Annexe C

Annexe des jeux de données

Sommaire

C.1 french bank cards

C.2 DNA press title

Bibliographie

Lampert, T., Dao, T.-B.-H., Lafabregue, B., Serrette, N., Forestier, G., Cremilleux, B., Vrain, C., & Gancarski, P. (2018). Constrained distance based clustering for time-series : a comparative and experimental study. *Data Mining and Knowledge Discovery*, *32*(6), 1663-1707. https://doi.org/10/gfbpj8

Liste des TODOs

Liste des figures

Liste des tableaux

Liste des algorithmes

@markListe des algorithmes

Liste des algorithmes

Glossaire

clustering !!TODO!!. 6

Index