Prévention du risque de suicide via les réseaux sociaux
Détection de points de rupture dans le comportement des personnes à risques

Étude bibliographique

Master **Sciences et Technologies**,
Mention **Informatique**,
Parcours DECOL

Auteur
Cédric Maigrot

Superviseurs
Sandra Bringay
Jérôme Azé

Lieu de stage
LIRMM UM5506 - CNRS, Université de Montpellier

version 5 — 5 mars 2015

Résumé

Le suicide devient d'année en année une problématique plus préoccupante. Les organismes de santé (l'OMS et l'ONS) se sont engagés à réduire le nombre de suicide de 10% dans l'ensemble des pays d'ici 2020. Même si le suicide est un geste impulsif, il existe souvent des comportements et des paroles qui peuvent révéler un mal être et être des signes précurseurs de prédispositions au suicide. L'objectif de ce stage est de mettre en place un système pour détecter semi-automatiquement ces comportements et ces paroles au travers des réseaux sociaux.

Des travaux précédents [1] ont proposé le classement de messages issus de Twitter suivant des thèmes liés au suicide : tristesse, blessures psychologiques, état mental, dépression, peur, solitude, description de la tentative de suicide, anorexie, réactions aux insultes et cyber-harcèlement. Dans ce stage, l'étude se focalise plus précisément sur la réaction des jeunes au sujet du cyber-harcèlement. Nous étudierons notament des messages anonymisés de jeunes qui contactent l'Organisme Arrêt Demandé International (OADI), spécialisé dans l'aide aux personnes victimes de harcèlement. La 1ère partie de cette bibliographie portera sur les méthodes d'analyse de sentiments.

Outre les problèmes classiques liés à l'analyse des textes mal rédigés issus des réseaux sociaux, un des principaux challenge de ce stage est de mettre en place un système de gestion des flux de données. De nouveaux messages pouvent apparaître à tout moment et doivent être pris en compte dans l'analyse. Cela implique la mise en place de techniques spécifiques telles que le *concept drift* sur lequel portera la 2e partie de cet état de l'art et qui permettront de capter l'apparition d'un nouveau sentiment comme la peur.

Mots clés : Analyse de sentiments, Flux de données, Concept drift

---

Abstract

This master thesis.

Table des matières

1

Contexte

≪ Le suicide est évitable. Pourtant, toutes les 40 secondes, une personne se suicide quelque part dans le monde et bien plus tente de mettre fin à ses jours. Toutes les régions et toutes les tranches d'âge sont touchées, notamment les jeunes de 15 à 29 ans, pour qui le suicide est la deuxième cause de mortalité à l'échelle mondiale ≫[1]. Le constat au niveau national, avec le premier rapport annuel de l'Observatoire National du Suicide (ONS) en décembre 2014, est identique[2]. Des mesures de prévention sont mises en place par les organismes de santé car le suicide devient un problème majeur.

Parmi les principales causes de suicide, on trouve le harcèlement. Lors d'un harcèlement "ordinaire", la victime est harcelée sur un lieu précis (école, travail, bus, ...). Alors que dans le cas d'un **cyber-harcèlement**, les messages vont continuer sur Facebook, Twitter ou tout autre réseau social que l'on serait amené à fréquenter en étant en dehors du lieu de harcèlement. Il n'existe alors plus de répit pour la victime qui se retrouve dans un contexte d'oppression menant parfois au suicide. Dans le cadre de ce stage, l'étude portera spécifiquement sur les liens entre cyber-harcèlement et idéations suicidaires.

L'Organisme Arrêt Demandé International[3] (OADI), vient en aide aux personnes victimes de harcèlement. Il étudie le phénomène du harcèlement à l'école dans douze pays grâce à ses filiales associatives internationales. Leur but est de comprendre comment le phénomène du harcèlement scolaire est appréhendé dans les autres pays du monde afin d'améliorer son approche établie en France. L'OADI met à notre disposition des messages de jeunes les contactant et pouvant exprimer des idéations suicidaires.

Le but du stage est de parvenir à prédire les points de rupture dans le comportement des personnes suite à un cyber-harcèlement. Les résultats pourront être appliqués aux futurs messages reçus par OADI, mais aussi plus généralement sur des comptes d'utilisateurs de réseaux sociaux. Pour mener à bien ce projet, il sera nécessaire de se focaliser sur trois aspects : 1) la détection des harcèlements ; 2) la détection des sentiments exprimés par la personne harcelée ; 3) la détection des *concept drifts* sur ces sentiments (e.g. l'apparition d'un sentiment de perte d'estime de soi). Ce rapport étant court, nous ne décrirons pas le premier aspect mais le travail bibliographique a également été réalisé. On peut citer par exemple les méthodes de [21] ou de [35]. Les deux autres aspects seront traités dans les sections 2 et 3.

[1] Prévention du suicide, L'état d'urgence mondial. OMS 2014. ISBN : 978 92 4 256477 8

[2] Marisol Touraine reçoit le premier rapport annuel de l'Observatoire national du suicide - Ministère des Affaires sociales, de la Santé et des Droits des femmes - http ://www.sante.gouv.fr/marisol-touraine-recoit-le-premier-rapport-annuel-de-l-observatoire-national-du-suicide.html

[3] http ://oadi.education/presentation/

2

Analyse de sentiments

Dans cette section, nous nous intéressons aux méthodes d'analyse de sentiments pouvant être appliquées pour capturer l'humeur des personnes harcelées. On distingue les méthodes portant sur la polarité (e.g. positive, négative) et sur les émotions (joie, colère, ...) qui nous intéresserons plus particulièrement. Nous décrivons dans la suite trois catégories de méthodes : 1) **Analyse fréquentielle**, en rapport à la fréquence d'apparition d'un mot ; 2) **Génération de règles**, en produisant des règles qui associent un ensemble de mots à un sentiment ; 3) **Apprentissage**, pour apprendre des modèles permettant de prédire des classes de sentiments pour de nouvelles instances. Nous nous focalisons sur celles liées soit à la détection des maladies mentales, soit aux flux de textes.

2.1 Détection par analyse fréquentielle

Une des techniques les plus utilisées est la pondération $TF - IDF$ [29]. Pour trouver le(s) mot(s) significatif(s) dans un texte, on utilise d'une part l'occurrence du mot dans un texte (Term Frequency - TF) et d'autre part le nombre de documents contenant ce mot (Inverse Documents Frequency - IDF). Un mot très fréquent dans un texte, mais peu présent dans un ensemble de textes, ne sera pas significatif. À l'inverse, un mot très fréquent dans un texte mais présent que dans ce texte sera utilisé pour le décrire. Lors d'une analyse avec des données sous forme de flux comme dans notre étude, l'utilisation de $TF-IDF$ est difficile car le calcul de IDF nécessite de connaître le nombre de textes. Pour les flux de documents, [28] propose de remplacer IDF par ICF (Inverse Corpus Frequency - ICF), le calcul de ICF est le même que IDF mais en se basant sur le corpus statique et non sur l'intégralité des textes du flux. Cela a pour effet d'approximer la valeur obtenue par IDF tout en gardant un faible temps de calcul. [24] propose une autre approche fréquentielle grâce à *Linguistic Inquiry and Word Count* ($LIWC$), un méthode qui calcule le degré selon lequel les personnes utilisent différentes catégories de mots dans un ensemble de textes. Il permet aussi de classer les textes selon une émotion positive ou négative. Cette méthode a été utilisée dans [9] pour analyser les messages laissés sur Twitter avant un suicide. Pour finir, les émoticones, ou `smileys`, sont très utilisés dans les réseaux sociaux pour exprimer les émotions liées à la phrase ou au message écrit. Ils peuvent être utilisés comme un mot de la phrase. [10] propose un lexique d'émoticones, ce qui a pour résultat d'augmenter la précision des méthodes de classification. [27] montre qu'avec la prise en compte des émoticones, les règles construites par le système d'apprentissage se simplifient.

2.2 Détection par génération de règles

[31] présente une méthodologie pour rattacher un sentiment à un texte dans le cas d'une analyse en streaming. À partir de quelques exemples annotés, des règles d'associations sont générées pour créer un premier modèle. Ces règles sont stockées sous la forme de couple clé/valeur où la clé est la règle et la valeur est le triplet $\sigma(X)$,$\sigma(X \cup s_i)$ et $\theta(X \rightarrow s_i)$ qui représentent respectivement le support de X (i.e. le nombre de règles ayant X en corps de règle), le support de X et s_i (i.e. le nombre de règles ayant X en corps de règle et s_i en tête de règle) et la confiance de $X \rightarrow s_i$ donnée par $\theta(X \rightarrow s_i) = \frac{\sigma(X \cup s_i)}{\sigma(X)}$. Grâce à ces valeurs, il est possible d'une part de vérifier rapidement si un nouveau texte correspond à un sentiment connu ou non, et d'autre part, de maintenir à jour le modèle en incrémentant les valeurs correspondantes et sans les recalculer constamment. Lors de la classification d'un nouveau texte, il est possible d'extraire de nouvelles règles pour mettre à jour le modèle. Celles-ci sont conservées si elles obtiennent une confiance supérieure à un seuil fixé par l'utilisateur. Un seuil de confiance élevé préserve des règles de haute qualité mais qui sont trop sélectives et trop peu souvent applicables.

2.3 Détection par apprentissage

Le principe des méthodes par apprentissage (`machine learning`) consiste à déduire un modèle à partir d'un corpus de textes donné en exemple dans lequel le système va apprendre des régularités. On distingue trois types de méthodes : 1) L'**Apprentissage supervisé** permet une classification des données à partir d'un corpus annoté (i.e. possédant l'information de la classe à laquelle le message appartient) qui permet de construire un modèle. Le système peut alors classer les nouveaux textes grâce à ce modèle. [1] identifie ainsi sur Twitter des messages à risques suicidaires ; 2) À l'inverse, un **Apprentissage non-supervisé** commence avec un ensemble non annotés. Le système identifie des structures récurrentes à partir desquelles un modèle est construit de manière à ce que les données considérées comme les plus similaires soient associées au sein d'un groupe homogène et qu'au contraire les données considérées comme différentes se retrouvent dans d'autres groupes distincts. Dans le cas d'une analyse de polarité, cette approche se montre efficace lors d'un classement en deux groupes (i.e. un cluster pour les messages positifs et un autre les négatifs) [18]. Cependant, l'apprentissage non-supervisé peut aussi s'appliquer à des émotions [13]. 3) Les approches d'**Apprentissage semi-supervisé** se basent sur l'utilisation d'un corpus annoté et d'un second non-annoté. Le principe de l'apprentissage semi-supervisé est de modifier ou de réorganiser les hypothèses effectuées sur le modèle à partir des données d'apprentissage [22].

Synthèse

Ces méthodes permettent d'associer un sentiment à un texte. Bien que performantes indépendamment, les meilleurs résultats sont généralement obtenus en les utilisant des approches hybrides comme [25]. Dans notre étude nous souhaiterons considérer l'évolution de ces concepts au cours du temps. Pour cela, nous nous intéresserons au `concept drift`. *Comment mesurer l'évolution, l'apparition d'un concept, par exemple, l'apparition ou la disparition du concept de défaut d'estime de soi au fil des messages d'un jeune harcelé ?*

3

Concept drift

3.1 Définitions

Concept drift

Dans cette partie seront énoncées les notions indispensables pour comprendre les différents aspects du **concept drift**. Ces définitions sont extraites et réécrites à partir de l'article de référence [8]. Pour illustrer ces définitions, elles seront accompagnées par un exemple tiré de la situation suivante : Exemple : On étudie les pratiques sportives intéressant les utilisateurs du réseau social Twitter. Initialement, le modèle distingue les utilisateurs qui suivent deux sports, à savoir le football et la natation.

Définition 1 (Variable cible)**.** *Lors d'un processus de classification, la* ***variable cible*** *Y est la donnée que l'on souhaite prédire.* Exemple : La variable cible est ici le sport que suit l'utilisateur.

Définition 2 (Concept)**.** *Un* ***concept*** *y désigne une modalité (parmi les c modalités) de la variable cible à laquelle les exemples peuvent être assignés.* Exemple : Deux concepts sont associés à la variable cible `sport` : football et natation.

Définition 3 (Exemple)**.** *Un* ***exemple*** *X désigne une instance. On différenciera un exemple d'apprentissage associé à un concept d'un exemple de test pour lequel le concept est à prédire.* Exemple : Un exemple d'apprentissage est un message écrit par une personne et associé au concept football et un exemple de test est un message pour lequel on n'a pas cette information.

Définition 4 (Feature)**.** *Un* ***feature*** *f est une caractéristique d'un exemple qui est utilisée pour l'apprentissage. On parlera du feature f_i pour désigner le i_{eme} feature.* Exemple : Le nombre de mots dans le message.

Définition 5 (Distribution)**.** *La* ***distribution*** *représente la répartition des probabilités des exemples dans les concepts au cours du temps. Soit X un exemple*[1] *et y un concept, alors $p(y|X)$ est la distribution de y sachant X (i.e. la probabilité que le concept y soit associé à l'exemple X) :*

$$p(y|X) = \frac{p(y) * p(X|y)}{P(X)}$$

[1] Un exemple est représenté par l'ensemble des features qui le définissent

Définition 6 (Attribution d'un exemple à un concept)**.** *L'attribution ou non d'un concept y à un exemple X dépend de la règle $p(y|X) \geq \delta$ où δ est un seuil fixé par l'utilisateur. Un seuil δ élevé permet de renforcer la cohérence d'un concept associé (forte corrélation entre l'exemple et le concept) mais sera plus difficile à atteindre. Cette règle a deux conséquences directes. Un exemple peut ne correspondre à aucun concept ($\forall y, p(y|X) < \delta$) et un exemple peut avoir plusieurs concepts associés.*

Définition 7 (Concept drift)**.** *Dans le cas d'un flux de données (stream), la classification peut se trouver erronée si la distribution de la variable cible est modifiée. Le* ***concept drift*** *entre les instants t_0 et t_1 peut être défini comme : $\exists X : p_{t_0}(y|X) \neq p_{t_1}(y|X)$ où $p_t(X, y)$ désigne la distribution au temps t entre l'ensemble X de données et la variable cible y.*

Définition 8 (Prédiction)**.** *Lors du traitement d'un nouvel exemple, le modèle* ***prédit*** *une valeur, notée $\widehat{y}$, pour la variable cible.*

Constitution d'un programme permettant d'analyser un concept drift

Dans un programme pour analyser un concept drift, on retrouve plusieurs fonctionnalités indispensables qui se répartissent en 4 modules : la **mémoire**, la **détection de changements** (de concepts), l'**apprentissage** et l'**estimation de pertes** (vérification des prédictions) [8]. Plusieurs approches sont possibles pour chaque module. Les exemples suivants suivront l'hypothèse d'exemples textuels, où les features sont des mots. De plus, le modèle sera constitué de règles associatives pour réaliser la classification ainsi que d'un seuil δ qui permet d'assurer une cohérence minimale entre un exemple et un concept (voir définition 6). Exemple : À un instant donné, le modèle commence à recevoir des messages qu'il n'arrive pas à classer (le seuil δ n'est ni atteint pour le concept football, ni pour natation). Il s'agit alors sûrement d'un nouveau concept. Le modèle doit détecter cela et modifier l'ensemble des règles pour traiter les messages liés au football, à la natation et à ce nouveau concept.

Définition 9 (Module mémoire)**.** *Permet la mémorisation des exemples. Ce module doit aussi gérer la suppression des règles devenues obsolètes en fonction du temps.* Exemple : En se plaçant dans le cas d'utilisation de règles associatives, un exemple lié aux deux concepts connus (football et natation) pourrait être :

$\{ballon\} \rightarrow football$	$\{piscine\} \rightarrow natation$
$\{passe, joueur, gardien\} \rightarrow football$	$\{bonnet\} \rightarrow natation$

Définition 10 (Module de détection de changement)**.** *Permet la détection de la présence d'un nouveau concept. Tant que ce module ne renvoit pas de réponse positive, le classifieur utilisera les concepts connus.* Exemple : Dans le cas où le modèle recevrait des messages : **"J'ai mes nouvelles roues !"**, **"La vitesse c'est trop bien"** et **"Mettre un casque c'est obligatoire"**. Si le modèle possède les règles ci-dessus, il ne peut en appliquer aucune. Il va alors détecter la présence d'un nouveau concept.

Définition 11 (Module apprentissage)**.** *Permet la prise en compte de nouvelles règles.* Exemple : Considérons qu'à partir de l'ensemble des nouveaux messages reçus, les règles suivantes sont apprises et ajoutées au modèle :

$\{roues\} \rightarrow y$	$\{vitesse\} \rightarrow y$	$\{casque\} \rightarrow y$

où y est un nouveau concept qui ne porte pour le moment pas de nom puisque le système ne peut inférer le sport associée. L'étiquette de *roller* sera éventuellement décrite par un humain. Il est important de préciser qu'il pourrait s'agir de plusieurs concepts différents apparaissant en même temps : roller, vélo, F1 ...

Définition 12 (Module d'estimation de pertes)**.** *Dans certains cas, le programme peut finalement connaître la vraie valeur de la donnée qu'il a prédit plus tôt.* Exemple : Considérons qu'au bout d'un certain temps, les utilisateurs indiquent le(s) sport(s) qu'ils suivent. On peut alors comparer les prédictions faites avec les vraies valeurs.

Définition 13 (Les différents types de concept drift)**.** *Selon la vitesse du changement, celui-ci est décrit comme [8] :* ***immédiat*** *(sudden/abrupt),* ***progressif*** *(incremental),* ***graduel*** *(gradual),* ***récurrent*** *(reoccuring concepts),* ***bruit*** *(outlier). Ce dernier n'est pas un vrai concept drift et doit, dans un comportement optimal, ne pas être pris en compte lors du processus (voir figure 3.1).*

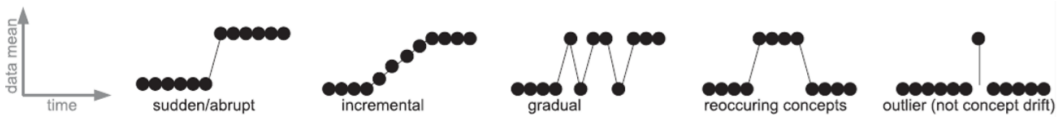

FIGURE 3.1 : Les différentes formes de Concept drift (extrait de [8])

Exemple : Le changement **abrupt** correspond à un utilisateur qui parle du jour au lendemain d'un nouveau sport sans jamais reparler de l'ancien. Le changement **incrémental** correspond au cas où l'utilisateur parle de plus en plus d'un nouveau sport et moins en moins de l'ancien. Le changement **graduel** représente un sport suivit la semaine et un second sport suivit uniquement lors des week-ends et/ou les vacances. Le changement **récurrent** est un utilisateur qui regarde la coupe du monde de football tout les 4 ans sans parler de football en dehors de cet évènement. Le **bruit** serait un message isolé parlant d'un autre sport.

3.2 Méthodes

Cette partie a pour but de présenter les différentes méthodes existantes pour la gestion du concept drift. Pour cela, le plan sera inspiré du classement proposé par [7]. Toutes les méthodes ne seront pas présentées par manque de place, seules les méthodes applicables à notre problématique seront introduites.

Module de mémorisation

La mémorisation d'informations lors d'une acquisition de données par un flux est difficile car, nous ne connaissons pas la taille du flux, ni même si il se terminera (flux de tweets). Il faut donc mettre en place des stratégies de mémorisation pour ne retenir que le minimum de messages nécessaires.

Gestion des données. La première contrainte est de déterminer le nombre d'exemples à garder. Pour cela, les méthodes créées peuvent se classer en deux catégories :1) **Exemple seul** Il est possible de ne garder qu'un seul exemple en mémoire. Cela implique d'adapter le modèle à chaque arrivée d'un exemple et de ne plus avoir accès aux exemples précédents ultérieurement. À chaque exemple X reçu à l'instant t, on prédit une valeur $\widehat{y}_t$. Une fois la vraie valeur de la variable cible y_t récupérée, on calcule la perte (voir partie 3.2) et on adapte le modèle. On retrouve ce principe dans l'algorithme $WINNOW$ [19] ; 2) **Exemples multiples** La seconde méthode pour mémoriser les exemples est de gérer une fenêtre contenant un certain nombre d'exemples. L'algorithme $FLORA$ a été un des premiers systèmes d'apprentissage incrémental supervisé et utilise un système de mémoire à plusieurs exemples [34]. Deux approches sont possibles pour définir les exemples mémorisés : 1) Mémoriser les n exemples les plus récents, ce qui permet d'utiliser les algorithmes avec le même nombre d'exemples à chaque utilisation. 2) Mémoriser les exemples récents durant une durée t. Cela implique de ne pas connaître le nombre d'exemples mais permet une mise à jour régulière du modèle.

Mécanisme d'oubli Afin d'améliorer les performances du modèle, il peut être nécessaire de supprimer certains exemples devenus obsolètes. La méthode s'appliquant *a priori* le mieux à ce stage est l'oubli progressif. Un coefficient est associé à chaque exemple, le plus souvent représentant son âge. Plus l'exemple est ancien, plus son coefficient est faible et donc l'exemple sera moins pertinent car ayant un poids plus faible. On retrouve cette approche dans [17].

Module Détection de changement

Cette partie présente le système qui valide ou non la présence d'un (ou plusieurs) nouveau(x) concept(s). Trois méthodes peuvent s'appliquer :

Maîtrise statistique des procédés - Statistical Process Control (SPC) Cette technique, initialement prévue pour contrôler la qualité d'une chaîne de production, est basée sur une analyse statistique. Pour un ensemble d'exemples, l'erreur est une variable aléatoire des essais ($trials$) de Bernouilli. La distribution polynomiale donne une forme générale de la probabilité alors qu'une variable aléatoire représente le nombre d'erreurs dans l'ensemble de n exemples [15]. Pour chaque temps t dans la séquence, le taux d'erreurs est la probabilité q_t d'observer une prédiction fausse ($\widehat{y}_t \neq y_t$) et une déviation standard $\sigma_t = \sqrt{\frac{q_t(1-q_t)}{t}}$. Le système gère deux valeurs q_{min} et σ_{min} durant l'exécution. Après avoir effectué la prédiction et avoir vérifié sa validité, on teste si $p_t + \sigma_t < q_{min} + \sigma_{min}$.

Gestion de deux distributions Ces méthodes reposent sur l'utilisation d'une fenêtre de taille fixe résumant les informations passées et d'une seconde fenêtre de taille dynamique contenant les exemples récents. La technique se base sur la comparaison statistique des distributions entre les deux fenêtres en admettant l'hypothèse nulle que les distributions sont égales. Si celle-ci est rejetée, cela implique la présence d'un nouveau concept au début de la fenêtre dynamique [12]. [33] présente une métrique basée sur l'entropie pour différencier les distributions entre les deux fenêtres. Un changement de concept est signalé lorsque la mesure passe en dessous d'un seuil fixé par l'utilisateur.

Approche contextuelle L'approche contextuelle se base sur l'utilisation du timestamp de l'exemple comme étant un feature à part entière. Pour cela deux étapes sont

réalisées : 1) Un arbre de décision est construit sur les exemples , si l'attribut temporel a été utilisé dans l'arbre de décision comme étant un élément discriminant entre les exemples, cela suppose plusieurs contextes ; 2) Dans un second temps, l'algorithme $C4.5$ [26] est utilisé pour déterminer les concepts intermédiaires (temporels). Cette idée est utilisé dans l'algorithme $Incremental\ Fuzzy\ Classification\ System$ ($IFCS$) [3].

Module Apprentissage

Une fois le changement de concept détecté, il est nécessaire de créer de nouvelles règles pour satisfaire les nouveaux exemples. Dans ce cas, il existe deux approches possibles : recréer le modèle ou l'adapter.

Ré-apprentissage du modèle Dans l'hypothèse où les exemples ont été stockés en mémoire, il est possible de réapprendre le modèle à chaque nouvel exemple. Pour cela, le modèle est oublié, le nouvel exemple fusionné avec le corpus d'anciens exemples, et un nouveau modèle est appris [14]. Deux autres approches permettent de modifier petit à petit le même modèle. La méthode **Incrémentale** traite les nouvelles données une à une et met à jour les données statistiques du modèle (résumé des données) à chaque fois [11]. La méthode **Online** effectue une mise à jour du modèle avec les exemples récents. La mise à jour du modèle s'effectue en fonction des erreurs de classement [19].

Méthodes adaptatives Afin de satisfaire au maximum le modèle, il est nécessaire de générer de nouvelles règles mais il n'est pas obligatoirement nécessaire de mettre à jour toutes les règles du modèle. Certaines approches mettent en place des méthodes pour réapprendre localement. **Remplacement global** : Lors d'utilisation de méthodes adaptatives (classifieurs discriminants par exemple), on construit entièrement le modèle lorsqu'un changement de distribution est détecté [6]. **Remplacement local** : Contrairement au remplacement global, si l'on détecte un changement dans une zone on remplace uniquement cette partie (sous-modèle). [4] utilisent cette méthode pour la classification de mails, où une catégorie de mail peut être modifiée et donc devoir être réapprise sans nécessairement nécessiter un réapprentissage de tous les autres concepts. Un changement de classification des mails de sports n'affectera pas nécessairement la classification des mails ne parlant pas de sport.

Pour gérer le modèle, deux approches sont également possibles : **Ensemble de modèles** La gestion de plusieurs modèles peut se catégoriser par trois approches qui ne sont cependant pas disjointes. Elles permettent, au contraire, de meilleurs résultats si plusieurs approches sont implémentées simultanément. La **gestion de modèles actifs de type "pool"** se base sur la création et la mémorisation de multiples modèles et en associe certains afin de s'adapter à la distribution actuelle [32]. La **sélection de modèles** qui effectue une mise à jour continue des règles d'apprentissages tels que les règles d'apprentissages sont soit recyclés dans un mode "batch" soit misent à jour 'online' en utilisant de nouvelles données [5]. Les **modèles pondérés** sont valorisés selon la cohérence entre leur distribution associée et la distribution réelle. Ainsi, plus un modèle sera pertinent plus son poids sera grand. Un modèle est actif si son poids est supérieur à un seuil fixé par l'utilisateur. Les règles d'apprentissages sont activées ou désactivées selon leur poids [16].

Concepts récurrents Comme cité dans la définition 13 (page 7), il existe un type appelé **récurrent** qui nécessite un traitement spécial car il peut avoir des conséquences

sur la quantité de calcul et de traitement. Ce type définit les concepts qui disparaissent et réapparaissent régulièrement. De part les mécanismes d'oubli, les modèles associés seront oubliés et devront être réappris lors de la réapparition du concept ayant pour conséquence le calcul, une nouvelle fois, des modèles associés. De plus, lors de l'apparition d'un nouveau concept, il est nécessaire de demander à un expert de l'étiqueter. Or, cela a été fait au moment de la première apparition du concept. [20] propose de garder en mémoire les anciens modèles et de ne pas les supprimer. Pour ne pas qu'ils interfèrent avec les modèles actuels, ils sont placés dans un second ensemble de modèles. Lorsqu'un changement de concept est détecté, les modèles mis de côté sont testés. Si un ancien concept est reconnu, il sera utilisé en tant que concept récurrent, sinon un nouveau concept est déclaré.

Module Estimation des pertes

Une approche très utilisée est la gestion de deux fenêtres : une première de petite taille contenant les exemples les plus récents et une plus grande contenant plus d'exemples récents (comprenant ceux de la petite fenêtre). Elles sont respectivement plus réactive et plus conservatrice. Quand un changement apparaît, il est reconnu en premier sur la fenêtre de petite taille. De façon semblable, il est possible de réaliser ce test avec des fenêtres glissantes (lorsqu'un nouvel exemple arrive, seulement le plus ancien est oublié). Basé sur cette hypothèse, [7] propose d'améliorer le test PH[23] avec le ratio de deux estimations d'erreurs : une estimation d'erreur à long terme sur la grande fenêtre (ou dans le cas d'une fenêtre temporelle, un coefficient proche de 1) et un second plus petit pour la deuxième fenêtre. Un changement est signalé lorsque le ratio de ces deux estimations croît significativement.

3.3 Évaluation

Définition 14 (Vraie et fausse alarme)**.** *Lorsque le module de détection de changement répond positivement, il se peut qu'il ne s'agisse pas réellement d'un concept drift mais d'un exemple isolé. Si un nouveau concept est détecté et qu'il existe réellement, on parle de* ***vraie alarme****, si il s'agit de bruit et que le module n'aurait pas dû répondre positivement, on parle de* ***fausse alarme****.*

[7] détermine différentes métriques et une méthodologie pour désigner la pertinence d'une approche. Toutes les mesures présentées ici peuvent être calculées grâce à des données synthétiques construites de manière à connaître les changements de concepts à l'avance. Afin de déterminer les performances de chaque approche, il est important de définir les notions sur lesquelles seront basées ces tests. Une première possibilité est d'utiliser les métriques traditionnelles que sont la **précision**, le **rappel** et l'**accuracy**.

		Valeurs réelles	
		Positif	Négatif
Valeurs prédites	Positif	VP	FP
	Négatif	FN	VN

- Précision : $\frac{VP}{VP+FP}$
- Rappel : $\frac{VP}{VP+FN}$
- Accurracy : $\frac{VP+VN}{VP+VN+FP+FN}$

Les mentions VP, FP, VN et FN représentent respectivement les **vrais positifs**, **faux positifs**, **vrais négatifs** et **faux négatifs**.

Certaines approches sont plus spécifiques aux problématiques du concept drift. Une première catégorie de métriques liées au concept drift est celle de l'apprentissage. Il est nécessaire de savoir mesurer le coût mémoire utilisée par le processus. Pour cela [2] propose l'unité de mesure $RAM - Hour$ qui correspond au nombre de GB de RAM utilisé par le processus en une heure. Une métrique complémentaire est le **Kappa-Statistic** qui est une mesure statistique pour déterminer la cohérence des concepts en prenant en compte le déséquilibre des concepts. Il est calculé par $\frac{a-a_r}{1-a_r}$, où a est la valeur d'accuracy d'un classifieur intelligent et a_r l'accuracy d'un classifieur aléatoire qui permute les prédictions du classifieur intelligent. Le résultat de Kappa-Statistic est une valeur entre 0 et 1, où 0 signifie que la précision atteinte est aléatoire.

Schéma d'expériences Une méthode utilisée habituellement pour comparer des approches est la **cross-validation**. Cependant, cette méthode n'est pas applicable ici car cela mélangerait l'ordre dans le temps des données. Une solution possible est de faire des "snapshots" à différents moments durant la création du modèle, autrement dit de récupérer à plusieurs instants toutes les données disponibles. Il est alors possible de voir les différentes étapes à la création du modèle.

Il existe deux approches possibles pour l'évaluation des techniques d'apprentissages supervisées adaptatives :1) **Holdout** Lorsque la taille de l'ensemble des données étudiées est grande, la **cross-validation** prend trop de temps, on mesure alors la performance sur un ensemble de données fixé. Cette méthode est surtout efficace lorsque la division entre l'apprentissage et les tests a été prédéfinie, de sorte que les résultats des différentes études puissent être directement comparés. Lors de l'essai d'un modèle à l'instant t, l'ensemble de maintien représente exactement le même contexte à ce moment t. Cependant, il n'est pas toujours possible d'utiliser cette approche car il n'est pas toujours possible de savoir avec certitude quels exemples appartiennent à un concept à l'instant t ; 2) **Prequential** Cette approche utilise les exemples courants en les appliquant au modèle avant d'adapter celui-ci. En effectuant les tests avant l'adaptation du modèle, on teste le modèle avec des situations jamais rencontrées auparavant. Ce système a l'avantage de n'utiliser aucun ensemble de maintien pour le test et utilise des données disponibles.

Performance par Benchmark

Pour effectuer une analyse comparative sur plusieurs méthodes, il faut utiliser des ensembles de données (datasets) et des implémentations logicielles des algorithmes à comparer. Il existe deux types de données : artificielles ou réelles. Dans les données artificielles, les concepts, changements de concepts sont connus. Toutefois, des données réelles sont également intéressantes car correspondant au monde réel. Il existe des données intermédiaires où les données sont initialement réelles mais certains changements de concepts sont forcés. On peut citer les jeux de données issues du framework MOA[2] : 1) **Synthétiques** : SEA Concepts Generator, STAGGER Concepts Generator, ... 2) **Réels** : Text Mining, Electricity, Email Spam, Business oriented, Games.

[2]http ://moa.cms.waikato.ac.nz/details/

4

Conclusion

4.1 Plan de stage

Les perspectives pour ce stage sont :

- **L'analyse de sentiments** : L'utilisation de données réelles (Données fournies par l'OADI notamment) pour apprendre des modèles est important afin d'utiliser des règles applicables aux exemples de test qui seront aussi des données réelles. Les travaux de [30] se montrent très intéressants mais en posant la problématique de l'ordre des mots et la distance entre les mots recherchés, les résultats peuvent être sûrement améliorés (Une phrase ayant les deux mots clés d'une règle mais où ces mots sont espacés de plusieurs mots ne sera pas reconnue, ie. très faible probabilité qu'il soit en lien dans la phrase). Enfin, il est intéressant de réaliser une analyse de sentiments hybride sur les données en associant les deux approches ci-dessus.

- **Concept drift** : Il est nécessaire de sélectionner pour chaque module les méthodes adaptées à la problématique du stage. Ainsi, le mécanisme d'oubli devra obligatoirement être **graduel** car l'âge d'un message est important (Un message posté la semaine dernière est plus important qu'un message âgé d'un an). Concernant la détection de changements, deux méthodes sont possibles : 1) La comparaison de deux distributions ; 2) Approche contextuelle (Si une personne présente une baisse de moral, il est probable que son comportement change et donc que les message soient différent après un instant précis)

  Les utilisateurs ne postant pas fréquemment (en comparaison avec la vitesse de traitement d'un ordinateur), il peut être intéressant d'utiliser une méthode d'apprentissage **incrémentale**. Quant à la gestion des modèles, il s'agit de combiner les approches **sélectives** et **pondérées** pour permettre une mise à jour facile des différents modèles. Enfin, il est très important que l'application prenne en compte les concepts récurrents afin de détecter le plus rapidement possible un état déjà atteint par une personne.

- **Application de l'étude** : Une application de cette étude serait de détecter des personnes à risque sur des réseaux sociaux tel que Facebook ou Twitter. De plus, les résultats de la détection de changement seront intégrés comme un indicateur dans une interface développée dans l'équipe et à destination des professionnels de santé (résultat d'un TER PRO en 2015).

Bibliographie

[1] Amayas Abboute, Yasser Boudjeriou, Gilles Entringer, Jérôme Azé, Sandra Bringay, and Pascal Poncelet. Mining twitter for suicide prevention. In *Natural Language Processing and Information Systems - 19th International Conference on Applications of Natural Language to Information Systems, NLDB 2014, Montpellier, France, June 18-20, 2014. Proceedings*, pages 250–253, 2014.

[2] Albert Bifet, Geoff Holmes, Bernhard Pfahringer, and Eibe Frank. Fast perceptron decision tree learning from evolving data streams. *Advances in Knowledge Discovery & Data Mining : 14th Pacific-Asia Conference, Pakdd 2010, Hyderabad, India, June 21-24, 2010. Proceedings. Part II*, page 299, 2010.

[3] A. Bouchachia. Fuzzy classification in dynamic environments. *Soft Computing*, 15(5) :1009 – 1022, 2011.

[4] José M Carmona-Cejudo, Manuel Baena-García, José del Campo-Ávila, Rafael Morales-Bueno, and Albert Bifet. Gnusmail : Open framework for on-line email classification. 2011.

[5] Alan Fern and Robert Givan. Online ensemble learning : An empirical study. *Machine Learning*, 53(1-2) :71–109, 2003.

[6] Joao Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with drift detection. In AnaL.C. Bazzan and Sofiane Labidi, editors, *Advances in Artificial Intelligence - SBIA 2004*, volume 3171 of *Lecture Notes in Computer Science*, pages 286–295. Springer Berlin Heidelberg, 2004.

[7] Joao Gama, Raquel Sebastiao, and Pedro Pereira Rodrigues. On evaluating stream learning algorithms. *Machine Learning*, 90(3) :317–346, 2013.

[8] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46(4) :44, 2014.

[9] John F Gunn and David Lester. Twitter postings and suicide : An analysis of the postings of a fatal suicide in the 24 hours prior to death. *Present tense*, 27(16) :42, 2012.

[10] Alexander Hogenboom, Daniella Bal, Flavius Frasincar, Malissa Bal, Franciska de Jong, and Uzay Kaymak. Exploiting emoticons in sentiment analysis. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 703–710. ACM, 2013.

[11] Geoff Hulten, Laurie Spencer, and Pedro Domingos. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 97–106. ACM, 2001.

[12] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. *Proceedings 2004 VLDB Conference*, pages 180 – 191, 2004.

[13] Sunghwan Mac Kim, Alessandro Valitutti, and Rafael A Calvo. Evaluation of unsupervised emotion models to textual affect recognition. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 62–70. Association for Computational Linguistics, 2010.

[14] Ralf Klinkenberg and Thorsten Joachims. Detecting concept drift with support vector machines. In *ICML*, pages 487–494, 2000.

[15] Ralf Klinkenberg and Ingrid Renz. Adaptive information filtering : Learning drifting concepts. In *Proc. of AAAI-98/ICML-98 workshop Learning for Text Categorization*, pages 33–40. Citeseer, 1998.

[16] Jeremy Z Kolter and M Maloof. Dynamic weighted majority : A new ensemble method for tracking concept drift. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 123–130. IEEE, 2003.

[17] Ivan Koychev and Ingo Schwab. Adaptation to drifting user's interests. In *Proceedings of ECML2000 Workshop : Machine Learning in New Information Age*, pages 39–46, 2000.

[18] Gang Li and Fei Liu. Sentiment analysis based on clustering : a framework in improving accuracy and recognizing neutral opinions. *Applied Intelligence*, 40(3) :441–452, 2014.

[19] Nick Littlestone. Learning quickly when irrelevant attributes abound : A new linear-threshold algorithm. *Machine learning*, 2(4) :285–318, 1988.

[20] Mohammad M Masud, Tahseen M Al-Khateeb, Latifur Khan, Charu Aggarwal, Jing Gao, Jiawei Han, and Bhavani Thuraisingham. Detecting recurring and novel classes in concept-drifting data streams. *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 1176–1181, 2011.

[21] Vinita Nahar, Sayan Unankard, Xue Li, and Chaoyi Pang. Sentiment analysis for effective detection of cyber bullying. *Web Technologies & Applications (9783642292521)*, page 767, 2012.

[22] Jonathan Ortigosa-Hernández, Juan Diego Rodríguez, Leandro Alzate, Manuel Lucania, Iñaki Inza, and Jose Antonio Lozano. Approaching Sentiment Analysis by Using Semi-supervised Learning of Multi-dimensional Classifiers. *Neurocomputing*, 92 :98–115, Septiembre 2012.

[23] ES Page. Continuous inspection schemes. *Biometrika*, pages 100–115, 1954.

[24] James W Pennebaker, Martha E Francis, and Roger J Booth. Linguistic inquiry and word count : Liwc 2001. *Mahway : Lawrence Erlbaum Associates*, 71 :2001, 2001.

[25] Rudy Prabowo and Mike Thelwall. Sentiment analysis : A combined approach. *Journal of Informetrics*, 3(2) :143 – 157, 2009.

[26] J Ross Quinlan. *C4. 5 : programs for machine learning*. Elsevier, 2014.

[27] Jonathon Read. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, pages 43–48. Association for Computational Linguistics, 2005.

[28] Joel W Reed, Yu Jiao, Thomas E Potok, Brian A Klump, Mark T Elmore, and Ali R Hurson. Tf-icf : A new term weighting scheme for clustering dynamic data streams. In *Machine Learning and Applications, 2006. ICMLA'06. 5th International Conference on*, pages 258–263. IEEE, 2006.

[29] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5) :513–523, 1988.

[30] Ismael S Silva, Janaína Gomide, Glívia AR Barbosa, Walter Santos, Adriano Veloso, Wagner Meira Jr, and Renato Ferreira. Observatorio da dengue : Surveillance based on twitter sentiment stream analysis. *XXVI Simpósio Brasileiro de Banco de Dados-Sessão de Demos*, 2011.

[31] Ismael Santana Silva, Janaína Gomide, Adriano Veloso, Wagner Meira, Jr., and Renato Ferreira. Effective sentiment stream analysis with self-augmenting training and demand-driven projection. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 475–484, New York, NY, USA, 2011. ACM.

[32] Alexey Tsymbal, Mykola Pechenizkiy, Pádraig Cunningham, and Seppo Puuronen. Dynamic integration of classifiers for handling concept drift. *Information Fusion*, 9(1) :56–68, 2008.

[33] Peter Vorburger and Abraham Bernstein. Entropy-based concept shift detection. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 1113–1118. IEEE, 2006.

[34] Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1) :69–101, 1996.

[35] Dawei Yin, Zhenzhen Xue, Liangjie Hong, Brian D Davison, April Kontostathis, and Lynne Edwards. Detection of harassment on web 2.0. *Proceedings of the Content Analysis in the WEB*, 2 :1–7, 2009.