Un algorithme *en ligne* pour la détection de nouveauté dans un flux de documents

Henri Binsztok, Patrick Gallinari

LIP6 – 8, rue du Capitaine Scott – 75015 Paris – {Henri.Binsztok,Patrick.Gallinari}@lip6.fr

Abstract

This paper deals with a recent task in Information Retrieval: Topic Detection & Tracking in streams of textual documents. Stakes are high especially in technological survey, with the uprising of huge quantities of data. Our proposal is based upon an incremental feature selection for on-line detection used to build event models. We obtain satisfying results while using few computational time.

Résumé

Cet article traite d'une tâche de recherche d'information qui est la détection et le suivi d'événements dans un flot de documents textuels. Avec la multiplication des flux et des sources d'information, les enjeux de cette tâche, notamment pour la veille technologique, économique, etc. sont considérables. Nous proposons d'améliorer des algorithmes utilisés pour la détection d'événement en combinant classification automatique et sélection de variables dans des algorithmes incrémentaux. Les expériences réalisées sur un corpus de grande taille montrent que les modèles proposés obtiennent de bonnes performances pour un faible temps de calcul.

Keywords: Recherche d'information - détection et suivi d'événements - TDT - Sélection de variables - Classification automatique incrémentale - Veille automatique

1. Introduction

L'information est plus que jamais devenue un facteur du développement économique, un moteur de l'innovation. Les entreprises cherchent à maintenir une connaissance actualisée de leur environnement global. La veille technologique, économique, financière, etc. est un des outils essentiels de cette connaissance. La multiplication de la quantité des sources d'information impose de développer des outils de traitement automatique pour la veille. La veille automatique comporte de nombreuses facettes. Parmi les problèmes soulevés par la veille, nous nous intéressons ici à l'analyse *en ligne* de flux d'information pour en extraire et analyser les événements relevant d'un ensemble de thématiques. Dans le domaine de l'informatique, une thématique pourra être les microprocesseurs et un événement la mise sur le marché d'un microprocesseur particulier à une date donnée. Dans le domaine de l'actualité, il existe de nombreuses sources émettant régulièrement des documents datés. C'est le cas par exemple des agences de presse (Reuters, AFP, etc.), où un nouveau document arrive toutes les cinq minutes environ. Dans ce cas, une thématique pourra correspondre à une des catégories classiquement utilisée en classification de documents (e.g. économie, bourse, etc.). Au sein de cette thématique, chaque document traite d'un événement de l'actualité (e.g. le rachat de Compaq par Hewlett-Packard est un événement). Nous nous intéressons à la détection automatique de ces événements pour un ou plusieurs flux d'information, et au suivi automatique des informations les concernant. Les événements ne sont bien sûr pas connus a priori, il peuvent survenir à n'importe quel moment

et leur "vie" peut être très variable. En terme de traitement automatique, il s'agit typiquement d'un problème d'apprentissage non supervisé où l'évolution temporelle de l'événement joue un rôle important. Les techniques que l'on cherche à développer doivent être génériques, car elles doivent pouvoir être utilisées sur de nombreuses sources d'information différentes, sans que cela donne lieu à des adaptations coûteuses. Les données peuvent comporter des informations multiples, les documents d'un même flux d'information peuvent être très hétérogènes à la fois dans leur forme et leur contenu.

Le programme TDT Un programme de recherche sur la veille automatique a été lancé aux États-Unis par la DARPA[1] en 1996. Ce programme, nommé TDT pour Topic Detection & Tracking, définit dans (Wayne, 1998) trois travaux de base pour la détection d'événements :

- S'il s'agit du flux continu (par exemple issu d'un journal télévisé retranscrit), il faut **segmenter** le flux de données pour le séparer en nouvelles indépendantes traitant chacune d'un seul événement.
- Les documents arrivant dans leur ordre chronologique, on cherche à **identifier** ceux qui traitent pour la première fois d'un événement : c'est la **détection**. Elle permet d'alerter un utilisateur lorsqu'un nouvel événement se produit dans l'actualité. Le cadre TDT distingue deux problématiques distinctes. Soit on cherche instantanément à savoir si une dépêche traite d'un événement nouveau : c'est la détection *en ligne*. Au contraire, on peut déterminer rétrospectivement si une dépêche traitait d'un événement nouveau : c'est la détection *hors ligne*.
- Enfin, une fois qu'un événement a été détecté, on cherche à **classer** les nouvelles se relatant à cet événement déjà connu : c'est le **suivi**. Il permet de constituer pour l'utilisateur des dossiers regroupant l'ensemble des documents traitant d'un événement.

Détection de nouvel événement Le travail que nous présentons traite de la détection *en ligne* de nouvel événement. Dans le cadre TDT, cette tâche apparaît comme la plus intéressante pour deux raisons :

- si la détection est parfaite, le suivi s'apparente à une tâche de classification, une des tâches principales de la recherche d'information (RI) qui est traitée largement dans la littérature,
- c'est la tâche la plus spécifique du TDT.

Pour la traiter, il faut résoudre une problématique de **recherche d'information** - comment représenter les documents et les événements - et une problématique d'**apprentissage** - quels algorithmes employer pour mener à bien la tâche de détection ? La section 2 fait un rapide état de l'art des algorithmes qui ont été proposés pour la détection et dans une moindre mesure le suivi. Dans la section 3, nous proposons un algorithme de modélisation incrémentale des événements. Nous verrons qu'il offre des performances satisfaisantes et l'avantage d'une complexité moindre par rapport aux méthodes existantes lors des expériences décrites dans la section 4.

2. État de l'art

(Allan et al., 1998) présente une synthèse des travaux réalisés pour la tâche TDT[2]. Plusieurs modèles ont été proposés quand cette tâche a été créée ; depuis ces travaux se sont focalisés sur les approches qui ont donné les meilleurs résultats. Les principaux problème à résoudre pour la détection sont : le développement de modèles non supervisés évoluant dans le temps, le caractère générique des modèles qui doivent s'adapter à des corpus très différents, la prise

[1] Defense Advanced Research Projects Agency, `http://www.darpa.mil`

[2] Ces travaux sont recensés sur `http://www.nist.gov/speech/publications/index.htm`

en compte d'informations très bruitées et l'aspect calculatoire. Les approches développées sont en général heuristiques car le problème lui-même a été peu formalisé. Les qualités des différents modèles sont jugées par leurs performances sur des bases de test développées au sein du projet TDT. Une des premières pistes explorée pour la détection (Allan et al., 1998) consiste à rechercher des changements brusques de vocabulaire dans le flux d'information. En particulier on peut penser qu'un nouvel événement sera souvent caractérisé par une nouvelle série d'entités nommées. Les essais réalisés montrent que cette information est pertinente, mais ne doit être utilisée qu'en appoint d'une autre méthode.

L'approche majoritaire utilise le modèle vectoriel qui est très largement employé en RI. A partir d'un codage vectoriel des documents et d'une fonction de similarité, on va essayer de détecter si un nouveau document traite d'un événement déjà présent dans le flux d'information ou introduit un nouvel événement. Pour cela on va utiliser un seuil sur la mesure de similarité. Deux grands types d'approche ont été proposées, l'une repose sur des algorithmes de classification incrémentale, et l'autre sur des techniques de plus proche voisin. Cette dernière classe de méthodes offre des performances sensiblement meilleures, la première a un coût moindre, et permet de réaliser simultanément détection et suivi. Ces deux raisons nous ont conduit à choisir ce type de méthode pour notre travail. L'approche qui illustre le mieux les travaux actuels est proposée par (Yang et al., 1998) puis (Yang et al., 1999). C'est un algorithme de classification simple, mais qui a l'immense avantage de fonctionner *en ligne*. Dans le formalisme du modèle vectoriel, les documents sont évalués séquentiellement, et les groupes croissent incrémentalement : un nouveau document est ajouté au groupe le plus proche si un seuil de similarité (noté t_c) est franchi, sinon un nouveau groupe est créé qui peut ou pas correspondre à un nouvel événement. La détection renvoie "Nouvel événement" si cette même mesure de similarité est inférieure à un second seuil t_n. Les expériences ont montré qu'en détection *en ligne*, il n'est pas souhaitable d'assigner plusieurs groupes à un même événement donc $t_c = t_n$.

Une troisième approche est constituée de modèles probabilistes, comme dans d'autres tâches de RI, les modèles et les performances sont assez similaires à ce que l'on obtient avec les modèles vectoriels.

Une des spécificités de la tâche TDT est que le temps y joue un rôle essentiel. Sa modélisation dans le cadre TDT est cependant complexe car l'évolution temporelle des événements présente une grande variabilité. Aussi, elle est uniquement prise en compte de façon rudimentaire dans les modèles actuels. Cette prise en compte est en général effectuée par le biais d'une fenêtre temporelle : les groupes ont une durée de vie limitée et les documents qui ont une date d'arrivée en dehors de cette fenêtre disparaissent du corpus, ce qui modifie en continu la composition des classes déjà détectée et donc le comportement de l'algorithme. Plusieurs options ont été considérées, notamment, des fenêtres triangulaires ou pondérées ont été utilisées à la place de l'approche binaire présentée ci-dessus. Toutefois, l'écart de performances n'est pas significatif. Nous avons retenu le principe d'une fenêtre rectangulaire pour les documents, un événement disparaissant dès que tous ses documents sont sortis de la fenêtre.

Un modèle classique de recherche d'information

Nous introduisons maintenant les notations utilisées dans ce travail et rappelons les bases du modèle vectoriel sur lequel nous nous appuyons. A un instant t, $D(t)$ et $E(t)$ sont respectivement les ensembles des documents et des évènements traités, leur cardinal sera respectivement noté $n(t)$ et $m(t)$. Chaque document est représenté par un ensemble de mots, chaque mot w_i ap-

partenant à un vocabulaire V. Chaque événement $e_k, k = 1 \cdots m(t)$, est constitué de l'union des documents qui lui sont rattachés. Dans le **modèle vectoriel** (Salton, 1968), chaque document d_i est représenté par un vecteur dans l'espace du vocabulaire V (où chaque mot du vocabulaire w_i constitue une dimension). Il reste alors à affecter un poids à chaque mot. Pour cela, on utilise deux estimateurs ; le premier est la fréquence de terme $\text{tf}(i,j)$ qui compte le nombre d'occurrences de w_i dans d_j. Le second, la fréquence de document $\text{df}(i)$ correspond au nombre de documents du corpus où apparaît w_i. Le codage des documents est effectué à l'aide de la mesure tf-idf[3], introduite par (Robertson and Jones, 1977), qui regroupe les estimateurs précédents. (Robertson et al., 1995) propose le codage suivant pour estimer la pertinence d'un mot dans un document :

$$\text{tfidf}(w_i, d_j) = \frac{\text{tf}(w_i, d_j)}{\text{tf}(w_i, d_j) + 0.5 + 1.5\frac{nl(d_j)}{\sum_{d_k} l(d_k)}} \frac{\log \frac{n}{\text{df}(w_i)}}{\log(n+1)},$$

où n et $l(d)$ désignent respectivement le nombre de documents du corpus à un instant donné et la longueur du document d. On peut alors estimer la similarité entre deux documents par la mesure cosine :

$$\cos(d_i, d_j) = \frac{\sum_l \delta_{i,l}\delta_{j,l}}{\sqrt{\sum_l \delta_{i,l}^2}\sqrt{\sum_l \delta_{j,l}^2}} \text{ avec } \delta_{i,l} = \text{tfidf}(w_l, d_i).$$

Malgré sa grande simplicité, ce codage est utilisé dans de nombreux systèmes de RI et constitue une référence. Ces mesures classiques sont définies pour des corpus fixés. Comme la détection *en ligne* est une tâche qui fait intervenir le temps, les comptages évoluent dans le temps, nous considérerons des mesures qui évoluent dans le temps, ce qui donne par exemple pour la fréquence de documents : $\text{df}(w_i) = \text{df}_t(w_i)$.

3. Algorithme de sélection *en ligne*

Nous allons maintenant présenter l'algorithme de détection que nous avons développé pour cette tâche. Cet algorithme suit une approche de classification automatique, que nous avons choisie pour des raisons de performance. Notre principale contribution consiste dans la proposition d'un algorithme qui réalise simultanément une sélection de caractéristiques et une classification. Cet algorithme réduit de façon considérable le temps d'exécution de la détection et semble bien adapté à la problématique *en ligne*. L'approche est, elle aussi, largement heuristique et sera évaluée sur un ensemble de test. Une sélection de variables est classiquement obtenue grâce à trois éléments : un critère d'évaluation permettant de comparer des sous-ensembles, un algorithme de parcours de l'espace des sous-ensembles possibles, ainsi qu'un critère d'arrêt. Dans la formulation usuelle du problème, on effectue une unique sélection de variables commune à l'ensemble des groupes. Le problème de la sélection de variables pour de la classification automatique *en ligne* avec un nombre de groupe qui évolue et des groupes dont les populations évoluent dans le temps n'a jamais été étudié à notre connaissance. C'est un problème difficile qui nous conduit à proposer une approche heuristique. Nous allons d'abord présenter la sélection de variables puis nous nous attacherons à l'algorithme de classification proprement dit.

[3] term frequency - inverse document frequency

3.1. Sélection et classification incrémentale

Pour le problème de détection *en ligne*, la modélisation des groupes doit évoluer avec le temps. Dans cette section, nous présentons un algorithme de sélection de variables et de classification incrémentaux qui recalculent les caractéristiques de l'ensemble des groupes pour chaque nouveau document. Cet algorithme de base est trop coûteux en pratique. Nous proposons ensuite en section 3.2 une modification de l'algorithme de sélection- classification qui ne modifie que le groupe auquel ce document est affecté - ou qui crée un nouveau groupe si le document correspond à un nouvel événement.

Sélection de variables par groupe Afin de permettre des mises à jour *en ligne*, nous avons choisi de sélectionner un ensemble de caractéristiques pour chaque groupe. Nous proposons un critère de sélection heuristique qui se base sur des fréquences de mots. Nous avons également testé un critère classique basé sur la mesure discriminante de Fisher, comme proposé par (Chakrabarti et al., 1997), mais les résultats peu satisfaisants nous ont conduit à abandonner cette approche. Pour tout événement e_k, soit $r(k)$ le nombre de variables utilisées pour la modélisation de e_k. Nous considérerons par la suite que $r(k) = r$ est une constante déterminée empiriquement. On se place à l'instant t, où $m(t)$ évènements ont été détectés, l'algorithme suivant permet la mise à jour des caractéristiques de l'ensemble des événements détectés, à l'instant t.

- Pour chaque événement $e_k, k = 1 \cdots m(t)$
  - Pour chaque mot w_i associé à l'événement e_k[4],
    - Calculer $\text{score}_k(w_i) = \text{etf}(w_i, e_k).\log \text{icf}(w_i)$, où $\text{etf}(w_i, e_k) = \sum_{d_j, e(d_j)=e_k} \text{tf}(w_i, d_j)$ (ou event term frequency) et $\text{icf}(w_i) = \frac{m(t)}{\text{ef}(w_i)}$ avec $m(t)$, nombre d'événements à l'instant t et $\text{ef}(w_i)$ (ou event frequency) le nombre d'événements contenant au moins un document dans lequel apparaît le terme w_i
  - Trier les valeurs $\text{score}_k(w_i)$ par ordre croissant,
  - Sélectionner les r mots de score le plus élevé pour constituer $M(e_k) = (w_{e_{k_1}} \cdots w_{e_{k_r}})$, sous-espace de V qui modélise e_k.

Le score utilisé exploite la même idée que le codage tf-idf. Une caractéristique est d'autant plus importante qu'elle est fortement représentée dans un groupe et absente dans les autres. Le but est d'identifier les mots qui sont spécifiques d'un événement. Cet algorithme heuristique est sous-optimal ; il s'agit d'un algorithme "en avant" sans remise en cause de la sélection effectuée. En effet, la sélection de variables qui est un problème combinatoire devient particulièrement délicate dans notre cas puisque la dimension de V se compte en dizaine de milliers. Pour étendre notre représentation des événements, il serait intéressant d'étudier par exemple la prise en compte des coocurrences proposée par (Ferret et al., 1997), qui utilise l'information présente dans le voisinage des mots pour mettre en évidence des descripteurs significatifs. Un codage tf-idf modifié pour intégrer l'écart-type des mots est utilisé. Également, (Pichon and Sébillot, 1999) propose un algorithme automatique pour désambiguïser le sens des mots sélectionnés dans le modèle d'un groupe en les rattachant à des thèmes.

[4] C'est-à-dire apparaissant dans au moins un document associé à cet événement

Algorithme de classification A partir de cette sélection de variable, on utilise un algorithme classique de classification de type k-moyenne qui utilise une mesure spécifique pour représenter les documents. On se place à l'instant $t, t > 0$ auquel arrive un nouveau document d_i. L'algorithme de classification qui fait intervenir un seuil de nouveauté t_n est le suivant :

Algorithme 1

- Pour chaque événement actif[5] $e_k, k = 1 \cdots m(t)$,
  - Calculer $M(e_k)$ à l'aide de l'algorithme de sélection
  - Calculer $\cos(e_k, d_i) = \cos(M(e_k), d_i)$

$$\cos(M(e_k), d_i) = \frac{\sum_{w \in M(e_k)} \mathsf{tf}(w, d_i)\mathsf{idf}(w, D)\mathsf{tf}(w, e_k)\mathsf{idf}(w, E_{|t})}{\sqrt{\sum_{w \in M(e_k)} (\mathsf{tf}(w, d_i)\mathsf{idf}(w, D))^2 \sum_{w \in M(e_k)} (\mathsf{tf}(w, e_k)\mathsf{idf}(w, E_{|t}))^2}}$$

- Soit l'événement le plus probable $e^* = \mathsf{argmax}_k \cos(M(e_k), d_i)$
- Si $\cos(e^*, d_i) > t_n$,
  - Affecter d_i à cet événement : $e(d_i) = e^*$
- Sinon,
  - Créer un nouvel événement $e_{m(t)+1}$
  - Affecter d_i comme premier document associé à $e_{m(t)+1}$: $e(d_i) = e_{m(t)+1}$

Un des problèmes des algorithmes de classification incrémentaux est qu'avec l'arrivée de nouveaux éléments ou le départ d'anciens éléments dû à la gestion du temps, les caractéristiques des groupes changent et les affectations des éléments aux différents groupes sont également susceptibles de changer. Nous avons pris le parti de ne pas revenir sur des affections de documents aux groupes, déjà effectuées. A un instant donné, la partition n'est donc pas optimale pour le corpus courant. Toutefois, une remise en cause de la partition serait peu compatible avec une exploitation en ligne. Tous les auteurs adoptent une démarche similaire, avec quelquefois des remises en cause partielles conduisant à l'éclatement de classes. Actuellement, ces choix reposent sur des heuristiques et les problèmes liés à cette gestion des évènements n'a pas été véritablement analysée.

3.2. Adaptation à la problématique en ligne

Les algorithmes précédents réévaluent à chaque arrivée d'un nouveau document les variables sélectionnées par la mesure etf $\log$ icf. Celle-ci prend en compte la répartition des mots sur l'ensemble des classes et est donc globale au corpus courant. Cette réévaluation est rédhibitoire pour une application *en ligne*. Différentes stratégies sont envisageables pour remédier à cela. Comme l'impact d'une réévaluation complète est limité, nous avons adopté une démarche incrémentale qui repose sur un calcul approché. Lors de l'ajout d'un document à un groupe donné, seul le modèle du groupe considéré sera réévalué. On conserve alors $M'(e_k)$, dernière estimation connue de $M(e_k)$. On obtient alors l'algorithme suivant lors de l'ajout d'un document d_i à l'instant t :

Algorithme 2

- Pour, chaque événement actif e_k,
  - Calculer $\cos(e_k, d_i) = \cos(M'(e_k), d_i)$, soit

$$\cos(M'(e_k), d_i) = \frac{\sum_{w \in M'(e_k)} \mathsf{tf}(w, d_i)\mathsf{idf}(w, D)\mathsf{tf}(w, e_k)\mathsf{idf}(w, E_{|t})}{\sqrt{\sum_{w \in M'(e_k)} (\mathsf{tf}(w, d_i)\mathsf{idf}(w, D))^2 \sum_{w \in M'(e_k)} (\mathsf{tf}(w, e_k)\mathsf{idf}(w, E_{|t}))^2}}$$

[5] L'intégration basique du temps reprise de (Yang et al., 1999) fait qu'un événement peut se désactiver s'il ne reçoit plus de documents : il sera alors définitivement ignoré par l'algorithme de classification.

- Soit l'événement le plus probable : $e^* = \text{argmax}_{e_k} \cos(M'(e_k), d_i)$
- Si $\cos(e^*, d_i) > t_n$,
  - Affecter d_i à cet événement : $e(d_i) = e^*$
  - Calculer $M'(e^*) = M(e^*)$ à l'aide de l'algorithme de sélection
- Sinon,
  - Créer un nouvel événement $e_{m(t)+1}$
  - Affecter d_i comme premier document associé à e_b : $e(d_i) = e_{m(t)+1}$
  - Calculer $M'(e_{m(t)+1}) = M(e_{m(t)+1})$ à l'aide de l'algorithme de sélection

L'étude théorique de cette modification est là aussi particulièrement complexe. Nous verrons dans les résultats expérimentaux que la démarche adoptée est raisonnable et performante. Nous allons discuter auparavant des avantages de notre démarche.

3.3. Avantages

Gain en complexité Comparons la complexité de notre démarche et celle de l'algorithme de référence proposé par (Yang et al., 1999), particulièrement coûteux en temps de calcul, lors de la détection de n documents. La complexité est exprimée en calculs de tf-idf, qui expérimentalement représentent 97% du temps de calcul. Soit le nombre de mots dans le vocabulaire à un instant donné $u(t) = \text{card } V_{|t}$; on rappelle que $m(t)$ représente le nombre d'événements à l'instant t. Pour ce calcul, le temps est compté en nombre de documents. On montre que la complexité de l'algorithme de référence est : $C_1 = \sum_{i=1}^{n} \sum_{k=1}^{m(i)} 4u(i)$. Par opposition, notre démarche a pour complexité : $C_2 = \sum_{i=1}^{n} \left[\left(\sum_{k=1}^{m(i)} 4r \right) + \left(\sum_{k=1}^{m(i)} y(e_k, i) z(e_k, i) \right) \right]$, où $y(e_k, i)$ représente la taille du vocabulaire des seuls documents présents dans l'événement e_k à l'instant i et $z(e_k, i)$ le nombre de documents associés à l'événement e_k à l'instant i. Sous quelques hypothèses, on montre qu'alors $C_1 \approx \frac{4}{3} nm|V|$, et qu'en introduisant γ_2, constante faisant intervenir le nombre moyen de documents par événement, $C_2 \approx \frac{\gamma_2}{m(t)} C_1$. Le coût de notre algorithme est donc une fraction de celui de l'algorithme de référence. Dans les expérimentations que nous avons réalisées, ce facteur est de l'ordre de $\frac{1}{10}$.

Interprétation des classes La sélection de variables que nous proposons repose sur l'identification d'un faible nombre de mots représentatifs d'un événement. Cet ensemble de mots permet d'interpréter les groupes trouvés par l'algorithme comme nous le verrons dans l'exposé des résultats.

4. Résultats

4.1. Mesures d'évaluation

Taux d'erreur et fausses alarmes Le système de détection renvoie une valeur booléenne {vrai ; faux} pour un couple $\{d_i, e(d_i)\}$, assorti d'une mesure de confiance. On a alors les mesures suivantes :

	EST positif	EST négatif
JUGÉ positif	a	b
JUGÉ négatif	c	d

L'évaluation repose alors sur deux indicateurs : le taux d'erreur $m = \frac{c}{a+c}$ (si $a + c > 0$, indéfini sinon) et le taux de fausse alarme $f = \frac{b}{b+d}$ (si $b + d > 0$, indéfini sinon) auxquels peuvent s'adjoindre des estimateurs plus classiques comme la précision $p = \frac{a}{a+b}$, et le rappel $r = \frac{a}{a+c}$. Il est possible d'agréger précision et rappel avec la mesure F, moyenne harmonique du rappel et de la précision : $F = \frac{2rp}{r+p}$.

Les algorithmes exposés précédemment utilisent un seuil sur la mesure de similarité pour détecter un nouvel événement. De façon triviale, si l'on augmente le seuil, le système de détection aura tendance a détecter plus d'événements. En faisant varier ce seuil, on change les performances obtenues pour les différentes mesures évoquées plus haut. Une des représentations classiques utilisée pour qualifier les systèmes de détection est la courbe Fausse Alarme / Taux d'Erreur (cf. figure 2).

4.2. Données disponibles

Nous avons constitué pour ces travaux un corpus (nommé `monde2`) créé à partir de Yahoo!News. Il s'agit, comme pour les corpus du programme américain TDT, de dépêches d'agence, mais en langue française. Il totalise 4819 dépêches d'agence issues de l'actualité internationale correspondant à une trentaine de thématiques et 113 événements. Chaque document contient les champs suivants : un titre, une date, un événement dont traite le document, et un corps de texte. A titre d'exemple, nous donnons dans le tableau 1 une partie des événements du corpus.

Accident de Kitzsteinhorn
Attentat contre le MI6 à Londres
Collision ferroviaire en Angleterre
Débat sur le bouclier antimissile
Détournement d'un avion russe à Istambul
Élections fédérales au Canada
Élections générales en Roumanie
Élections législatives en Thaïlande
Élections présidentielles au Ghana
Élections présidentielles au Sénégal
Élections présidentielles au Venezuela
Élections présidentielles aux États-Unis
Géorgie : la disparition de 3 membres du CICR

TAB. 1 – *Exemples d'événements de notre corpus, dont plusieurs représentatifs de la thématique Élections*

4.3. Performances

Valeur optimale de r Chaque événement est modélisé par r variables. Pour étudier l'influence de r, nous avons établi sur la figure 1 la courbe exprimant la mesure $F(r)$. Il est intéressant de noter qu'en ne prenant en compte que les 100 premiers documents du corpus, on obtient sensiblement le même résultat qu'avec 1000 documents. Ainsi, et par exemple dans le cas d'un apprentissage semi-supervisé, il est possible d'apprendre les principaux paramètres du modèle avec une faible quantité de documents étiquetés. La valeur de r apprise est faible : 11 variables par rapport aux 20000 mots du dictionnaire. Nous verrons qu'elle permet en outre de bien interpréter les résultats.

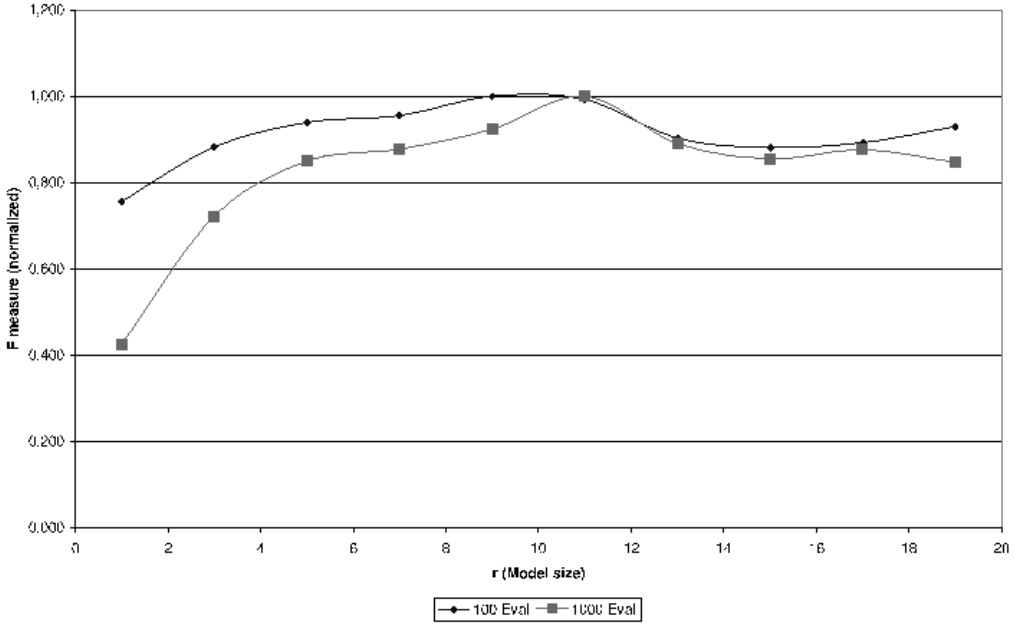

FIG. 1 – *Apprentissage d'une valeur optimale de r. Les losanges représentent la valeur obtenue avec 100 documents, et les carrés avec 1000 documents.*

Comparaison des méthodes En jouant sur le seuil d'acceptation d'un document, il est possible de choisir un point de fonctionnement sur une courbe Taux de Fausse Alarme / Taux d'Erreur. Il est donc important de déterminer quels sont nos besoins. Si le but est d'assister un opérateur par le système de détection, il est préférable d'avoir un taux d'erreur faible, même si un plus grand nombre de fausses alarmes demanderont confirmation à l'opérateur. Si, par contre, le but est d'obtenir un système automatisé ne générant que des alertes justifiées, nous privilégierons un faible taux de fausses alarmes au détriment d'un plus grand nombre d'erreurs. La figure 2 présente les performances de notre sélection incrémentale. Les performances sont globalement semblables au système de référence pour un coût dix fois moindre. Il est intéressant de remarquer que l'adaptation *en ligne* de la sélection de variables (Algorithme 2) dans l'algorithme de classification n'apporte aucune dégradation de performances supplémentaires par rapport à l'algorithme avec réévaluation complète (Algorithme 1).

4.4. Visualisation

La démarche adoptée fait que le modèle d'un événement est réévalué à chaque affectation d'un document à un événement. Par conséquent, dans le cadre d'une problématique *en ligne*, le modèle d'un événement évolue au cours du temps. Cette approche permet de visualiser les évolutions du vocabulaire liées au déroulement d'un événement. A titre d'exemple, la figure 3 présente pour un événement donné (Conflit dans les Balkans) l'évolution de la modélisation

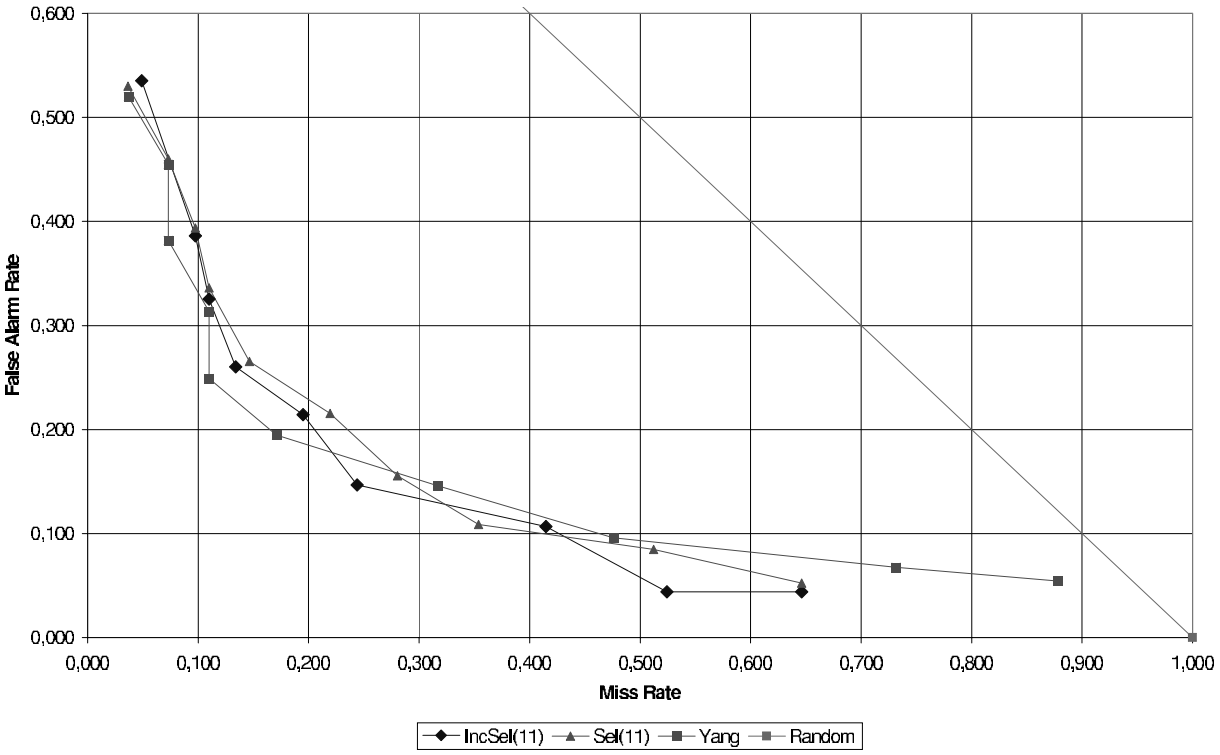

FIG. 2 – *Courbe Taux de Fausse Alarme/Taux d'Erreur de l'algorithme de sélection incrémentale. Les carrés représentent l'algorithme de référence, les losanges, notre algorithme avec sélection en ligne (Algorithme 2) et les triangles l'algorithme avec réévaluation complète (Algorithme 1).*

au cours du temps. On distingue ainsi les mots-clés toujours présents : Yougoslav[6], Serb, Alban ; des mots qui évoluent. Par exemple, UCPMB ou Bujanovac sont plus présents en début de conflit, mais s'effacent pour laisser place à Monténégro. On remarque qu'à chaque instant, le modèle semble bien décrire l'événement rattaché.

5. Perspectives

Ce travail ouvre plusieurs voies de recherche, qui peuvent s'organiser sur plusieurs grands axes : l'utilisation d'un meilleur modèle de RI, l'utilisation d'algorithmes de classification plus performants et une meilleure prise en compte du temps dans la tâche.

5.1. Quelques pistes pour la classification

Autres mesures de similarité L'algorithme proposé dans cet article dérive d'un algorithme de classification dans lequel la représentation d'une classe est celle des centroïdes. En effet, lors du calcul des modèles d'événements, c'est l'union des documents constitutifs d'un événement qui est prise en compte. Pourtant il existe différentes approches, classiques (Fukunaga, 1990), ou plus récentes (Tishby and Slonim, 2001). Nous avons vu que le centroïde peut devenir peu représentatif d'une classe qui évolue en permanence, il est donc important d'explorer ces alternatives.

[6]Les mots sont tronqués par l'utilisation d'un stemmer lors du prétraitement des documents

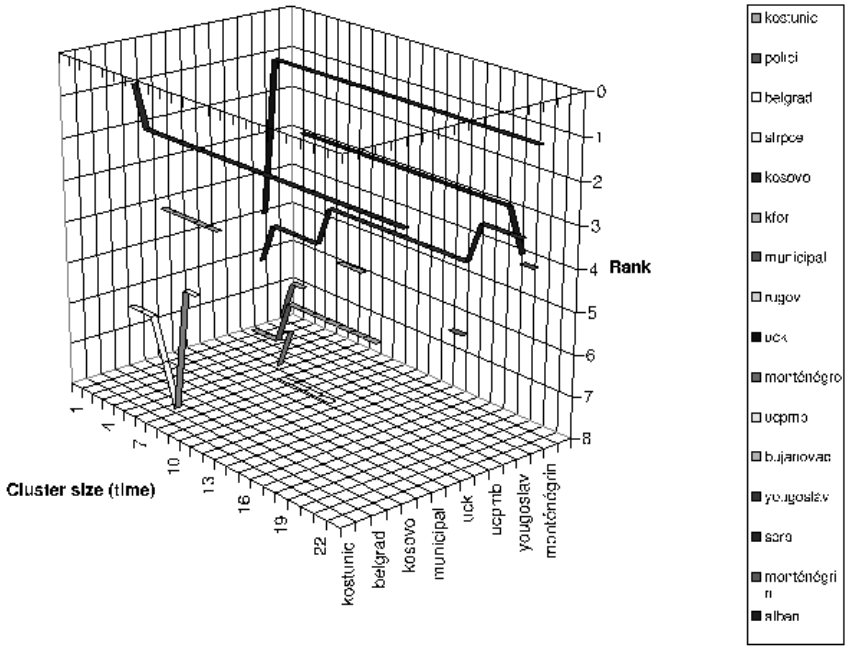

FIG. 3 – *Évolution des mots modélisant un événement dans le temps. L'axe vertical représente l'ordre du mot dans le modèle (1 signifiant mot le plus pertinent), les axes horizontaux représentant respectivement les mots sélectionnés et le temps compté en nombre de documents.*

Fusionnement et séparation Nous avons motivé la non remise en cause des classes lors du fonctionnement de l'algorithme par la spécificité de l'application *en ligne*. Toutefois, dans le contexte du suivi d'événements, il pourrait être souhaitable de fusionner deux groupes similaires, ou de séparer un groupe trop hétérogène. Bien sûr, plus le suivi est performant plus la détection est efficace, aussi ce point constitue-t'il une des principales extensions à développer pour notre modèle.

5.2. La détection, un problème temporel

La détection fait fortement intervenir le temps. Nous avons déjà commencé à introduire différentes mesures temporelles et à les utiliser notamment pour adapter la taille du modèle d'événement dans le temps. Les premiers résultats sont satisfaisants et ils constituent la suite immédiate de nos travaux.

6. Conclusion

Cet article a présenté une contribution algorithmique et expérimentale pour une tâche récente de RI : la détection de nouveauté dans des flux de données textuelles. Nous avons introduit un cas pratique, qui a servi de problématique à notre étude : dans un flux de dépêches d'agences,

comment détecter les documents qui traitent d'un événement nouveau ?

Nous avons basé notre approche sur un formalisme classique : le modèle vectoriel et le codage tf-idf pour effectuer une classification incrémentale. Nous avons proposé une approche permettant de modéliser les événements. Elle repose sur une sélection de variables et une classification incrémentales qui offrent plusieurs avantages :

– des performances au niveau de l'état de l'art,
– une possible Interaction Homme-Machine grâce à la visualisation des mots-clés associés à chacun des événements,
– un temps de calcul significativement inférieur à celui de l'algorithme de référence ce qui permet de façon effective une utilisation en ligne.

Nous remercions les reviewers anonymes pour leurs conseils et remarques.

Références

Allan J., Carbonell J., Doddington G., Yamron J., and Yang Y. (1998). Topic detection and tracking pilot study final report.

Chakrabarti S., Dom D., Agrawal R., and Raghavan P. (1997). Using taxonomy, discriminants and signatures for navigating in text databases. In *23rd VLDB Conference*, Athens, Greece.

Ferret O., Grau B., and Masson N. (1997). Utilisation d'un réseau de cooccurrences lexicales pour améliorer une analyse thématique fondée sur la distribution des mots.

Fukunaga K. (1990). *Introduction to Statistical Pattern Recognition, Second Edition*. Academic Press, Boston, MA.

Pichon R. and Sébillot P. (1999). Différencier les sens des mots à l'aide du thème et du contexte de leurs occurrences : une expérience. In *TALN*, Cargèse.

Robertson S., Walker S., and Jones S. (1995). Okapi at trec-3. In 3rd Annual Text REtrieval Conference. NIST.

Robertson S. E. and Jones K. S. (1977). Relevance Weighting of Search Terms. *Journal of the American Society for Information Science*, 27.

Salton G. (1968). Automatic information organization and retrieval.

Tishby N. and Slonim N. (2001). Data clustering by markovian relaxation via the information bottleneck method.

Wayne C. (1998). Topic detection and tracking (tdt) overview and perspective.

Yang Y., Carbonell J., Brown R., Pierce T., Archibald B., and Liu X. (1999). Learning approaches for detecting and tracking news events.

Yang Y., Pierce T., and Carbonell J. (1998). A study on retrospective and on-line event detection. In *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 28–36, Melbourne, AU.