DEFT2007

Actes du troisième DÉfi Fouille de Textes

Proceedings of the Third DEFT Workshop

3 juillet 2007

Grenoble, France

DEFT2007

Actes du troisième DÉfi Fouille de Textes

3 juillet 2007
Grenoble, France

Préface

Après le succès des éditions précédentes du défi fouille de texte (DEFT) consacrées à l'identification de locuteurs dans des discours politiques en 2005, et à la segmentation thématique de textes politiques, scientifiques et juridiques en 2006, une troisième édition a été programmée pour l'année 2007. Cette édition s'inscrit dans le cadre de la plate-forme de l'Afia (Association Française d'Intelligence Artificielle) organisée à Grenoble du 2 au 6 juillet 2007.

Le thème retenu cette année concerne la détection automatique de valeurs d'opinion dans des textes présentant des avis argumentés, positifs ou négatifs, sur un sujet donné.

La classification d'un corpus en classes pré-déterminées, et son corollaire le profilage de textes, est une problématique importante du domaine de la fouille de textes. Le but d'une classification est d'attribuer une classe à un objet textuel donné, en fonction d'un profil qui sera explicité ou non suivant la méthode de classification utilisée. Les applications sont variées et vont du filtrage de grands corpus pour faciliter la recherche d'information ou la veille scientifique et économique, à la classification par le genre de texte pour adapter les traitements linguistiques aux particularités d'un corpus.

La tâche que nous proposons vise le domaine applicatif de la prise de décision. Attribuer une classe à un texte, c'est aussi lui attribuer une valeur qui peut servir de critère dans un processus de décision. Et en effet, la classification d'un texte suivant l'opinion qu'il exprime a des implications notamment en étude de marchés. Certaines entreprises veulent désormais pouvoir analyser automatiquement si l'image que leur renvoie la presse est plutôt positive ou plutôt négative. Des centaines de produits sont évalués sur Internet par des professionnels ou des internautes sur des sites dédiés : quel jugement conclusif peut tirer de cette masse d'informations un consommateur, ou bien encore l'entreprise qui fabrique ce produit ? En dehors du marketing, une autre application possible concerne les articles d'une encyclopédie collaborative sur Internet comme Wikipédia : un article propose-t-il un jugement favorable ou défavorable, ou est-il plutôt neutre suivant en cela un principe fondateur de cette encyclopédie libre ?

Pour cette tâche, nous avons choisi des textes d'opinion venant de différents domaines :

- Média : les critiques de films, de livres, de spectacles et de jeux vidéo ;
- Articles scientifiques : les commentaires de révision d'articles de conférences ;
- Projets de loi : les interventions des parlementaires sur les projets de loi votés à l'Assemblée Nationale.

L'objectif de cette édition du défi vise à attribuer une valeur d'opinion pour chacun des documents composant ces corpus.

Remerciements

Le comité d'organisation de l'édition 2007 du défi remercie les organisateurs des années passées pour leurs conseils et l'aide qu'ils ont apportée dans le lancement de cette nouvelle édition.

Nous remercions également l'AFIA qui a bien voulu prendre en charge cet atelier dans le cadre de la plate-forme qu'elle organise cette année.

Nous présentons également nos remerciements les plus sincères aux sites Internet partenaires qui ont accepté de mettre à notre disposition le contenu de leurs activités (`www.avoir-alire.com` et `www.jeuxvideos.com`) ainsi qu'aux personnes qui nous ont fourni des relectures d'articles scientifiques.

Enfin, nous saluons chaleureusement chacune des équipes ayant participé à cette édition du défi et nous les remercions pour l'intérêt qu'elles ont manifesté à l'égard de cette campagne.

Le comité d'organisation de DEFT'07.

Sponsors

L'atelier DEFT'07 a été financé par l'AFIA.

AFIA
Association Française d'Intelligence Artificielle
LRI – Université Paris Sud – 91405 Orsay
http://afia.lri.fr/

LIMSI–CNRS
Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur
BP 133 – 91403 Orsay Cedex
http://www.limsi.fr/

LRI
Laboratoire de Recherche en Informatique
Université Paris Sud – 91405 Orsay
http://www.limsi.fr/

À voir, à lire
http://www.avoir-alire.com/

Jeuxvideos.com
http://www.jeuxvideos.com/

EGC
Association Extraction et Gestion des Connaissances
4, rue Jean Debay – 44000 Nantes
http://www.polytech.univ-nantes.fr/associationEGC/

Les actes ont été tirés à 30 exemplaires.

Comités

Comité de programme

Responsables : Benoît Habert (LIMSI-CNRS – LIR), Patrick Paroubek (LIMSI-CNRS – LIR) et Violaine Prince (LIRMM – TAL).

- Nathalie Aussenac-Gilles (IRIT) ;
- Catherine Berrut (CLIPS) ;
- Fabrice Clérot (France Telecom) ;
- Guillaume Cleuziou (LIFO) ;
- Béatrice Daille (LINA) ;
- Marc El-Bèze (LIA) ;
- Patrick Gallinari (LIP6) ;
- Éric Gaussier (Xerox Research) ;
- Thierry Hamon (LIPN) ;
- Fidélia Ibekwe-SanJuan (ELICO) ;
- Éric Laporte (IGM–LabInfo) ;
- Pascal Poncelet (LGI2P) ;
- Christian Rétoré (LABRI) ;
- Christophe Roche (LISTIC) ;
- Mathieu Roche (LIRMM) ;
- Pascale Sébillot (IRISA) ;
- Yannick Toussaint (LORIA) ;
- François Yvon (ENST).

Comité d'organisation

Responsables : Thomas Heitz (LRI – I&A) et Martine Hurault-Plantet (LIMSI-CNRS – LIR).

- Jean-Baptiste Berthelin (LIMSI-CNRS – LIR) ;
- Sarra El Ayari (LIMSI-CNRS – LIR) ;
- Cyril Grouin (LIMSI-CNRS – LIR) ;
- Michèle Jardino (LIMSI-CNRS – LIR) ;
- Zohra Khalis (Épigénomique) ;
- Michel Lastes (LIMSI-CNRS – AMIC), *webmestre*.

Table des matières

Programme

Mardi 3 juillet 2007

Session : présentation du défi approches statistiques

- **9h00 :** Présentation et résultats de DEFT'07. *Patrick Paroubek, Cyril Grouin et Martine Hurault-Plantet.*
- **9h30 :** La GRATOUNETTE : classification automatique générique de textes d'opinion. *Alejandro Acosta et André Bittar.*
- **10h00 :** Quel modèle pour détecter une opinion ? Trois propositions pour généraliser l'extraction d'une idée dans ce corpus. *Eric Charton et Rodrigo Acuna-Agost.*
- **10h30 :** Pause.

Session : approches statistiques, SVM

- **11h00 :** Approches naïves l'analyse d'opinion. *Eric Crestan, Stéphane Gigandet et Romain Vinot.*
- **11h30 :** Défi DEFT'07 : Comparaison d'approches pour la classification de textes d'opinion. *Michel Plantié, Gérard Dray et Matthieu Roche.*
- **12h00 :** Classification de textes et estimation probabiliste par Machine Vecteur de Support. *Anh-Phuc Trinh (LIP6).*
- **12h30 :** Pause déjeuner.

Session : approches hybrides, vocabulaire d'opinion

- **14h00 :** Défi : classification de textes subjectifs. *Michel Généreux et Marina Santini.*
- **14h30 :** Classification de textes d'opinions : une approche mixte n-grammes et sémantique. *Matthieu Vernier, Yann Mathet, François Rioult, Thierry Chamois, Stéphane Ferrari et Dominique Legallois.*
- **15h00 :** Classification d'opinions par méthodes symbolique, statistique et hyrbide. *Sigrid Maurel, Paolo Curtoni et Luca Dini.*
- **15h30 :** Comment faire pour que l'opinion forgée la sortie des urnes soit la bonne ? Application au défi DEFT 2007. *Juan Manuel Torres-Moreno, Marc El-Bèze, Frédéric Béchet et Nathalie Camelin.*
- **16h00 :** Pause.

Session : approches statistiques, LSA – Discussion

- **16h30 :** Le concours DEFT'07 envisagé du point de vue de l'Analyse de la Sémantique Latente (LSA). *Murat Ahat, Wolfgang Lenhart, Herbert Baier, Vigile Hoareau, Sandra Jhean-Larose et Guy Denhière.*
- **17h00 :** discussion générale sur DEFT'07 et DEFT'08 ;
- **18h00 :** Clôture.

Présentation et résultats

Présentation de DEFT'07 (DÉfi Fouille de Textes)

Les membres du Comité d'Organisation de DEFT'07 :
Cyril Grouin[1], Jean-Baptiste Berthelin[1], Sarra El Ayari[1], Thomas Heitz[2], Martine Hurault-Plantet[1], Michèle Jardino[1], Zohra Khalis[3] et Michel Lastes[1]

[1] LIMSI–CNRS,
{cyril.grouin, jean-baptiste.berthelin, sarra.elayari, martine.hurault-plantet, michel.lastes}@limsi.fr

[2] LRI, Université Paris-Sud,
heitz@lri.fr

[3] Épigénomique, Génopole d'Évry,
zkhalis@epigenomique.genopole.fr

Résumé : Le thème de cette édition du Défi Fouille de Textes est la classification de textes d'opinion. Pour réaliser ce défi, nous avons rassemblé quatre corpus venant de domaines différents, critiques de spectacles, tests de jeux, relectures d'articles scientifiques et débats sur des projets de loi. Dans cet article, nous présentons les corpus, ainsi que les pré-traitements de nettoyage que nous avons dû effectuer. Nous décrivons ensuite les tests manuels de la tâche de classification en valeurs d'opinion que avons effectués dans le but d'évaluer sa faisabilité sur nos corpus. Nous décrivons enfin les scores utilisés pour l'évaluation des résultats.

1 Introduction

Après le succès des éditions précédentes du défi fouille de texte (DEFT) consacrées à l'identification de locuteurs dans des discours politiques en 2005, et à la segmentation thématique de textes politiques, scientifiques et juridiques en 2006 (voir Azé *et al.* (2006)), une troisième édition a été programmée pour l'année 2007. Cette édition s'inscrit dans le cadre de la plate-forme de l'Afia (Association Française d'Intelligence Artificielle) organisée à Grenoble du 2 au 6 juillet 2007.

Le thème retenu cette année concerne l'attribution automatique de valeurs d'opinion à des textes présentant un avis argumenté, positif ou négatif, sur un sujet donné.

2 Présentation des corpus

Corpus « À voir, à lire »

Ce corpus comprend environ 3 000 documents (7,6 Mo), pour l'essentiel des critiques de livres, complétés par des critiques de films et de spectacles. Ces documents proviennent du site Internet www.avoir-alire.com. Trois valeurs d'opinion sont proposées pour ce corpus : favorable (classe 2), neutre (classe 1) et défavorable (classe 0).

Critiques de jeux vidéos

Ce corpus se compose d'environ 4 000 critiques de jeux vidéos (28,3 Mo) portant sur divers aspects du jeu (graphisme, jouabilité, durée, son, etc.) et provenant du site Internet www.jeuxvideos.com. Trois valeurs d'opinion sont proposées pour ce corpus : appréciation positive du jeu vidéo (classe 2), appréciation moyenne (classe 1) et appréciation négative (classe 0).

Relectures d'articles scientifiques

Ce corpus intègre environ 1 000 relectures d'articles (2,4 Mo) relatifs au domaine de l'Intelligence Artificielle. Ces relectures sont issues des conférences JADT[1], RFIA[2] et TALN[3]. Trois valeurs d'opinion sont proposées pour ce corpus : article accepté en l'état ou après modifications mineures (classe 2), article accepté après modifications majeures (classe 1) et article rejeté (classe 0).

Débats parlementaires

Ce corpus regroupe 28 832 interventions de Députés à l'Assemblée Nationale (38,1 Mo) extraites des débats portant sur la loi relative à l'énergie. Ces débats ont été aspirés depuis le site Internet de l'Assemblée Nationale[4]. Contrairement aux précédents corpus, seules deux valeurs d'opinion sont disponibles pour ce corpus : vote favorable à la loi en examen (classe 1) et vote défavorable à la loi en examen (classe 0).

Seuls les textes ont été retenus dans la composition des documents de chaque corpus, toute autre information (images, tableaux, etc.) ayant été supprimée.

Chaque corpus a été segmenté en corpus d'apprentissage et de test sur la base d'une répartition à 60 et 40%.

3 Préparation des données

3.1 Traitements spécifiques effectués sur chaque corpus

De manière générale et pour chaque corpus, des phases de nettoyage se sont révélées nécessaires (encodage des caractères en ISO Latin-1, éliminations des accents sur les caractères en dehors de l'ISO Latin-1 : « ō » devenant ainsi « o » dans « Tōkyō », homogénéisation des fins de ligne) puis de conversion des documents au format XML. Une DTD a été réalisée pour l'ensemble des corpus du défi.

Corpus « À voir, à lire » et « Jeux vidéos »

Les traitements ont été globalement les mêmes pour ces deux corpus. Tout d'abord il a fallu aspirer[5] les pages des critiques. Les pages HTML obtenues ont ensuite été analysées pour en extraire uniquement les textes des critiques et les notes qui leur sont associées.

Relectures

Les documents d'origine de ce corpus ont été rédigés dans des traitements de texte ou des tableurs. Ils ont été convertis au format texte brut avec conversion de l'encodage et élimination des formules mathématiques LaTeX. L'ensemble des documents a été anonymisé. Seules les relectures rédigées en français ont été conservées dans le corpus des JADT[6].

Débats parlementaires

Les compte-rendus des débats parlementaires ont été aspirés depuis le site Internet de l'Assemblée Nationale. Les questions au Gouvernement ont été manuellement retirées de ces comptes-rendus, qui ont ensuite été formatés en XML. Effectuer la correspondance entre l'intervention d'un locuteur et la valeur du vote de ce locuteur n'a pas posé de problème particulier étant donné que chaque compte-rendu de l'Assemblée Nationale reprend, en préambule, la liste des votes des parlementaires.

Pour éviter tout biais, le contenu des interventions a été anonymisé sur la base des noms de personnes (250 hommes politiques), de lieux (une dizaine de métonymies politiques : *l'Élysée, Matignon, Place Beauvau, rue de Grenelle*) et de partis politiques (*UMP, PS, droite, gauche, républicain, extrême droite*).

Seules les interventions de plus de 300 caractères ont été conservées pour le défi, les document en-deçà de ce seuil n'ayant pas été jugés exploitables après les tests réalisés auprès de juges humains.

[1] Journées internationales d'Analyse statistique des Données Textuelles.

[2] Reconnaissance des Formes et Intelligence Artificielle.

[3] Traitement Automatique des Langues Naturelles.

[4] L'intégralité des séances de débats sur ce projet de loi est accessible à l'adresse `http://www.assemblee-nationale.fr/12/debats/`

[5] Après accord des propriétaires des sites, bien évidemment.

[6] Ce corpus comprenant des relectures rédigées en anglais, en français ou en italien.

3.2 Principales difficultées rencontrées

Du fait de l'hétérogénéité des sources des différents documents composant nos corpus, nous avons dû faire face à plusieurs formats de documents (pages web, documents Word, tableaux Excel, fichiers en texte brut). Le premier « défi » qui s'est imposé a été celui de la conversion de l'ensemble de ces documents en fichiers exploitables pour la suite de la campagne. Il ne nous a pas toujours été donné de convertir automatiquement les documents, en particulier dans le cas des tableaux réalisés sous Excel.

Un second problème, fortement lié au point précédent, concerne les encodages de caractères et des fins de ligne. Les documents rédigés au moyen du traitement de textes Word intègrent notamment quelques caractères encodés en UTF-8 tels que : le symbole de l'euro (codé 200 en octal), les points de suspension (codés 205 en octal), la ligature « œ » (codée 234 en octal pour la version en minuscules et 214 pour la version en majuscules) et les guillemets simples « à l'anglaise » ouvrantes et fermantes (codées 221 et 222 en octal).

Malgré ces précautions, il reste probablement quelques coquilles dans nos corpus. Mais, après tout, cela fait partie des difficultés du traitement de la langue naturelle.

3.3 Évaluations manuelles des corpus

Chacun des corpus proposés dans le cadre de ce défi a auparavant été testé auprès de juges humains qui ont eu pour charge d'attribuer une valeur à quelques extraits des quatre corpus. Les résultats de chacun des juges ont été confrontés par le biais du coefficient κ (Kappa) de Cohen (1960) qui permet de mettre en évidence le taux d'accord entre deux juges[7].

Juge	**Réf.**	**1**	**2**
Réf.		0,17	0,12
1	0,17		0,03
2	0,12	0,03	

Juge	**Réf.**	**1**	**2**
Réf.		0,74	0,79
1	0,74		0,74
2	0,79	0,74	

FIG. 1 – Coefficient κ entre juges humains et la référence sur le corpus des jeux vidéos.
Échelle de notes de 0 à 20 (tableau de gauche) et de 0 à 2 (tableau de droite).

Les évaluations humaines ont permis de tester différentes échelles de notes. Les tableaux n° 1 donnent ainsi les coefficients κ obtenus par deux juges humains – entre eux et vis-à-vis de la référence – pour le corpus des jeux vidéos selon deux échelles de notes : une échelle large de 0 à 20 (notes d'origine) pour le tableau de gauche et une échelle restreinte de 0 à 2 pour le tableau de droite. Le changement d'échelle est le suivant : classe 0 de 0 à 9, classe 1 de 10 à 14 et classe 2 de 15 à 20.

Ces résultats démontrent qu'il y a un mauvais accord entre les juges sur l'échelle large (coefficient κ inférieur à 0,20) tandis que l'accord est qualifié de « bon » sur l'échelle restreinte (coefficient κ compris entre 0,61 et 0,80). Le mauvais accord entre juges sur l'échelle large s'explique par la dispersion des notes de 0 à 20.

Juge	**Réf.**	**1**	**2**	**3**	**4**	**5**
Réf.		0,10	0,29	0,39	0,46	0,47
1	0,10		0,37	0,49	0,48	0,35
2	0,29	0,37		0,36	0,30	0,43
3	0,39	0,49	0,36		0,49	0,54
4	0,46	0,48	0,30	0,49		0,60
5	0,47	0,35	0,43	0,54	0,60	

Juge	**Réf.**	**1**	**2**	**3**	**4**	**5**
Réf.		0,27	0,62	0,53	0,56	0,67
1	0,27		0,45	0,43	0,57	0,37
2	0,62	0,45		0,73	0,48	0,54
3	0,53	0,43	0,73		0,62	0,62
4	0,56	0,57	0,48	0,62		0,76
5	0,67	0,37	0,54	0,62	0,76	

FIG. 2 – Coefficient κ entre juges humains et la référence sur le corpus « *à voir, à lire* ».
Échelle de notes de 0 à 4 (tableau de gauche) et de 0 à 2 (tableau de droite).

Ces différences d'accord entre juges se retrouvent sur l'ensemble des corpus composant cette édition du défi. Les tableaux n° 2 renseignent des coefficients κ obtenus par cinq juges sur le corpus « *à voir, à lire* », pour deux échelles de notes : une échelle large (de 0 à 4) pour le tableau de gauche et une échelle restreinte (de 0 à 2) pour le

[7] L'accord entre deux juges est ainsi qualifié selon la valeur prise par le coefficient κ : excellent de 0,81 à 1,00 – bon de 0,61 à 0,80 – modéré de 0,41 à 0,60 – médiocre de 0,21 à 0,40 – mauvais de 0 à 0,20 – très mauvais en négatif.

tableau de droite. Le changement d'échelle est le suivant : classe 0 de 0 à 1, classe 1 pour la note 2 et classe 2 pour les notes 3 à 4.

À l'instar du corpus des jeux vidéos, les accords entre juges sur le corpus « *à voir, à lire* » sont meilleurs sur une échelle restreinte (de 0 à 2) que sur l'échelle large (de 0 à 4). Sur l'échelle large, les coefficients κ sont compris entre 0,10 et 0,60 (accords mauvais à modérés) tandis qu'ils s'échelonnent entre 0,27 et 0,76 (accords médiocres à bons) sur l'échelle restreinte.

Suite à ces évaluations manuelles, nous avons choisi d'utiliser des échelles restreintes pour l'ensemble des corpus du défi : une échelle de 0 à 2 pour les corpus « *à voir, à lire* », des jeux vidéos et des relectures, et une échelle de 0 à 1 pour le corpus des débats parlementaires (voir tableau n° 3).

	A voir, à lire	**Jeux vidéos**	**Relectures**	**Débats**
0	Mauvais	Mauvais	Article rejeté	Contre la loi
1	Moyen	Moyen	Article accepté après modifications majeures	Pour la loi
2	Bon	Bon	Article accepté en l'état ou après modifications mineures	

FIG. 3 – Valeurs associées à chaque classe selon les corpus.

D'autre part, les évaluateurs humains ont jugé la tâche plus facile sur les corpus des jeux vidéo et des débats parlementaires que pour le corpus « *à voir, à lire* ». Les coefficients κ sur les échelles restreintes sont également meilleurs pour les deux premiers corpus que pour le dernier ; ils sont compris entre 0,74 et 0,79 pour le corpus des jeux vidéos (bon accord), entre 0,60 et 0,80 pour le corpus des débats parlementaires (bon accord) et entre 0,27 et 0,76 pour le corpus « *à voir, à lire* » (donc, des accords médiocres à modérés).

3.4 Indice de confiance

Définition

Un système peut attribuer à un document une distribution de probabilité sur les différentes classes au lieu de lui attribuer une seule classe. L'indice de confiance est cette probabilité pour un document d'appartenir à une classe donnée.

Usage

L'indice de confiance introduit une pondération de la confiance et du rappel, donc du F-score. Il permet donc de comparer des classifieurs sur une base différente du « tout ou rien ».

Pertinence dans DEFT'07

Nous avons proposé aux concurrents un recours optionnel à cette variable.

Si l'on examine la situation dans les quatre corpus, on constate que son utilité n'est sans doute pas la même dans chacun des cas. Les critiques de films, par exemple, sont souvent identifiables assez nettement comme soit favorables, soit défavorables. Au contraire, dans le cas des relectures d'articles scientifiques, les documents sont plus difficiles à interpréter, et il semble alors légitime de pouvoir représenter le fait qu'un même jugement contient « du pour » et « du contre ».

De même, dans les débats parlementaires, certaines interventions ne se laissent pas facilement catégoriser comme purement favorables ou défavorables à un projet de loi, les attitudes mitigées sont possibles.

Les valeurs de l'indice de confiance peuvent traduire cette difficulté relative à choisir entre deux ou trois classes. Si elles avoisinent 0 ou 1, on est dans un cas tranché, si elles sont de l'ordre d'un demi ou d'un tiers, il s'agit d'un cas plutôt équivoque.

Parmi les concurrents, certains ont, dans un premier temps, mis en compétition plusieurs classifieurs dont ils disposaient. Ceux qui produisaient des jugements bien tranchés étaient préférés à ceux qui se montraient perplexes. C'est précisément grâce à leurs indices de confiance qu'une telle comparaison a pu s'effectuer.

4 Déroulement du défi

Les équipes ayant participé au défi sont au nombre de dix dont trois équipes constituées uniquement de jeunes chercheurs :

- **CELI France (Grenoble) :** Sigrid Maurel, Paolo Curtoni et Luca Dini ;
- **EPHE (Paris) et Universität Würzburg (Würzburg, Allemagne) :** Murat Ahat, Wolfgang Lenhard, Herbert Baier, Vigile Hoareau, Sandra Jhean-Larose et Guy Denhière ;
- **GREYC (Caen) :** Matthieu Vernier, Yann Mathet, François Rioult, Thierry Charnois, Stéphane Ferrari et Dominique Legallois ;
- **Lattice (Paris) :** Alejandro Acosta et André Bittar, *équipe jeunes chercheurs* ;
- **LGI2P (Nîmes) et LIRMM (Montpellier) :** Michel Plantié, Gérard Dray et Mathieu Roche ;
- **LIA (Avignon) :** Juan Manuel Torres-Moreno, Marc El-Bèze, Frédéric Béchet et Nathalie Camelin ;
- **LIA (Avignon) :** Éric Charton et Rodrigo Acuna-Agost, *équipe jeunes chercheurs* ;
- **LIP6 (Paris) :** Anh-Phuc Trinh, *équipe jeune chercheur* ;
- **NLTG–Université de Brighton (Royaume-Uni) :** Michel Généreux et Marina Santini ;
- **Yahoo ! Inc. (Paris) :** Eric Crestan, Stéphane Gigandet et Romain Vinot.

4.1 Organisation du défi

Corpus d'apprentissage

Les corpus d'apprentissage ont été diffusés à partir du 4 janvier 2007. Il a été autorisé aux différents participants d'utiliser des bases de connaissances. En revanche, nous avons exclu la possibilité d'utiliser des corpus d'apprentissages autres que ceux que nous avons fournis.

Nous donnons ci-dessous un extrait du corpus d'apprentissage des débats parlementaires (les passages anonymisés de ce corpus ont été remplacés par des balises) :

```
<DOCUMENT id="4:6">
 <EVALUATION nombre="1">
  <NOTE valeur="0" confiance="1.00" />
 </EVALUATION>
 <TEXTE>
 <![CDATA[
 Au nom de cette nouvelle gouvernance, vous affirmez la nécessité d'un grand
 nombre de réformes dans l'Etat, et notamment d'une nouvelle étape de la dé-
 centralisation. Les <partiPolitique /> qui, en 1982, avec <hommePolitique />
 et <hommePolitique />, ont élaboré et voté les grandes lois de décentrali-
 sation contre une <partiPolitique /> qui y voyait une menace contre l'unité
 de la République et un affaiblissement de l'Etat, ne peuvent que partager
 cette perspective. Sur ce socle, vous proposez d'organiser notre administra-
 tion selon un nouveau schéma. Pour une part, il s'agit de constitutionnali-
 ser une institution comme la région – qui pourrait sérieusement s'y opposer ?
 – de reconnaître le principe de l'autonomie financière des collectivités lo-
 cales, et d'introduire les référendums locaux : autant de thèmes sur lesquels
 nous pouvons converger. Enfin vous voulez faire droit au principe d'expéri-
 mentation. Nous y avons nous-mêmes recouru.
 ]]>
 </TEXTE>
</DOCUMENT>
```

Quatre équipes se sont désistées, trois avant la phase de tests, l'une pendant la phase de tests, ce qui constitue un taux d'abandon de 28,6%.

Corpus de test

La phase de tests a été conçue sous la forme d'une fenêtre de trois jours à définir dans un délai de deux semaines – du 19 au 30 mars 2007 –, les candidats ayant dès lors toute latitude pour choisir le premier jour du test dans cette période.

Les onze équipes ayant participé au test ont toutes choisi la deuxième semaine pour soumettre leurs résultats.

4.2 Évaluation des résultats

4.2.1 Définition du F-score utilisé pour le classement final

Chaque fichier de résultat a été évalué en calculant le F-score de chacun des corpus avec $\beta = 1$.

$$\frac{\text{F}_{\text{score}}(\beta) = (\beta^2 + 1) \times \text{Précision} \times \text{Rappel}}{\beta^2 \times \text{Précision} + \text{Rappel}}$$

Lorsque le F-score est utilisé pour évaluer la performance sur chacune des n classes d'une classification, les moyennes globales de la précision et du rappel sur l'ensemble des classes peuvent être évaluées de 2 manières (voir Nakache & Métais (2005)) :

- La micro-moyenne qui fait d'abord la somme des éléments du calcul – vrais positifs, faux positifs et négatifs – sur l'ensemble des n classes, pour calculer la précision et le rappel globaux ;
- La macro-moyenne qui calcule d'abord la précision et le rappel sur chaque classe i, puis en fait la moyenne sur les n classes.

Dans la micro-moyenne chaque classe compte proportionnellement au nombre d'éléments qu'elle comporte : une classe importante comptera davantage qu'une petite classe. Dans la macro-moyenne, chaque classe compte à égalité.

Micro-moyenne

$$\text{Précision} = \frac{\sum_{i=1}^{n} TPi}{\sum_{i=1}^{n}(TPi+FPi)} \qquad \text{Rappel} = \frac{\sum_{i=1}^{n} TPi}{\sum_{i=1}^{n}(TPi+FNi)}$$

Macro-moyenne

$$\text{Précision} = \frac{\sum_{i=1}^{n}\left(\frac{TPi}{(TPi+FPi)}\right)}{n} \qquad \text{Rappel} = \frac{\sum_{i=1}^{n}\left(\frac{TPi}{(TPi+FNi)}\right)}{n}$$

Avec :

- TPi = nombre de documents correctement attribués à la classe i ;
- FPi = nombre de documents faussement attribués à la classe i ;
- FNi = nombre de documents appartenant à la classe i et non retrouvés par le système ;
- n = nombre de classes.

Les classes d'opinion étant inégalement réparties dans les corpus, nous avons choisi de calculer le F-score global avec la macro-moyenne pour que les résultats sur chaque classe comptent de la même manière quelle que soit la taille de la classe.

Par ailleurs, dans la mesure où plusieurs classes peuvent être attribuées au même document avec des indices de confiance, nous avons établi les règles suivantes d'attribution d'une classe à un document pour le calcul du F-score strict.

Un document est attribué à la classe i si :

- Seule la classe i a été attribuée à ce document, sans indice de confiance spécifié ;
- La classe i a été attribuée à ce document avec un meilleur indice de confiance que les autres classes. S'il existe plusieurs classes possédant l'indice de confiance le plus élevé, alors nous retiendrons celle qui sera la première d'entre elles dans la balise `<EVALUATION>`.

Dans le calcul de ce F-score, l'indice de confiance n'est pris en compte que pour sélectionner la classe d'opinion attribuée à un document.

4.2.2 Définition du F-score pondéré par l'indice de confiance

Un système de classification automatique peut attribuer à un document une distribution de probabilité sur les différentes classes au lieu de lui attribuer une seule classe. L'indice de confiance est cette probabilité pour un document d'appartenir à une classe d'opinion donnée.

Le F-score pondéré par l'indice de confiance sera utilisé à titre indicatif pour des comparaisons complémentaires entre les méthodes mises en place par les équipes.

Dans le F-score pondéré, la précision et le rappel pour chaque classe sont pondérés par l'indice de confiance. Ce qui donne :

$$\text{Précision}_i = \frac{\sum_{\text{attribué correct. } i=1}^{\text{Nombre attribué correct. } i} \text{indice de confiance}_{\text{attribué correct.}_i}}{\sum_{\text{attribué } i=1}^{\text{Nombre attribué } i} \text{indice de confiance}_{\text{attribué}_i}}$$

$$\text{Rappel}_i = \frac{\sum_{\text{attribué correct. } i=1}^{\text{Nombre attribué correct. } i} \text{indice de confiance}_{\text{attribué correct.}_i}}{\text{nombre de documents appartenant à la classe } i}$$

Avec :
- Nombre attribué correct.$_i$: nombre de documents attribué correct.$_i$ appartenant effectivement à la classe i et auxquels le système a attribué un indice de confiance non nul pour cette classe ;
- Nombre attribué$_i$: nombre de documents attribués$_i$ auxquels le système a attribué un indice de confiance non nul pour la classe i.

Le F-score pondéré est ensuite calculé à l'aide des formules du F-score classique (voir section 4.2.1).

4.2.3 Algorithme utilisé pour désigner le vainqueur de DEFT'07

Les équipes ont été classées en fonction des rangs obtenus sur l'ensemble des corpus et en considérant chaque soumission comme atomique.

Le rang d'une soumission est donc égal à la somme des rangs associés au F-score classique de cette soumission sur chaque corpus. Ainsi, c'est le classement pour chaque corpus qui compte, et non les valeurs cumulées du F-score.

L'algorithme utilisé est présenté ci-dessous :

début
Pour chaque corpus (corpus ∈ {à voir à lire, jeux, relectures, débats}) **faire**
/ Score : liste qui associe à chaque couple (équipe, soumission) son F-score */*
Score(soumission, équipe) = F-score(corpus, soumission, équipe)
/ Tri de la liste Score dans l'ordre décroissant du F-score */*
Score trié(soumission, équipe) = tri(Score(soumission, équipe))
/ Tableau des rangs obtenus par chaque soumission de chaque équipe, pour le corpus considéré */*
Rang[corpus][soumission][équipe] = rang(Score trié(soumission, équipe))
fin Pour
Pour chaque équipe ayant soumis **faire**
/ Somme, sur tous les corpus, des rangs obtenus pour chaque soumission */*
Rang global[soumission][équipe] = $\sum_{\text{corpus}}$ rangs[corpus][soumission][équipe]
/ Choix de la meilleure soumission (rang le plus faible) */*
Rang[équipe] = $\min_{\text{soumission}}$(rangs[soumission][équipe])
fin Pour
/ Choix du vainqueur : équipe dont le rang est le plus faible */*
ÉquipeV telle que : Rang[ÉquipeV] = $\min_{\text{équipe}}$(Rang[équipe])
fin

FIG. 4 – Algorithme pour désigner le vainqueur

5 Conclusion

Cet article présente l'édition 2007 du défi fouille de textes dont l'objectif vise à attribuer automatiquement une classe à un texte d'opinion relevant de trois thématiques différentes (média, scientifique et juridique). L'ensemble des documents composant nos corpus est rédigé en français.

Après avoir exposé la tâche à réaliser et présenté les corpus utilisés pour ce défi, nous avons décrit les étapes de préparation des corpus en mettant l'accent sur les traitements spécifiques effectués sur chaque corpus et sur les problèmes rencontrés. Nous avons ensuite détaillé le déroulement du défi en présentant notamment la procédure d'évaluation des résultats et de classement des équipes.

Références

AZÉ J., HEITZ T., MELA A., MEZAOUR A.-D., PEINL P. & ROCHE M. (2006). Présentation de DEFT'06 (DÉfi Fouille de Textes). In *Actes de DEFT'06*.

COHEN J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, (20), 37–46.

NAKACHE D. & MÉTAIS E. (2005). Evaluation : nouvelle approche avec juges. In *INFORSID*, p. 555–570, Grenoble.

Résultats de l'édition 2007 du DÉfi Fouille de Textes

Les membres du Comité d'Organisation de DEFT'07 :
Jean-Baptiste Berthelin[1], Sarra El Ayari[1], Cyril Grouin[1], Thomas Heitz[2], Martine Hurault-Plantet[1], Michèle Jardino[1], Zohra Khalis[3] et Michel Lastes[1]

[1] LIMSI–CNRS,
{jean-baptiste.berthelin, sarra.elayari, cyril.grouin, martine.hurault-plantet, michel.lastes}@limsi.fr

[2] LRI, Université Paris-Sud,
heitz@lri.fr

[3] Épigénomique, Génopole d'Évry,
zkhalis@epigenomique.genopole.fr

Résumé : Cet article présente les résultats obtenus par chacun des participants à l'édition 2007 du Défi Fouille de Textes (DEFT). Ces résultats font apparaître une gradation des difficultés de traitement sur les différents corpus. Outre une vue d'ensemble des résultats, notre article décrit les méthodes retenues par les candidats lors des deux grands étapes du traitement : la *représentation* des textes et leur *classification*, pour laquelle les méthodes hybrides semblent prometteuses.

Mots-clés : F-score, rappel, précision, front de Pareto, tf*idf, représentation de textes, classification de textes.

Introduction

Pour cette édition du défi, chaque candidat avait la possibilité de soumettre jusqu'à trois résultats pour chacun des corpus. Chaque soumission a été considérée comme étant un ensemble indissociable portant sur les quatre corpus.

Pour toutes les soumissions, nous avons calculé le F-score strict (avec $\beta = 1$) puis, sur la base de ces calculs, nous avons défini la meilleure soumission de chaque équipe. Nous avons ensuite procédé au classement final des équipes en ne prenant en compte que la meilleure soumission de chacun des participants.

Les quatre corpus avaient été préalablement été soumis à des évaluations humaines, afin d'obtenir une approximation qualitative de la faisabilité d'une évaluation automatique. L'examen des résultats obtenus par les participants a été complété par celui des méthodes qu'ils ont employées, tant pour la représentation des textes que pour leur classification. Cette étude fait ressortir que la sélection des traits représentant chaque texte joue, dans ce cadre, un rôle crucial.

1 F-scores stricts

Au regard des résultats obtenus par chacun des participants sur chaque corpus (voir tableau n° 1), il apparaît assez nettement que les quatre corpus ont posé des problèmes distincts dans les traitements mis en œuvre. Nous pouvons ainsi établir un classement des corpus sur la base des F-scores obtenus, ces résultats traduisant les difficultés de traitement qu'ont rencontré les participants :

1. Corpus des jeux vidéos : les F-scores stricts des participants sont compris entre 0,784 et 0,457 ;
2. Corpus des débats parlementaires : les F-scores stricts sont compris entre 0,720 et 0,540 ;
3. Corpus « À voir, à lire » : les F-scores stricts sont compris entre 0,602 et 0,377 ;
4. Corpus des relectures : les F-scores stricts sont compris entre 0,566 et 0,398.

On observe les résultats les meilleurs pour les corpus des jeux vidéos et des débats parlementaires et, à l'inverse, de moins bons résultats pour les corpus des critiques et les relectures. Cette tendance semble partagée par l'ensemble des participants au défi comme l'attestent les graphiques n° 3 (F-scores stricts pour toutes les soumissions) et n° 4 (F-scores stricts pour les meilleures soumissions).

Équipe	Soumission	À voir, à lire	Jeux vidéos	Relectures	Débats
J.-M. Torres-Moreno (LIA)	1	0.602	0.784	0.564	0.719
J.-M. Torres-Moreno (LIA)	2	0.603	0.782	0.563	0.720
J.-M. Torres-Moreno (LIA)	3	0.603	0.743	0.566	0.709
G. Denhière (EPHE et U. Würzburg)	1	0.476	0.640	0.398	0.577
G. Denhière (EPHE et U. Würzburg)	2	0.599	0.699	0.507	0.681
S. Maurel (CELI France)	1	0.513	0.706	0.536	0.697
S. Maurel (CELI France)	2	0.418	0.538	0.477	0.697
S. Maurel (CELI France)	3	0.519	0.700	0.505	0.697
M. Vernier (GREYC)	1	0.577	0.761	0.414	0.673
M. Vernier (GREYC)	2	0.532	0.715	0.474	0.639
M. Vernier (GREYC)	3	0.532	0.715	0.474	0.673
E. Crestan (Yahoo ! Inc.)	1	0.529	0.670	0.441	0.652
E. Crestan (Yahoo ! Inc.)	2	0.523	0.673	0.462	0.703
M. Plantié (LGI2P et LIRMM)	1	0.421	0.783	0.478	0.618
M. Plantié (LGI2P et LIRMM)	2	0.424	0.732	0.442	0.671
M. Plantié (LGI2P et LIRMM)	3	0.472	0.547	0.442	0.608
A.-P. Trinh (LIP6)	1	0.542	0.659	0.427	0.676
A.-P. Trinh (LIP6)	2	0.490	0.580	0.467	0.665
M. Généreux (NLTG)	1	0.453	0.623	0.471	0.540
M. Généreux (NLTG)	2	0.464	0.626	0.463	0.554
M. Généreux (NLTG)	3	0.441	0.602	0.435	0.569
E. Charton (LIA)	1	0.377	0.619	0.433	0.616
E. Charton (LIA)	2	0.504	0.457	0.469	0.553
E. Charton (LIA)	3	0.504	0.619	0.419	0.553
A. Acosta (Lattice)	1	0.392	0.536	0.437	0.582

FIG. 1 – F-scores stricts ($\beta = 1$) pour toutes les soumissions sur chaque corpus.
La meilleure soumission de chaque équipe apparaît sur une ligne grisée.

Outre le fait que cette gradation de difficulté sur les différents corpus apparaît partagée par l'ensemble des participants au défi, nous remarquons également que ces résultats rejoignent les évaluations opérées par les juges humains (voir tableaux n° 2) :

1. Corpus des jeux vidéos : les F-scores stricts des juges humains sont compris entre 0,90 et 0,73 ;
2. Corpus « À voir, à lire » : les F-scores stricts sont compris entre 0,79 et 0,52 ;
3. Corpus des relectures : les F-scores stricts sont compris entre 0,58 et 0,41.

Les évaluateurs humains ont obtenu de meilleurs résultats sur les corpus des jeux vidéos et « *à voir, à lire* » que les systèmes automatiques des participants au défi. En revanche, les résultats sont quasi-identiques entre juges humains et systèmes automatiques sur le corpus des relectures, corpus jugé complexe par les humains.

Juge	1	2	3
F-score	0,73	0,86	0,90

Juge	1	2	3	4	5
F-score	0,52	0,76	0,69	0,70	0,79

Juge	1	2
F-score	0,41	0,58

FIG. 2 – F-scores obtenus par les juges humains sur les corpus « *à voir, à lire* » (tableau de gauche), des jeux vidéos (tableau central) et des relectures (tableau de droite).

Des méthodes d'analyse distinctes pour chaque type de corpus

Du fait de l'existence de corpus de différentes qualités littéraires (des phrases bien formulées dans les débats parlementaires aux phrases courtes et mal accentuées des relectures d'articles), des méthodes d'analyses distinctes ont été appliquées sur chaque corpus. Ces différences de méthodes ressortent dans les courbes des graphiques des F-scores stricts.

Si l'on considère qu'il existe une thématique « critiques » rassemblant les corpus des jeux vidéos et « *à voir, à lire* » (autrement dit, les critiques de livres et de films), il semblerait – d'après le graphique n° 3 – que les candidats

ont chacun appliqué la même méthode sur ces deux corpus ; pour une soumission donnée, on trouvera ainsi le même type de résultats (bon ou mauvais) pour ces deux corpus. Il en résulte que ces deux courbes évoluent globalement en parallèle sur les deux graphiques.

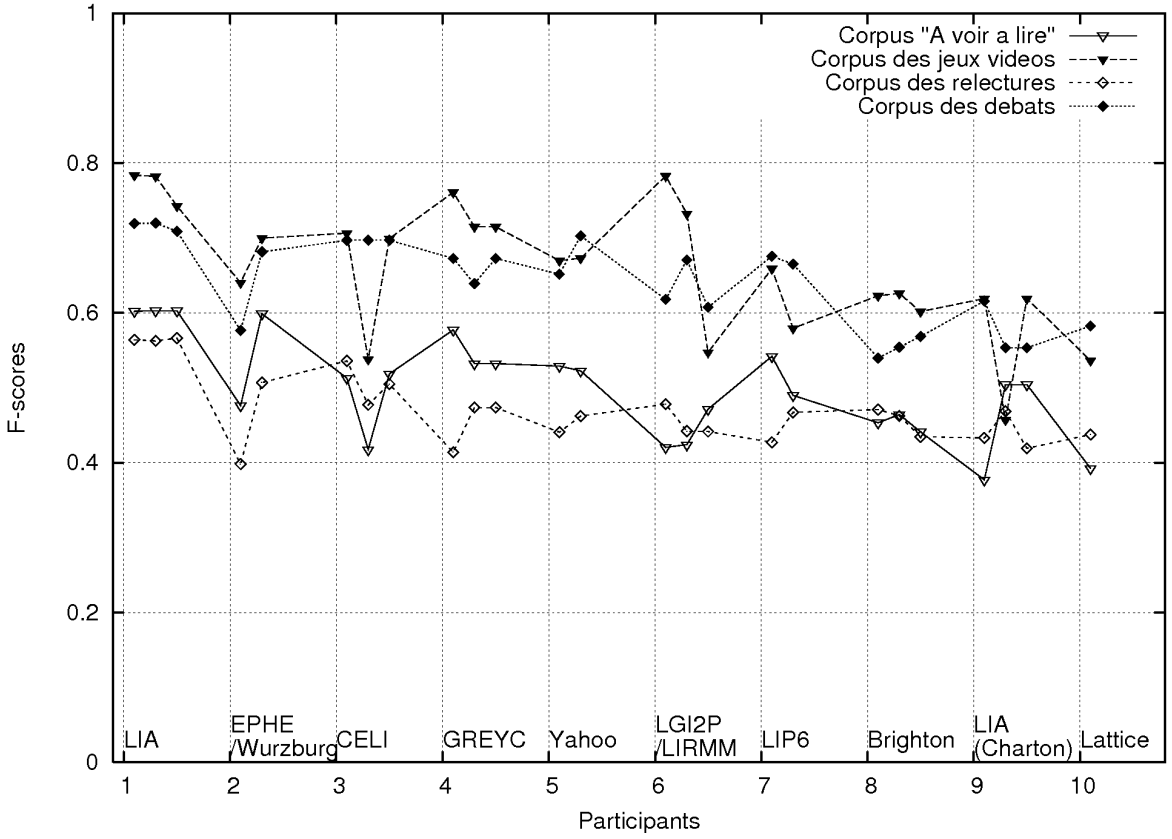


FIG. 3 – F-score strict ($\beta = 1$) pour l'ensemble des soumissions de chacun des candidats.

Un corpus difficile : les relectures d'articles

Malgré les difficultés rencontrées sur le corpus des relectures, quelques équipes semblent avoir eu moins de difficultés pour ce corpus que pour celui des critiques de livres et de films. Il en est ainsi pour deux soumissions du CELI, mais également de l'équipe LGI2P/LIRMM, de l'équipe de Michel Généreux (noté « Brighton »), et de deux des trois équipes jeunes chercheurs : une soumission pour Eric Charton et la soumission du Lattice.

Un corpus apprécié des équipes jeunes chercheurs : les débats parlementaires

Si l'on considère les équipes jeunes chercheurs indépendamment des autres équipes, une singularité émerge quant au corpus des débats parlementaires. Alors que les meilleurs résultats ont été obtenus sur le corpus des jeux vidéos, les équipes de jeunes chercheurs (notées « LIP6 », « LIA (Charton) » et « Lattice » en légende des graphiques) ont obtenu leurs meilleurs résultats sur le corpus des débats parlementaires.

Pour le cas où ces équipes auraient soumis plusieurs résultats, la meilleure soumission de chacune de ces équipes demeure celle où les résultats sur le corpus des débats parlementaires sont les meilleurs. Cette constatation s'avère assez flagrante sur le graphique n° 4. À ce titre, l'équipe « Yahoo » est la seule équipe hors catégorie « jeunes chercheurs » à avoir pour meilleure soumission celle où les résultats obtenus sur le corpus des débats parlementaires sont les plus élevés.

L'incidence de l'indice de confiance sur les résultats

Les participants ont eu la possibilité d'associer un indice de confiance à chaque note attribuée aux documents des corpus. Cet indice de confiance était proposé de manière optionnelle.

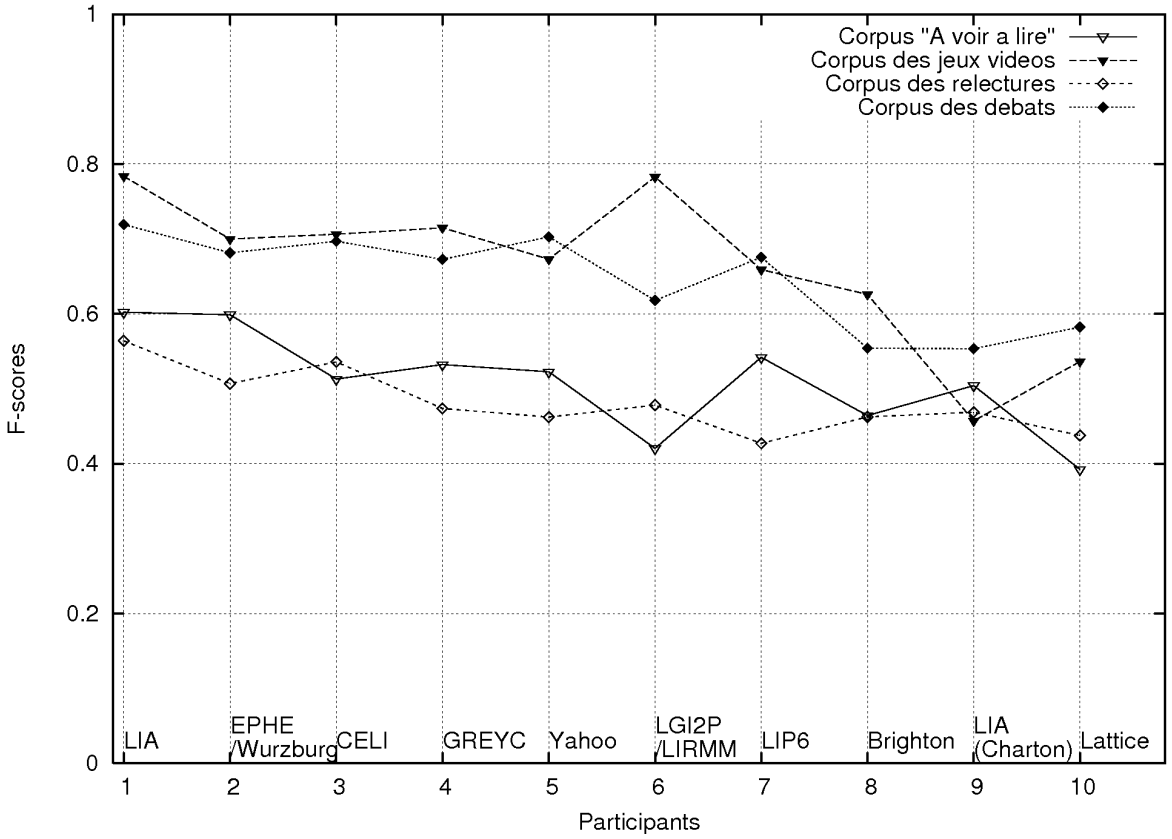


FIG. 4 – F-score strict ($\beta = 1$) pour les meilleures soumissions de chacun des candidats.

Sur les dix participants au défi, six y ont recouru. Sur ces six participants, certains l'ont appliquée pour chaque soumission, d'autres n'ont proposé que certaines soumissions avec indice de confiance (en générale deux soumissions avec indice de confiance, une soumission sans) :

- **Soumissions avec indice de confiance :** LIA (soumissions n° 1, 2 et 3), CELI France (n° 1, 2 et 3), GREYC (n° 1 uniquement), LGI2P/LIRMM (n° 1 et 3), LIP6 (n° 1 et 2) et M. Généreux (n° 1, 2 et 3) ;
- **Soumissions sans indice de confiance :** LPC/UP8 (soumissions n° 1 et 2), GREYC (n° 2 et 3), Yahoo ! Inc. (n° 1 et 2), LGI2P/LIRMM (n° 2 uniquement), E. Charton (n° 1, 2 et 3) et Lattice (n° 1).

Les résultats ne nous permettent pas d'établir une corrélation entre les scores obtenus et l'utilisation de l'indice de confiance dans les notes attribuées.

2 Front de Pareto

Définition

Le front de Pareto est défini par l'ensemble des approches qui sont telles qu'aucune autre approche ne présente de meilleurs résultats pour tous les critères étudiés, en l'occurrence le rappel et la précision.

Représentation graphique

Le rappel est présenté sur l'axe des abscisses, la précision sur l'axe des ordonnées. Les courbes correspondent aux valeurs de F-score comprises entre 0,1 et 0,9 (avec $\beta = 1$).

Le front de Pareto est symbolisé sur ces schémas par l'ensemble des points qui sont reliés par des tirets. Les points isolés sont donc exclus du front de Pareto.

Les numéros aux côtés des points permettent d'identifier les équipes, un point représentant une soumission pour le corpus considéré (notez que le numéro de la soumission n'apparaît pas sur ces schémas) :

Numéro	Équipe
3	M. Généreux (NLTG–Université de Brighton)
4	M. Plantié (LGI2P et LIRMM)
5	G. Denhière (LPC–Université de Provence et Université Paris 8)
6	M. Vernier (GREYC)
7	E. Crestan (Yahoo ! Inc.)
8	A.-P ; Trinh (LIP6), *équipe jeunes chercheurs*
9	A. Acosta (Lattice), *équipe jeunes chercheurs*
11	J.-M. Torres-Moreno (LIA)
13	S. Maurel (CELI France)
14	E. Charton (LIA), *équipe jeunes chercheurs*

Une difficulté partagée sur certains corpus

L'analyse des histogrammes n° 5 à 8 met en avant plusieurs éléments. En premier lieu, la difficulté partagée par l'ensemble des participants sur certains corpus, en particulier celui des relectures (figure n° 8) pour lequel les résultats sont moins bons que pour les autres corpus, avec des valeurs de rappel, de précision et de F-score strict qui s'échelonnent entre 0,4 et 0,6 sans trop de disparités entre chaque candidat.

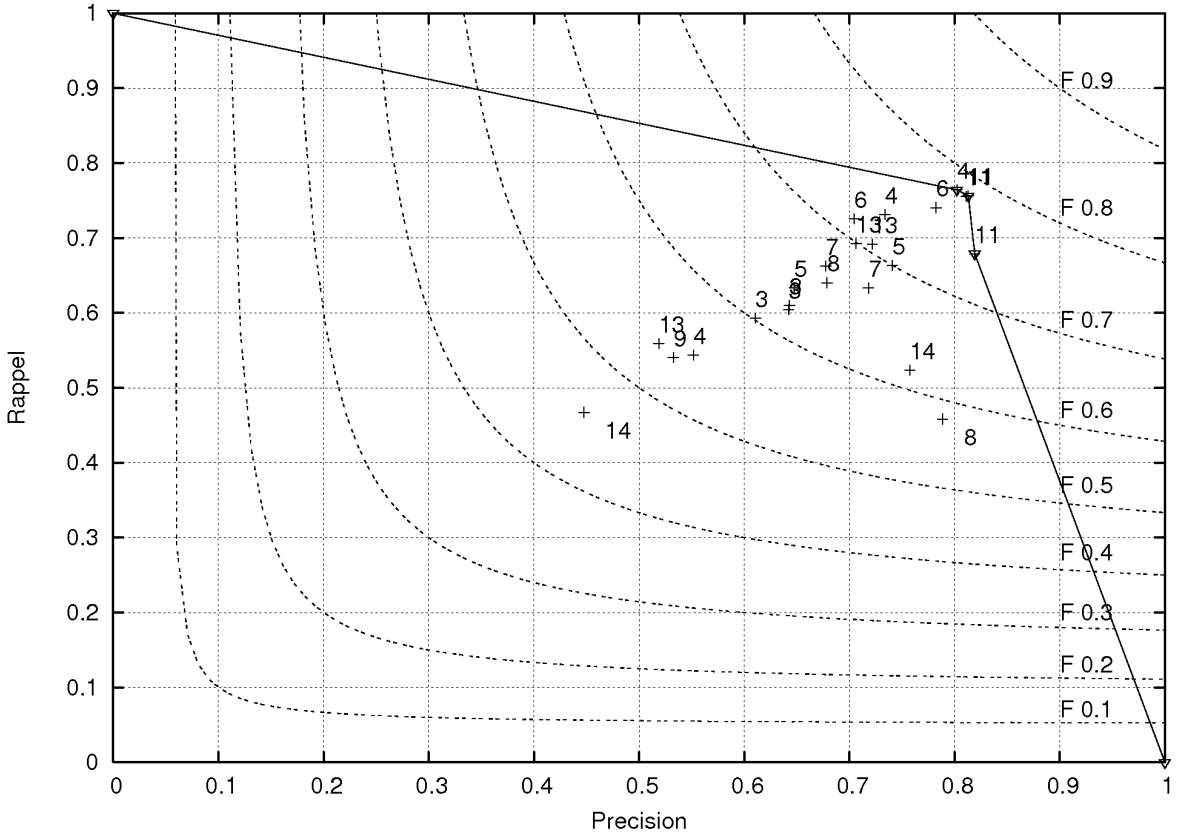


FIG. 5 – Front de Pareto pour le corpus des jeux vidéos.

A contrario, l'histogramme n° 5 confirme la bonne réussite des méthodes d'analyse du corpus des jeux vidéos, corpus pour lequel la majorité des valeurs de rappel, précision et F-score strict dépasse 0,5. La réussite sur le corpus des débats parlementaires est également visible sur l'histogramme n° 6 où les valeurs de rappel, de précision et de F-score strict sont comprises entre 0,5 et 0,75.

Homogénéité et hétérogénéité des résultats

Il est possible de tirer un second enseignement de la part de ces histogrammes : celui de l'homogénéité des résultats entre participants.

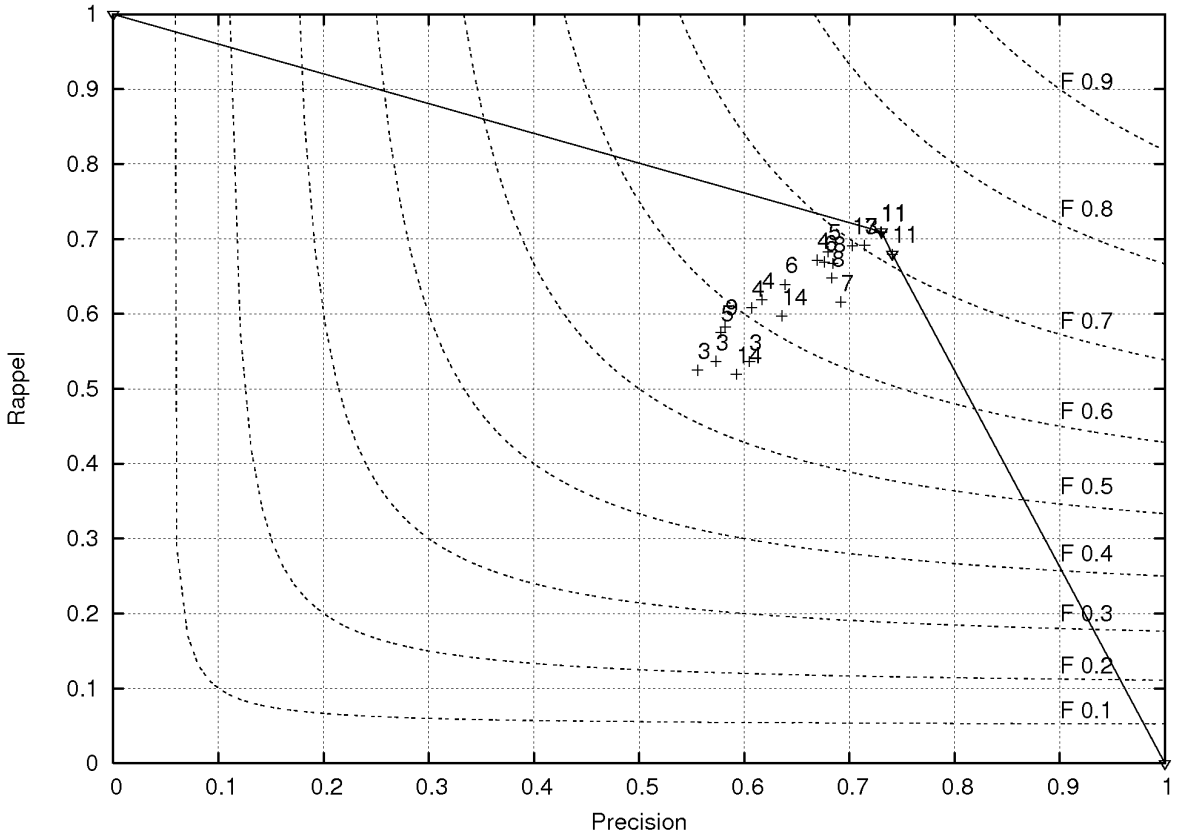


FIG. 6 – Front de Pareto pour le corpus des débats parlementaires.

Alors que les résultats sur les corpus des jeux vidéos et « *à voir, à lire* » sont assez hétérogènes selon les candidats et les soumissions et apparaissent clairsemés dans ces histogrammes (figures n° 5 et 7), les résultats portant sur les corpus des débats parlementaires et des relectures sont davantage homogènes et se présentent sous la forme de nuages de points assez compacts sur les histogrammes (figures n° 6 et 8).

Deux interprétations sont possibles pour ces nuages de points : soit le corpus était difficile à analyser et les résultats sont tous moyens (c'est semble-t-il le cas pour le corpus des relectures), soit au contraire le corpus était facile à analyser et les résultats ne pouvaient dès lors qu'être bons (voir tableau n° 1) ; pour ce dernier cas, c'est – nous l'espérons et le supposons – le cas du corpus des débats parlementaires où deux arguments militent en faveur de cette analyse : d'une part, l'échelle de notes réduites à deux classes (pour ou contre) et d'autre part, la qualité littéraire des retranscriptions des débats (assez peu de fautes d'orthographe et de grammaire en comparaison du corpus des relectures) permettant d'appliquer des traitements robustes.

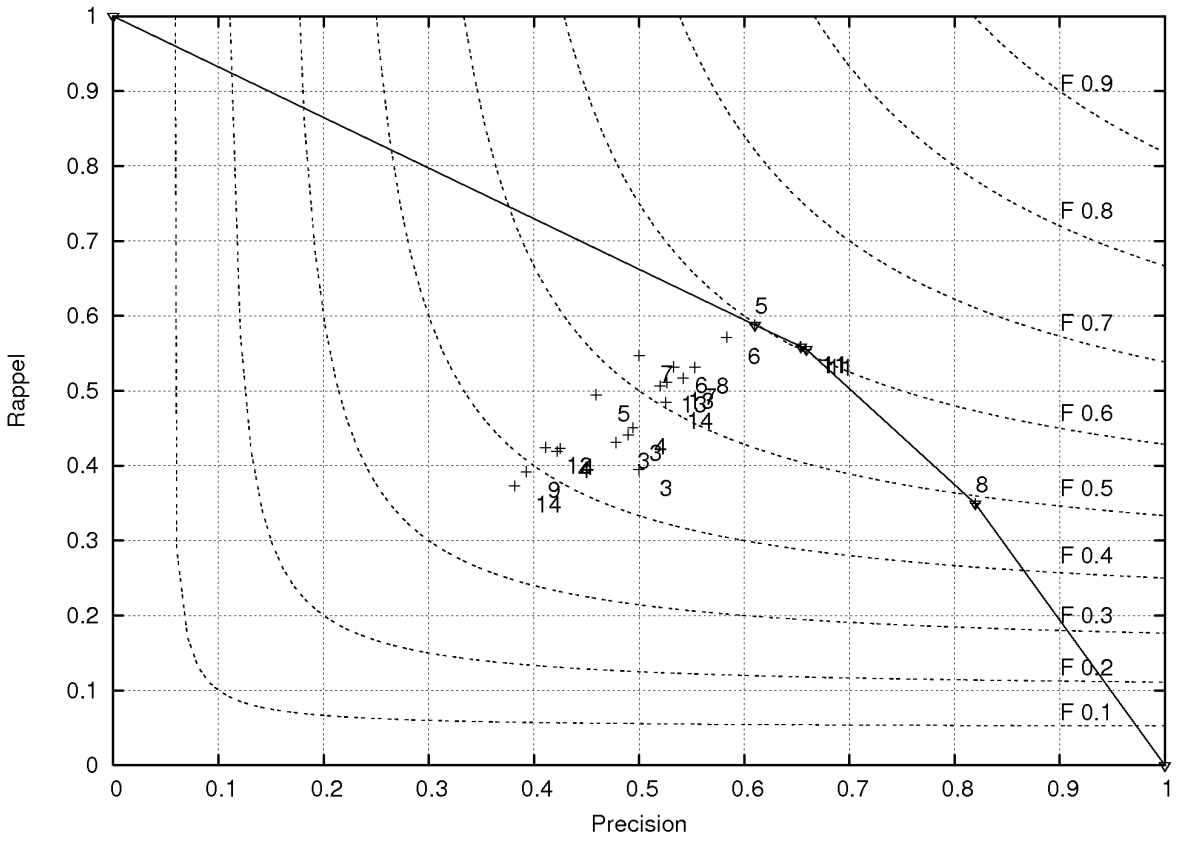


FIG. 7 – Front de Pareto pour le corpus « *à voir, à lire* ».

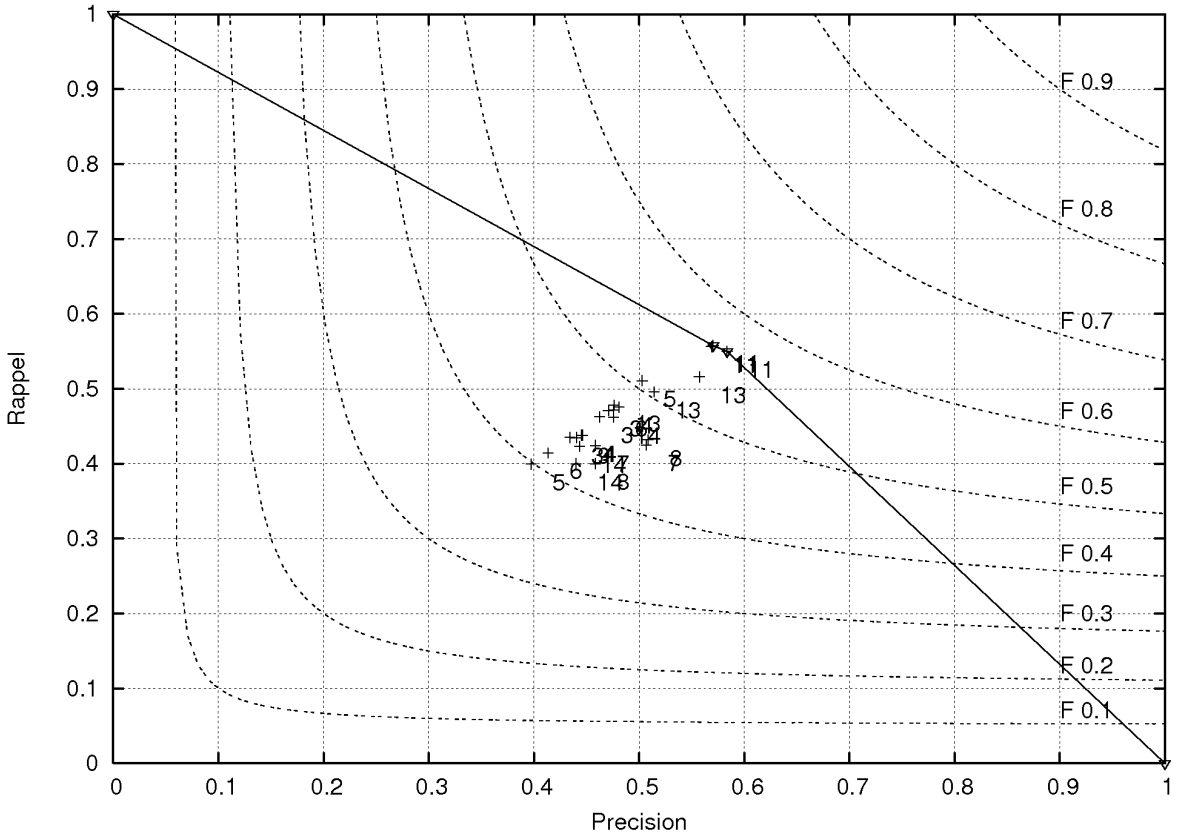


FIG. 8 – Front de Pareto pour le corpus des relectures.

3 Les méthodes utilisées par les participants

On peut séparer le processus généralement utilisé en deux grandes étapes : une première étape de représentation du texte, et une deuxième étape de classification.

L'étape de représentation du texte peut être plus ou moins élaborée, mais elle aboutit toujours à une réduction, parfois drastique, de l'ensemble des traits pouvant représenter les textes. Certaines équipes ont choisi de ne retenir qu'une partie du texte : les segments qui leur paraissaient pertinents pour l'évaluation de l'opinion. Ce peut être des paragraphes délimités tels que l'introduction et la conclusion du texte (Lattice, LIA-jeunes), ou bien des extraits trouvés par une méthode d'extraction de relations d'opinion (CELI-France), ou encore l'extraction du segment de texte autour d'un mot attracteur (LIA-jeunes). Dans le même esprit, l'équipe du GREYC-CRISCO a choisi de donner des poids différents aux différentes parties du texte.

Les méthodes de sélection des traits représentant les textes ont également été très variées. Plusieurs équipes ont utilisé un vocabulaire d'opinion (LIA, CELI-France, NLTG-Brighton, GREYC-CRISCO) soit pour pondérer les termes d'opinion dans les textes, soit pour les sélectionner. Les termes du domaine du corpus (par exemple article ou film) ont également été utilisés comme des attracteurs ou sélecteurs de termes d'opinion pertinents (LIA-jeunes, CELI-France, GREYC-CRISCO). L'équipe du Lattice a ajouté comme traits des statistiques sur les parties du discours. L'équipe du EPHE-CNRS et Universität-Würzburg a construit des concepts par analyse sémantique latente. Enfin, plusieurs participants ont utilisé plus classiquement une discrimination des traits importants pour chaque classe ou chaque texte par un critère statistique ou probabiliste tel que tf*idf, gain d'information, ou information mutuelle (LIA-jeunes, Lattice, Yahoo !Inc., LIP6, NLTG-Brighton).

L'étape de classification est également riche en méthodes différentes. Le classifieur le plus utilisé a été la machine à vecteur de support (SVM), mais ce n'est pas celui qui a produit les meilleurs résultats. Certaines équipes ont conçu des méthodes hybrides utilisant au moins deux classifieurs (LIA, LGI2P-LIRMM, CELI-France). C'est l'équipe du LIA qui a poussé le plus loin la méthode en prenant 6 classifieurs avec des variantes dans la représentation du texte donnant 9 systèmes de décision, un même poids étant attribué à chaque système dans la fusion finale. Cette méthode a produit les meilleurs scores. Une autre méthode utilisée plusieurs fois avec un certain succès a été la sommation de scores calculés sur chaque terme d'un document (Yahoo !Inc., LIA-jeunes, GREYC-

CRISCO) ou sur chaque relation d'opinion (CELI-France). Parmi les autres méthodes de classification on trouve les arbres de décision (Lattice, LIA, LGI2P-LIRMM), la régression logistique (LIA-jeunes, Lattice), des méthodes probabilistes (LIA, LGI2P-LIRMM, CELI-France), des réseaux de neurones (LGI2P-LIRMM), un algorithme de boosting (LIA), l'algorithme des k plus proches voisins (LIA), un classifieur à base de règles d'association (GREYC-CRISCO), et un calcul de similarité entre le vecteur représentant une classe et le vecteur représentant un texte (LIA-jeunes, EPHE-CNRS et Universität-Würzburg).

Conclusion

Les résultats ont été bons dans l'ensemble et finalement assez proches les uns des autres. Les résultats des tests faits avec les juges humains sont légèrement supérieurs mais montrent le même ordre de difficulté dans le traitement des corpus que les méthodes automatiques : le corpus des relectures est le plus difficile à évaluer, et celui des tests des jeux vidéos le plus facile. Les participants ont utilisé des méthodes très variées allant des approches statistiques à des approches linguistiques, syntaxiques ou sémantiques. L'utilisation d'un vocabulaire d'opinion a produit de bons résultats. La sélection des traits représentant le texte semble, au vu des résultats, presque plus importante que la classification proprement dite. Par ailleurs, les méthodes hybrides de classification semblent prometteuses.

Approches statistiques

LaGratounette : classification automatique générique de textes d'opinion

Alejandro Acosta, André Bittar

LATTICE-CNRS (UMR 8094), Université Paris 7
{aacosta,abittar}@linguist.jussieu.fr

Résumé : Nous présentons le bilan de la participation de l'équipe de jeunes chercheurs de l'équipe TALANAdu laboratoire Lattice au 3^e Défi Fouille de Textes (DÉFT'07). Le défi de cette année était de classifier les documents de 4 corpus différents selon l'opinion exprimée par chacun d'entre eux. Cet article présente le travail entrepris par notre équipe - notre méthodologie, les ressources utilisées, les étapes suivies lors du traitement et les résultats obtenus par l'application de notre approche.

Mots-clés : classification de documents, textes d'opinion, chaîne de traitement, Weka

1 Introduction

L'approche proposée utilise les modules d'une chaîne de traitement linguistique pour enrichir les documents des corpus DÉFT'07 d'annotations diverses. Ces annotations, et des dictionnaires lexicales générés automatiquement à partir du corpus d'apprentissage ont été utilisés pour la construction de classifieurs pour chaque classe de corpus. Les classifieurs les plus performants ont été choisis avec une plate-forme de exploration de données et utilisés pour le traitement des corpus de test.

Dans un premier temps, notre équipe de jeunes chercheurs a envisagé une approche qui comportait trois niveaux de paramètres à étudier et utiliser dans la tâche de la classification des textes d'opinion. Ces trois niveaux étaient : un niveaux des statistiques sur le comptage d'annotations linguistiques (par exemple, le nombre d'adjectifs, de pronoms clitiques négatifs, etc.), un niveaux modélisant les lexies isolées qui caractérisent chaque classe d'opinion dans la collection de documents utilisée pour l'apprentissage et, finalement, un niveau comportant de paramètres avec des éléments langagières plus complexes (des collocations, des expressions figées, etc.).

Les deux premiers de ces niveaux constituent l'ensemble de paramètres génériques de l'approche ; les techniques utilisées pour calculer les paramètres de ces niveaux peuvent s'appliquer sans aucune modification à une tâche de classification différente, c'est-à-dire qu'il s'agit de paramètres qui ne dépend pas des détails de la tâche de classification des textes d'opinion et qui caractérisent, tout simplement, des textes en langue naturelle.

Les annotations linguistiques, d'une part, ne dépendent pas du domaine du corpus (du type des textes, de leur contenu, des thèmes qu'ils abordent). Quant au vocabulaire, s'il est étudié strictement en isolation (c'est-à-dire en fonction des occurrences des mots dans un corpus d'apprentissage) et non pas en fonction de, par exemple, un champ lexical spécifique à un domaine, les techniques utilisées pour l'exploiter de manière automatique peuvent aussi s'appliquer à tout autre texte.

En revanche, les paramètres du troisième niveau concernent des éléments qui ne pourraient pas vraiment être définis indépendamment du domaine d'application. Dans le cas de la classification des textes d'opinion, la définition de ces paramètres permettrait une analyse plus riche, mais aussi moins générique, car elle repose sur des caractéristiques propres à l'expression de l'opinion (voire propres à l'opinion dans un type de corpus d'opinion).

Malheureusement, tous les membres de l'équipe initiale n'ont pas pu s'investir jusqu'au bout dans le projet prévu. Par conséquent, nous sommes restés au niveau générique de l'apprentissage et de la classification des documents du corpus. Pour ce faire, nous avons procédé en plusieurs étapes :

1. Filtrage du corpus
2. Pré-traitement
3. Construction de dictionnaires de classification

4. Reconnaissance des lexies de classification
5. Calcul des paramètres de classification
6. Evaluation des modèles de classification
7. Ibid 1, 2, 4 et 5 pour le corpus d'évaluation
8. Classification du corpus d'évaluation avec les modèles choisis

On remarquera que nous n'avons finalement pas utilisé des informations sur des structures du troisième niveau présenté ci-dessus. En effet, nous nous sommes limités à une étude élémentaire des paramètres des deux premiers niveaux. Or, nos trouvons qu'il est tout de même important de présenter nos techniques à la communauté qui à participé au DÉFT'07.

Le reste de ce document est organisé de la manière suivante : dans la section 2 nous présentons les grandes lignes de l'approche générique que nous avons utilisée, dans la section 3 nous présentons les idées concernant le filtrage des corpus, dans la section 4 nous présentons la chaîne de traitement linguistique utilisée pour le pré-traitement des corpus, dans la section 5 nous présentons la technique utilisée pour l'obtention des dictionnaires de classification lexicalisés, dans la section 6, nous présentons la technique utilisée pour calculer les paramètres pour chaque document, dans la section 7 nous présentons la plateforme d'exploration de données utilisée pour l'évaluation des modèles de classification, dans la section 8 nous parlons des résultats obtenus, et finalement dans la section 9 nous présentons les conclusions de notre expérience.

2 Classification générique

Le point de départ de la classification générique est l'ensemble des idées suivantes : les annotations linguistiques peuvent être utilisées comme paramètres pour la classification de documents en langue naturelle. Par ailleurs, certaines formes lexicales ont une importance plus grande que d'autres lorsqu'il s'agit d'identifier des classes différentes. Finalement, il y a des segments des documents qui sont plus importants pour leur classification.

Nous considérons qu'il n'est pas nécessaire d'utiliser la totalité du texte de chaque document pour le classifier. Dans chaque document il peut y avoir des segments qui ne sont pas très pertinents pour sa classification. Il en va de soi qu'il y a un segment (ou des segments) qui sont plus importants que d'autres pour identifier un texte comme appartenant à une classe particulière. Un des buts de notre approche est donc de choisir les extraits les plus pertinents pour la classification des documents et ignorer le reste de leur contenu.

Dans les segments pertinents, ceux qui résultent d'un filtrage des documents, il existe des marqueurs lexicaux précis qui se distinguent par leur association aux classes prédéfinies. Ainsi, on trouvera, par exemple, que des extraits comme *un très bon film* ou *je m'oppose*, sont (dans des corpus de critiques de films et de débats politiques, respectivement) plus utiles que d'autres pour classifier des documents. Ces marqueurs peuvent être des mots isolées ou des expressions plus longues, composées de plusieurs mots[1].

Par ailleurs, le type de langage utilisé dans chaque classe de document (et pour chaque type de corpus) varie en fonction de sa classe. Le texte d'une relecture d'un article qui rejette ce dernier est souvent critique et par conséquent plus négatif qu'affirmatif, par exemple. Les formes et structures choisies par l'auteur d'un texte ne sont donc pas sans rapport avec le type de texte dont il s'agit. On peut utiliser des annotation linguistiques génériques dans le but de capturer les particularités du langage des classes différentes des documents.

L'approche générique consiste alors à filtrer les documents pour ne garder que les segments (qui risquent d'être) pertinents pour leur classification. Les documents filtrés sont ensuite annotés par des modules génériques d'annotation linguistique, et on détecte aussi les occurrences des lexies avec des distributions intéressantes dans les différentes classes. Ces deux types de données (annotations et lexies) sont utilisées pour décrire chaque document, elles deviennent les paramètres de classification.

Le corpus d'apprentissage est utilisé pour trouver un modèle statistique qui donne de bons résultats avec les paramètres qui peuvent être calculés de manière automatique.

[1] On appelle ici *expression*, de manière très générique, une séquence de mots dans un texte. Nous ne faisons aucune hypothèse sur le statut ou la nature linguistique de ces objets.

3 Filtrage des documents

Le choix des organisateurs du DÉFT'07 de diviser le corpus d'évaluation en 4 classes différents est en fait un premier filtrage qui s'opère sur le corpus d'évaluation (là où la classification des textes d'opinion, dans un sens général, pourrait être définie pour tout type de textes). Le résultat est un ensemble de sous-classes de textes d'opinion. Pour chacune de ces sous-classes de textes d'opinion, un autre niveau de classification a été établi, et c'est à ce niveau que les systèmes participant au DÉFT'07 s'intéressent.

Nous considérons que les documents qui font partie de chaque sous-classe peuvent à leur tour passer par un nouveau filtrage. Le but de ce nouveau filtrage est de repérer les segments qui sont plus pertinents pour la classification de chaque classe, pour pouvoir ignorer les segments qui sont moins orientés vers une ou une autre. Si l'on cible les contenus qui nous intéressent à l'intérieur de chaque document, on simplifie le calcul de paramètres.

Le filtrage que nous avons appliqué aux corpus d'apprentissage et d'évaluation a été très élémentaire. Nous nous sommes basés sur nos intuitions et sur un survol des corpus pour déterminer les segments les plus pertinents pour la classification de chacun.

Ainsi, pour le corpus de critiques de films, livres, spectacles et bandes dessinées nous avons choisi de ne garder que les premières 4 phrases de chaque critique. Pour le corpus de tests de jeux vidéo seul le dernier paragraphe (celui employé comme le résumé de la critique) a été gardé pour la classification. Quant au corpus de relectures d'articles , nous avons gardé aussi les 4 premières phrases. Finalement, pour le corpus de débats parlementaires nous avons chois de garder les premières deux et les dernières deux phrases de chaque document, celles qui correspondent, grosso modo, à l'introduction et la conclusion de la participation d'une personne dans un débat.

Bien qu'assez grossier, ce filtrage est une approximation d'un filtrage qui saurait bien distinguer le contenu pertinent de celui qui l'est moins.

4 Les outils MACAON

Après le filtrage, les documents ont été enrichis avec des annotation linguistiques standards. Ces annotations constituent le pré-traitement du contenu qui est utilisé pour calculer les paramètres de classification pour chaque corpus.

MACAON[2] est une architecture modulaire de traitement automatique de langues. Plusieurs modules de cette architecture sont en cours de développement pour l'annotation des textes en français. Les modules utilisés pour l'annotation des documents des corpus DÉFT'07 ont été tirés de cette collection. Il s'agit des modules qui s'occupent des tâches suivantes :

1. Segmentation en phrases
2. Tokenisation
3. Reconnaissance d'entités nommées
4. Analyse lexicale
5. Etiquettage morpho-syntaxique
6. Analyse morphologique
7. Analyse syntaxique partielle

Ces modules ont été appliqués dans l'ordre d'apparition ci-dessus. L'entrée de cette chaîne de modules était donc le texte filtré de chaque document. La sortie est un document XML structuré et enrichi avec des annotations.

5 Construction des dictionnaires

Un dictionnaire a été construit à partir du corpus d'apprentissage pour chaque classe de texte à évaluer, suivant une même procédure.

Le texte des segments retenus après le filtrage (voir section 3) a été découpé en items lexicaux, c'est-à-dire en séquences de caractères séparés par un espace. Ensuite, pour chaque corpus, nous avons calculé :

[2] http ://code.google.com/p/macaon/

1. Le nombre d'occurrences de chaque item
2. Les items uniques à chaque classe
3. La classe maximisant le nombre d'occurrences de chaque item

Ces données nous ont permis de trier les items *uniques* à chaque classe par le nombre de leurs occurrences et de calculer l'importance des items dont les occurrences étaient *maximales* dans chaque classe. Cette importance (I dans la formule 1) résulte de la magnitude de la différence du nombre d'occurrences de l'item dans la classe dans laquelle il apparaît le plus souvent (i_{max}) et la classe dans laquelle il apparaît le moins souvent (i_{inf} avec $i_{inf} > 0$).

$$I(i) = \frac{i_{max} - i_{inf}}{i_{max}} \quad (1)$$

Etant donnée que chaque type de corpus d'apprentissage comportait un nombre très différent de documents, les comptages des occurrences des items lexicaux ont dû être interprétés pour chaque corpus.

Nous n'avons pas implémenté une méthode automatique de sélection des items de chaque dictionnaire. La dernière étape de leur constitution a donc consisté à décider, après un survol des résultats, quels étaient les seuils permettant de trouver un bon compromis entre la quantité d'entrées et la capacité de classification des entrées. Nous noterons, par exemple, que les mots uniques à une classe particulière, mais qui n'apparaissent qu'une fois dans tout le corpus, sont moins importants pour la classification qu'un mot qui apparaît autant de fois qu'il y a de documents.

Les paramètres à déterminer étaient (pour chaque corpus) les suivants :

1. Le nombre maximal d'items *uniques*
2. Le nombre maximal d'items *maximaux*
3. Pour les items *uniques*, le seuil de pertinence du nombre d'occurrences.
4. Pour les items *maximaux*, le seuil de pertinence du nombre d'occurrences.
5. Pour les items *maximaux*, le seuil d'importance de la déviation de ses occurrences

Le module de repérage d'entités nommées de MACAON a été utilisé pour marquer les occurrences des items du dictionnaire dans les corpus d'apprentissage et d'évaluation.

6 Calcul de paramètres

Pour calculer les différents paramètres utilisés pour la création du modèle de classification nous sommes partis des collections de fichiers XML pré-traités correspondant à chaque corpus.

Le calcul des paramètres a été fait par le programme LAGRATOUNETTE, qui prend en entrée une collection de documents pré-traités et une configuration de paramètres à calculer pour donner, en sortie, un fichier dans le format ARFF utilisé par WEKA[3] pour la création du classifieur ou l'application d'un modèle de classification à un corpus.

La configuration des paramètres de LAGRATOUNETTE se fait avec un fichier qui liste les étiquettes à associer aux paramètres (leurs noms) ainsi que la description des éléments à considérer et le type de calcul à effectuer. Ces calculs peuvent être de simples comptages, des facteurs, des déviations de la moyenne dans le corpus, ou simplement la présence ou l'absence d'un élément.

De cette manière on a calculé le nombre d'occurrences des différents parties du discours ; par exemple, la proportion de groupes nominaux par rapport aux groupes verbaux, de groupes verbaux finis par rapport aux groupes verbaux, etc. ; les déviations dans le nombre de dates ou de formes verbales dans les différents modes et temps ; etc.

Dans la figure 1 nous présentons la séquence de tâches effectuées avec les différents corpus lors de l'évaluation. Dans un premier temps le corpus d'évaluation est filtré et pré-traité avec MACAON (en 1), ensuite, on repère les occurrences des éléments des dictionnaires de classification (en 2). Les documents enrichis avec les annotations linguistiques et celles qui correspondent aux lexies sont utilisés par LAGRATOUNETTE (en 3) pour produire les descriptions des documents (en termes de vecteurs de paramètres). Ces descriptions

[3] Voir section 7 pour la présentation de ce logiciel de gestion de classifieurs.

sont ensuite classifiées suivant un modèle (en 4) et, enfin, les résultats de la classification faite par le classifieur, en format ARFF, sont converties en XML (en 5), selon la structure établie par les organisateurs de DÉFT'07.

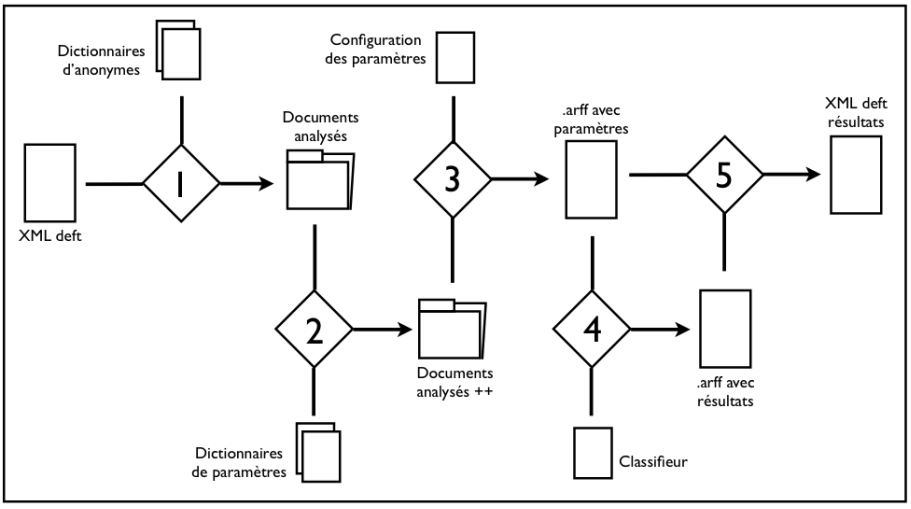


FIG. 1 – La séquence de traitements lors de l'évaluation

7 Evaluation de classifieurs

Weka[4] est un logiciel libre implémenté en Java qui comprend une collection d'algorithmes d'apprentissage (classfieurs) pour des tâches de fouille de données. Il fournit des outils pour le pré-traitement, la classification, la régression, le clustering, des règles d'association et la visualisation des données.

Nous avons utilisé certains des classifieurs pour attribuer une classe, dans notre cas un score soit de 0,1 ou 2, soit de 0 ou 1, à chacun des documents du corpus. Chaque document de corpus est représenté pour WEKA par un ensemble d'attributs (appelé une "instance"). Ces attributs correspondent aux paramètres que nous avons calculés pour chaque document lors du pré-traitement – le nombre d'occurrences de certains mots, étiquettes morphologiques, etc. – ainsi qu'à la note qui lui a été attribuée, dans le cas du corpus d'apprentissage.

Afin de déterminer quel classifieur était le mieux adapté à chaque type de corpus, nous avons effectué des tests préalables, par validation croisée, sur des portions de corpus pré-traités. Les classifeurs qui ont donné les meilleurs résultats pendant ces tests ont été sélectionnés pour l'évaluation.

Un ensemble différent d'attributs a été utilisé selon le type de corpus, les attributs les plus pertinents n'étant pas les mêmes pour tous. Le bilan des paramètres les plus utiles pour la classification est détaillé ci-dessous. Nous présentons ces attributs dans l'ordre de pertinence décroissant pour les 4 corpus.

7.1 Corpus critiques de films, livres, spectacles et bandes dessinées

- **ZERO unique, fréquence=L**
  Le nombre de mots, ayant une fréquence relativement basse, apparaissant uniquement dans les documents de score 0. Le graphe ci-dessous représente la distribution des classes de documents selon le nombre d'occurrences de tels mots. L'abscisse représente l'attribut (c'est-à-dire, le nombre de mots de ce type dans un document), et l'ordonné le nombre de documents ayant ce nombre d'occurrences. Chaque barre horizontale est séparée en trois couleurs, chacune d'entre elles représentant une classe (score) où gris foncé = 0, gris moyen = 1 et gris clair = 2. Ainsi, le graphique représente les proportions du total attribuées à chaque classe.

[4]http ://www.cs.waikato.ac.nz/ml/weka/

Pour cet attribut, la figure 2 montre que lorsqu'un document compte un mot de ce type, il y a environ 5 fois plus de chance qu'il soit de score 0 que de score 1, et environ 10 fois plus que de score 2. Si le document compte plus d'un tel mot, il est sûr d'avoir un score de 0.

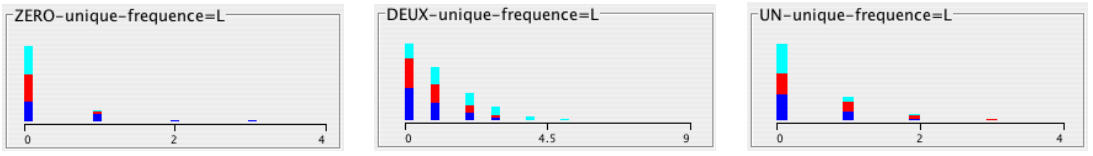


FIG. 2 – 1^{er}, 2^{me} et 3^{me} meilleurs paramètres, corpus critiques de films, livres, spectacles et bandes dessinées

- **DEUX unique, fréquence=L**
  Le nombre de mots, ayant une fréquence relativement basse, apparaissant uniquement dans les documents de score 2. Une tendance s'établit lorsqu'un document possède trois mots de cette catégorie, figure 2. Dans ce cas, un score de 2 est environ quatre fois plus probable que 0 ou 1. Avec quatre mots, la probabilité qu'un document ait un score de 2 est environ huit fois plus que pour 1 ou 0. Au-delà de quatre mots, le document est sûr d'avoir un score de 2.

- **UN unique, fréquence=L**
  Le nombre de mots, ayant une fréquence relativement basse, apparaissant uniquement dans les documents de score 1. Dans la figure 2, on constate une légère préférence pour un score de 1 lorsqu'un document compte un de ces mots. La distinction devient beaucoup plus marqué pour un document comportant 2 mots de ce type, avec une probabilité de score 1 environ 3 fois plus que pour les autres scores respectivement. A trois mots, il reste une petite probabilité de score 0, mais un score de 1 est massivement plus probable. Au-delà de trois mots, le document est sûr d'avoir un score de 1.

- **comptage dates**
  le nombre d'entités nommées. La figure 3 représente une courbe. On peut constater que dans l'attaque de la courbe (entre 3 et 10 occurrences) la probabilité d'un score 0 est plus importante que pour les autres scores. Au milieu de la courbe, et jusqu'à la chute, les distributions sont relativement proches. A partir de la chute de la courbe, la probabilité d'un score 0 diminue de façon significative et il y a une probabilité plus importante pour un score de 2 (environ deux fois plus probable) que pour un score de 1. Un document comptant un nombre plus élevé d'entités nommées est donc probablement un document de score 2.

- **deviation dates**
  la déviation de la moyenne du nombre d'entités nommées qui sont des dates. La figure 3 représente les tendances inverse de la précédente. En général, pour les valeurs de déviation de la moyenne du nombre de dates entre -70 et -28, la probabilité d'un score de 2 est largement plus élevée. Sur l'attaque de la courbe, les score 2 et 1 sont plus probables que 0. Au milieu de la courbe, et jusqu'à la chute, les distributions sont relativement proches. A partir de la chute de la courbe (une déviation de la moyenne du nombre de dates de 20 à 30), la probabilité d'un score de 0 est plus importante que pour les autres scores.

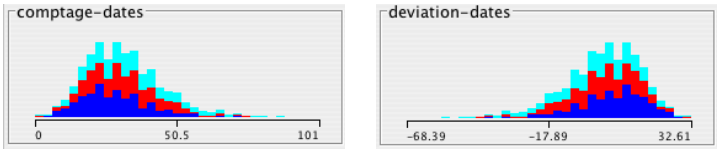


FIG. 3 – 4^{eme} et 5^{me} meilleurs paramètres, corpus critiques de films, livres, spectacles et bandes dessinées

7.2 Corpus tests de jeux vidéo

- **ZERO unique, fréquence=L**
  Le nombre de mots, ayant une fréquence relativement basse, apparaissant uniquement dans les documents de score 0. La figure 4 montre qu'avec un mot de ce type, un document est environ quatre fois plus probable d'avoir un score de 0 que chacun des autres notes respectivement. A deux occurrences il y a une probabilité massive d'un score de 0, une minuscule probabilité d'avoir un score de 1, et aucune probabilité de 2. Au-delà, un score de 0 est certain.

- **ZERO unique, fréquence=M**
  Le nombre de mots, ayant une fréquence moyenne, apparaissant uniquement dans les documents de score 0. Dans la figure 4 on remarque qu'avec un mot de ce type, la probabilité d'un score de 0 est presque 1, avec une petite probabilité de 1. Un score de 0 est certain pour un document comptant plus d'une occurrence d'un tel mot.

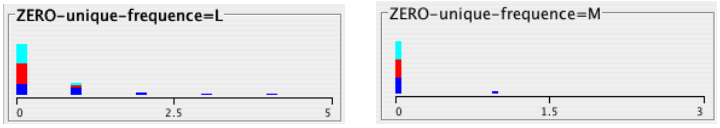


FIG. 4 – 1^{er} et 2^{me} meilleurs paramètres, corpus tests de jeux vidéo

- **DEUX unique, fréquence=L**
  Le nombre de mots, ayant une fréquence relativement basse, apparaissant uniquement dans les documents de score 2. La figure 5 montre qu'un score de 2 est environ trois fois plus probable pour un document comptant un mot de ce type et qu'au-delà un score de 2 est une certitude.

- **UN max, déviation=H, comptage=L**
  Ce paramètre représente le nombre de mots apparaissant plus souvent dans les documents de score 1 que dans les autres, dont le nombre d'occurrences n'est pas élevé et dont la déviation de la moyenne d'occurrences dans les classes différentes est importante. La figure 5 montre qu'un document qui ne contient aucun mot de ce type a une forte probabilité d'avoir un score de 0. De 1 à 3 occurrences, les probabilités pour chaque score sont relativement proches, mais à partir de 4 mots la probabilité d'un score de 0 réduit dramatiquement et un score de 1 est environ deux fois plus probable qu'un score de 2.

- **DEUX unique, fréquence=M**
  Le nombre de mots, ayant une fréquence moyenne, apparaissant uniquement dans les documents de score 2. Avec un mot de ce type, la figure 5 montre qu'un score de 2 est environ quatre fois plus probable que les autres scores respectivement. Au-delà, un score de 2 est une certitude.

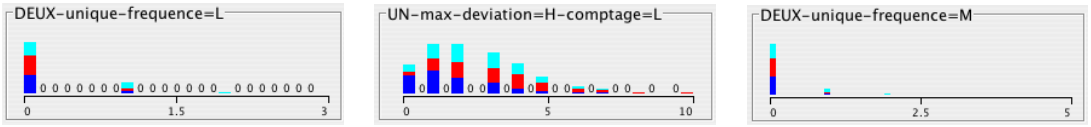


FIG. 5 – 3^{me}, 4^{me} et 5^{me} meilleurs paramètres, corpus tests de jeux vidéo

7.3 Corpus relectures d'articles

- **ZERO unique, fréquence=L**
  Le nombre de mots, ayant une fréquence relativement basse, apparaissant uniquement dans les documents de score 0. Comme nous montre la figure 6, un score de 0 est environ deux fois plus probable qu'un score de 1 et environ trois fois plus probable qu'un score de 2 si le document contient un mot

de ce type. A deux et trois occurrences, cette tendance est exagérée. A partir de quatre occurrences, un score de 0 est certain.

- **UN unique, fréquence=L**
  Le nombre de mots, ayant une fréquence relativement basse, apparaissant uniquement dans les documents de score 1. Ce graphe, figure 6, représente une courbe descendante, qui montrent une probabilité croissante d'un score de 1 en fonction du nombre d'occurrences d'un mot de ce type. A partir de cinq occurrences, un score de 1 est sûr.

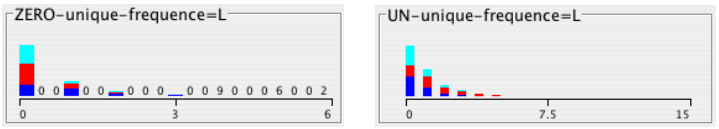


FIG. 6 – 1^{er} et 2^{me} meilleurs paramètres, corpus relectures d'articles

- **ZERO max, déviation=M, comptage=H**
  Le nombre de mots apparaissant plus souvent dans les documents de score 0 que dans les autres, dont le nombre d'occurrences est élevé et dont la déviation de la moyenne d'occurrences dans les classes différentes est assez importante. La figure 7 montrent la même tendance que la précédente, mais pour un score de 0. Plus il y a de mots de ce type dans un document, plus il est probable d'avoir un score de 0.

- **DEUX unique, fréquence=L**
  Le nombre de mots, ayant une fréquence relativement basse, apparaissant uniquement dans les documents de score 2. La figure 7 montre encore une courbe descendante. La probabilité d'un score de deux augmente en fonction du nombre d'occurrences des mots de ce type. A partir de quatre mots, un score de 2 est une certitude.

- **DEUX unique, fréquence=M**
  Le nombre de mots, ayant une fréquence moyenne, apparaissant uniquement dans les documents de score 2. Avec un mot de ce type, la figure 7 montre qu'un score de 2 est environ deux fois plus probable qu'un score de 1 ou de 0. A deux occurrences, un score de 2 est quasiment sûr et au-delà devient certain.

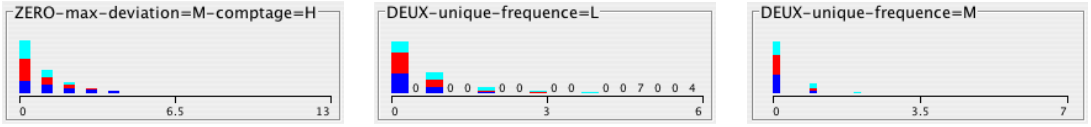


FIG. 7 – 3^{me}, 4^{me} et 5^{me} meilleurs paramètres, corpus relectures d'articles

7.4 Corpus débats parlementaires

- **CONTRE max, déviation=M, comptage=L**
  Ce paramètre représente le nombre de mots apparaissant plus souvent dans les documents classés *contre* que dans ceux classés *pour*, dont le nombre d'occurrences est relativement basse et dont la déviation du nombre d'occurrences dans la classe *pour* est assez importante. La figure 8 représente une courbe descendante où la probabilité d'un score de 0 (vote "contre") augmente en fonction du nombre d'occurrences d'un mot de ce type.

- **POUR unique, fréquence=L**
  Le nombre de mots, ayant une fréquence relativement basse, apparaissant uniquement dans les documents de score 1 (vote "pour"). Cette figure, 8, montre qu'avec un mot de ce type, un score de 1 est

environ trois fois plus probable qu'un score de 0. Au-delà, cela devient une certitude.

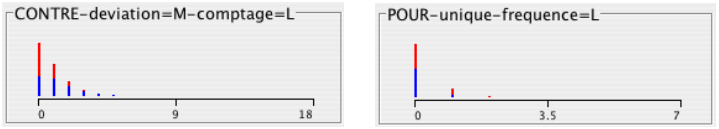


FIG. 8 – 1^{er} et 2^{me} meilleurs paramètres, corpus débats parlementaires

- **CONTRE unique, fréquence=L**
  Le nombre de mots, ayant une fréquence relativement basse, apparaissant uniquement dans les documents de score 0. La figure 9 montre la tendance inverse de la précédente. A une occurrence d'un mot de ce type, un document a une probabilité d'un score de 0 environ trois fois plus élevée que pour un score de 1. Au-delà cela devient une certitude.

- **CONTRE max, déviation=H, comptage=L**
  Ce paramètre représente le nombre de mots apparaissant plus souvent dans les documents classés *contre* que dans ceux classés *pour*, dont le nombre d'occurrences est relativement basse et dont la déviation du nombre d'occurrences dans la classe *pour* est importante. Avec une seule occurrence d'un mot de ce type, la figure 9 montre qu'un score de 0 est environ quatre fois plus probable qu'un score de 1. Au-delà d'une occurrence, un score de 0 est certain.

- **POUR max, déviation=H, comptage=L**
  Ce paramètre représente le nombre de mots apparaissant plus souvent dans les documents classés *pour* que dans ceux classés *contre*, dont le nombre d'occurrences est relativement élevé et dont la déviation du nombre d'occurrences dans la classe *contre* est faible. La tendance ici est similaire. La figure 9 montre qu'avec une occurrence d'un mot de ce type, la probabilité qu'un document ait un score de 1 est environ trois fois plus importante que pour un score de 0. Au-delà d'une occurrence, le document est sûr d'avoir un score de 1.

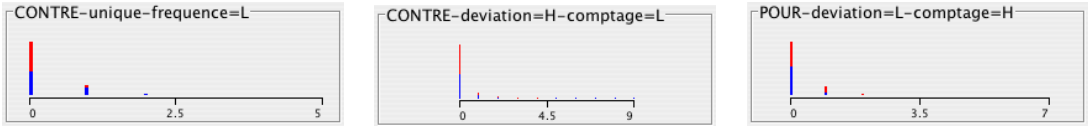


FIG. 9 – 3^{me}, 4^{me} et 5^{me} meilleurs paramètres, corpus débats parlementaires

Lors de notre étude des différents modèles de classification, nous avons aussi évalué la performance des deux différents types de paramètres : les paramètres calculés sur les annotations linguistiques, et ceux issus des dictionnaires. A première vue, on remarque que la présence des paramètres lexicaux est prépondérante dans les listes que l'on vient de présenter. Mais la haute pertinence de ces paramètres n'exclut pas celle des autres.

En effet, quand on regarde les résultats des 5 meilleurs paramètres par eux-mêmes (c'est-à-dire, avec un modèle de classification qui ignore tous les paramètres qui ne sont pas listés dans cette section), on constate une perte importante (d'environ 10 %, pour le corpus d'apprentissage) de performance.

En revanche, la performance ne diminue que de très peu lorsqu'on exclut les paramètres les plus pertinents au moment de la création du modèle de classification. Ces paramètres ne suffisent donc pas, à eux tous seuls, pour modéliser la classification des corpus.

Pour les corpus de critiques de films, livres, spectacles et bandes dessinées , de tests de jeux vidéo et de débats parlementaires, c'est un classifieur J48 qui a donné les meilleurs résultats. Ce type de classifieur est un arbre de décision, une structure simple qui où les noeuds non terminaux représentent des tests sur un ou plusieurs attributs et les noeuds terminaux représentent les décisions prises. Quant au corpus de relectures d'articles, c'est un classifieur Logistic qui a été choisi pour l'évaluation. Ce classifieur implémente la technique de régression logistique, qui prédit les valeurs prises par une variable catégorielle binaire à partir d'une série de variables explicatives continues et/ou binaires.

8 Résultats

Les résultats de l'évaluation envoyés par les examinateurs comportent trois paramètres : la précision, le rappel et un F-score strict. Nous présentons ci-dessous, pour chaque corpus, les résultats sur l'ensemble des soumission ainsi que les résultats de notre équipe avec l'écart de nos résultat vis-à-vis de la moyenne de l'ensemble des participants au DÉFT'07. Les résultats sont présentés dans les tableaux 1 à 4.

Paramètre	Résultats sur l'ensemble	Nos résultats	Ecart de la moyenne
Précision	0.5276 +/- 0.0982	0.3927	0.1349
Rappel	0.4829 +/- 0.0683	0.3920	0.0909
F-score	0.5004 +/- 0.0668	0.3923	0.1081

TAB. 1 – Corpus de critiques de films, livres, spectacles et bandes dessinées

Paramètre	Résultats sur l'ensemble	Nos résultats	Ecart de la moyenne
Précision	0.6925 +/- 0.0996	0.5324	0.1601
Rappel	0.6367 +/- 0.0921	0.5405	0.0962
F-score	0.6604 +/- 0.0864	0.5365	0.1319

TAB. 2 – Corpus de tests de jeux vidéo

Paramètre	Résultats sur l'ensemble	Nos résultats	Ecart de la moyenne
Précision	0.4804 +/- 0.0490	0.4403	0.0401
Rappel	0.4617 +/- 0.0477	0.4348	0.0269
F-score	0.4706 +/- 0.0468	0.4375	0.0331

TAB. 3 – Corpus de relectures d'articles

Comme on a déjà vu dans la section 7, les paramètres les plus pertinents de notre approche ont été les éléments lexicaux dans le corpus d'apprentissage associés à une certaine opinion. Les documents qui ont été classifiés correctement doivent avoir une distribution des lexies semblable à celle qui a produit les dictionnaires générés automatiquement.

En ce qui concerne les documents mal classifiés, on peut supposer que les paramètres génériques n'ont pas suffit. Il reste à voir si l'utilisation de paramètres spécifiques aux textes d'opinion (et même pour les types différents de textes d'opinion) améliore les résultats de manière significative. La réponse se trouve sans doute dans les rapports des équipes qui ont intégré des connaissances spécifiques de ces domaines à la construction de leur modèles de classification.

Il est cependant intéressant de constater que les résultats de notre approche générique ne s'écartent pas trop de la moyenne, surtout dans le cas des corpus de relectures d'articles et de débats parlementaires .

9 Conclusion

Nous avons présenté une approche à la classification des textes d'opinion fondée sur un modèle statistique dont l'apprentissage tient compte de deux types de paramètres : un premier ensemble de paramètres correspond à des statistiques concernant des annotations linguistiques associées aux documents. Un deuxième ensemble correspond à des paramètres issus d'une analyse statistique des items lexicaux avec une incidence importante sur la classification des documents. Ces deux ensembles de paramètres ont été calculés automatiquement ; aucune information lexicale (synonymes, expressions figées, collocations, etc.) extérieur au corpus d'apprentissage n'a été ajoutée avant l'application des modèles de classification au corpus d'évaluation.

Les résultats ne sembleraient pas se trouver parmi les plus performants de la campagne d'évaluation DÉFT'07. Or, compte tenu de l'écart entre les résultats des différentes équipes, il n'est pas sans intérêt de remarquer qu'une approche à la classification qui se contente de filtrer grossièrement les documents et d'utiliser un modèle générique de classification donne déjà des résultats qui ne s'éloignent pas trop de la moyenne.

Paramètre	Résultats sur l'ensemble	Nos résultats	Ecart de la moyenne
Précision	0.6545 +/- 0.0564	0.5820	0.0725
Rappel	0.6298 +/- 0.0645	0.5830	0.0468
F-score	0.6416 +/- 0.0594	0.5825	0.0591

TAB. 4 – Corpus de débats parlementaires

Par ailleurs, nous avons remarqué que les paramètres qui modélisent le type de langage utilisé dans les documents sont, par eux mêmes, assez utiles pour classifier les corpus. En effet, bien que les paramètres issus des dictionnaires d'items lexicaux soient les plus pertinents pour la classification, si l'on ne se sert que de ces paramètres on a une perte de performance importante. Si l'on les exclut, la perte de performance dans les résultats est moindre.

Références

WEISS S. M., INDURKHYA N., ZHANG T. & DAMERAU F. (2005). *Text Mining : Predictive Methods for Analyzing Unstructured Information*. Springer.

WITTEN I. H. & FRANK E. (1999). *Data Mining : Practical Machine Learning Tools and Techniques with Java Implementations*. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann.

Quel modèle pour détecter une opinion? Trois propositions pour généraliser l'extraction d'une idée dans un corpus

Eric Charton, Rodrigo Acuna-Agost[†]

Laboratoire Informatique d'Avignon, Université d'Avignon et des Pays de Vaucluse
B.P 1228 84911 Avignon Cedex 9 - France
{eric.charton, rodrigo.acuna}@univ-avignon.fr

Résumé : Nous décrivons dans cet article trois méthodes d'extraction d'une opinion dans un corpus, mises en œuvre dans le cadre de la campagne DEFT07. La première repose sur des mesures de similarité cosine et de probabilité d'appartenance d'un document à une classe en fonction des mots qu'il contient. La seconde exploite la régression logistique, méthode rarement utilisée en classification de textes. La troisième met en œuvre une technique à base de mesure de densité et de compacité inspirée des systèmes de question-réponse. Notre approche tente de tirer parti de la pluridisciplinarité de nos travaux pour obtenir une solution algorithmique de classification adaptée de manière générique à la recherche d'idées dans un texte.

Mots-clés : Ingénierie des connaissances, Optimisation, Classification de textes, Apprentissage machine, Similarités, Système de question-réponse, Recherche d'informations.

1 Introduction

La classification d'un corpus en classes pré-déterminées est une problématique importante du domaine de la fouille de textes. De manière générale, la classification revient à rechercher des dissimilarités ou des similarités entre groupes d'individus dans une population donnée. Les applications sont variées : détection de langues, filtrage de grands corpus, recherche d'information, classement thématique. DEFT07 nous propose d'explorer le domaine applicatif de la classification non plus orientée par une thématique, mais plutôt vue sous l'angle des idées. Plus précisément, ici, la tâche proposée consiste à identifier une opinion, mais une méthode appropriée pour répondre à ce besoin pourrait tout aussi bien être transposée à des segmentations de documents par les jugements, avis ou tendance qu'ils expriment. Les possibilités applicatives sont nombreuses : mesure d'une opinion retournée par la presse suite à un lancement de produit, analyse de l'engagement d'un média dans le cadre d'une étude politique, classification automatiquement des actes juridictionnels, etc...

Pour ce qui concerne DEFT07, l'étiquetage d'un corpus avec deux ou trois classes représentant n opinions "bonne", "moyenne" ou "mauvaise" (ou "favorable" et "défavorable" dans le cadre des débats parlementaires) revient à un partionnement de corpus en n classes. Nous remarquons que les corpus de textes présentés ont pour particularité de provenir de quatre sources très différentes, et par voie de conséquence, d'exprimer l'opinion sous des formes elles aussi très différentes. Ceci ne sera pas sans incidence sur nos choix de méthodes de classification. Les systèmes de classification de textes les plus efficaces sont des algorithmes d'apprentissage supervisés qui peuvent être entrainés sur un jeu de données déjà étiquetées. Ces apprentissages conduisent à la construction de modèles de classes, qui, dans notre exemple, correspondront pour un corpus de textes donné, à une opinion. Quelle forme discriminante pourrait caractériser une opinion ? Peut-elle être caractérisée par un vocabulaire et ainsi entrer dans le cadre d'un système de recherche d'information classique (mesure de distance, de similarité cosine, de probabilité d'apparition de mots, mesure statistique) ? Doit-elle être considérée comme une réponse à une question (qui aurait la forme d'une question booléenne de type "ce produit est-il bon" ?), et dans ce cas, être recherchée avec un système de question-réponse (QR) ?

[†] Supported in part by ALFA Grant II-0457-FA-FCD-FI-FC

Pour nous faire une première opinion, nous avons étudié visuellement ces textes, puis utilisé des outils de mesures statistiques et de comptage des mots contenus dans les textes (le logiciel countworld.pl[1], ainsi que l'outil LSA de mesure de co-occurences)(Favre *et al.*, 2005).

1.1 Les corpus de DEFT07

On observe en lisant le tableau TAB.1 que les corpus d'apprentissage fournis sont très disparates : le corpus relectures d'articles, par exemple contient une quantité relativement faible de documents (881) alors qu'à l'opposé, le corpus de débats parlementaire est très volumineux. On note également que ces deux corpus, à l'inverse des deux premiers, de taille plus moyenne (JeuxVidéo, et Avoir A Lire) ont pour particularité de présenter des textes dont la taille peut varier de manière conséquente.

- Nous avons pu relever d'après ces premières investigations que le corpus issu de *jeuxvidéo.com* répond à des normes journalistiques classiques en matière d'essais de produits. Ce qui revient à exprimer l'opinion de l'auteur dans le chapô [2] d'introduction et dans la conclusion, en adoptant des séquences répétitives, faisant appel à un vocabulaire essentiellement qualificatif et relativement restreint.
- Celui issu de *AvoirAlire*, qui relève lui aussi de la critique, exprime les opinions sur des produits culturels de manière bien moins tranchée et localisée, en adoptant un vocabulaire plus étendu que celui de *jeuxvidéo.com*.

Ces deux corpus, bien que répondant à des disciplines d'expression journalistique identiques - la critique - expriment les opinions de leurs auteurs par des moyens radicalement opposés : tranché et localisé dans un cas, subtil et diffus dans l'autre.

- Pour ce qui est du corpus de relectures d'articles, on observe très nettement une concentration de la qualification autour d'un ensemble de mots très réduit (il est possible de visualiser ce phénomène en utilisant LSA pour mesurer les co-occurences de mots les plus fréquentes).
- Dans le corpus *Débats*, on relève une modalité d'expression de l'opinion (qui est en réalité l'expression d'un engagement) très variée, et parfois délicate à évaluer, y compris après une lecture "humaine".

Nom	Classes	Nbr Train	Doc Taille max	Doc Taille min
Avoir A Lire	3	2074	4167	931
JeuxVideo	3	2537	13703	4600
Relectures	3	881	7584	153
Débats parlementaires	2	11533	3100	300

TAB. 1 – Caractéristiques des Corpus de DEFT07

1.2 Algorithmes

Partant de ces constats, nous avons imaginé des propositions originales de classification d'opinions. Nous allons dans cet article, décrire trois algorithmes d'apprentissage et de classification supervisés, appliqués sur les corpus de DEFT07.

- Nous présentons en premier lieu un algorithme classique de mesure de distance entre un document et une classe par mesure de similarité cosine entre deux vecteurs de poids de mots. Ces vecteurs représentant pour l'un une classe, pour l'autre un document. Dans cette méthode, la construction des classes repose sur une préparation du texte en vue d'en supprimer les éléments non discriminants (lemmatisation, filtrages), d'en extraire les éléments significatifs (localisation de l'opinion dans le texte, utilisation de n-grammes). La construction de classes est réalisée par apprentissage sur une partie des corpus d'entrainement. La qualité discriminante de la classe est ensuite évaluée par calcul de F-Score en testant les classes sur la partie du corpus d'entrainement non utilisée pendant l'apprentissage. Nous procédons à une optimisation de F-Score en mesurant les résultats obtenus avec les combinaisons d'options de filtrages et en ne conservant que le paramétrage le plus performant.
- En second lieu, nous utilisons un algorithme statistique basé sur la régression logistique. La méthode cherche par un processus de calcul de fréquences de mots observés dans les différentes classes du

[1] Outil trivial de comptages des occurences de mots dans un document, mis au point par Benoit Favre.

[2] Le terme *chapô* décrit, dans le secteur de l'édition périodique, le résumé intercalé entre le titre et le corps de l'article

corpus d'apprentissage, à élaborer les caractéristiques d'un modèle. Ces caractéristiques sont les variables explicatives. Ces caractéristiques sont ensuite utilisées pour analyser les valeurs prises par une variable quantitative catégorielle, correspondant aux observations faites sur un document à classer dans l'une des catégories d'opinion. On déduit de cette analyse une probabilité d'appartenance à une classe d'opinion.

– Notre troisième algorithme est inspiré des mesures de densité et de compacité de mots. Ces méthodes sont mises en oeuvre dans les systèmes de question-réponse élaborés pour les campagnes Trec ou Technolangue-EQueR. Nous cherchons dans chaque sous corpus (par exemple Corpus :*Débat* Classe :*favorable*), à localiser un ou plusieurs mots centroïdes susceptibles de représenter le milieu d'une phrase ou d'un passage, exprimant l'opinion. Ces mots identifiés, nous construisons des classes représentant la probabilité d'apparition d'un mot ou d'un n-gramme à proximité d'un centroïde (exemple *"bon"* à côté de *"article"*), pour une opinion donnée. Nous utilisons ensuite ces probabilités pour attribuer une classe à un document.

Cet article est organisé comme suit : dans la section 2 nous exposons les modèles de nos trois algorithmes. Dans la section 2.1, nous présentons notre méthode de classification par mesure de similarité cosine. Dans la section 2.2, nous détaillons notre implémentation de la méthode d'affectation d'un document à une classe d'opinion par régression logistique. Dans la section 2.3 nous détaillons notre proposition d'implémentation d'un système de mesure de densité et de compacité. Dans la section 3, nous développons les résultats obtenus à DEFT07 avec les données d'entrainement, puis avec les données d'évaluation, et les commentons. Nous terminons cet article par un ensemble de conclusions et tentons d'élaborer, à la lumière de nos résultats, quelques pistes de recherches futures.

2 Méthodes de classification proposées

2.1 Classification par mesures de similarité

La principale application de la recherche documentaire par mesure de similarité est constituée des moteurs de traitement de requêtes. On retrouve son principe au cœur de tous les méta-chercheurs, de Google à Yahoo, MSN, Exalead... L'idée directrice de cette méthode est la possibilité de mesurer la distance qui sépare un groupe de mots contenus dans une requête de plusieurs groupes de mots représentant les documents d'un corpus. En projetant ces ensembles dans un espace vectoriel sous une forme numérique (par exemple en affectant des poids aux mots) on évalue leur degré de proximité. Les références des ensembles dont on a ainsi mesuré la distance sont ensuite retournées sous forme d'une liste triée en fonction de leur degré d'éloignement avec la requête. Intrinsèquement, cette méthode est donc d'autant plus efficace que les sous ensembles de mots comparés ont des périmètres clairement délimités et que leurs intersections sont les plus reduites possibles. En d'autres termes, la mesure de similarité est adaptée aux extractions d'informations thématiques, mais les subtiles nuances que l'on peut observer dans l'expression d'une opinion, semblent - de prime abord - plus délicates à traiter. Particulièrement dans la langue française, où un même ensemble de mots agencés différement, peut exprimer deux opinions radicalement opposées. Ainsi, par exemple, les phrases *"Il n'est pas très bon ce film .."* et *"Il est très bon ce film n'est ce pas"*, bien qu'antagonistes, sont pourtant quasiment indiscernables avec une mesure de distance, et établissent clairement les limites applicatives du modèle. Néanmoins, la méthode de recherche par similarité étant à la fois très répandue, et peu coûteuse en terme de temps de calcul, il nous a semblé important de l'exploiter ici pour en délimiter la portée, dans le cadre applicatif de DEFT07. La recherche par similarité telle qu'elle existe dans les moteurs de recherche documentaire sous la forme $Distance(\textbf{\textit{requête}}, document_k)$, peut être adaptée à un système de classification thématique $Distance(classe_k, document)$ selon le modèle suivant :

– Considérant les vecteurs $\vec{A_k}$, représentant k classes d'opinion.
– Considérant un vecteur $\vec{D}$, représentant un document à classer d'après $\vec{A_k}$.

Les vecteurs $\vec{A_k}$ et $\vec{D}$ sont composés de poids des mots contenus dans les documents qu'ils représentent. Ces poids sont calculés par la formule TF.IDF (Salton & Buckley, 1988) :

$$w_{ti} = tf_{ti}.idf_{ti} = tf_{ti}\left(log\left(\frac{N}{df_{ti}}\right) + 1\right)$$

où pour $\vec{A_k}$ et $\vec{D}$:

- w_{ti} est le poids du terme ti dans le document où dans la classe
- tf_{ti} dit Term Frequency est le nombre d'apparitions du terme ti dans le document où dans la classe
- N étant le nombre de documents composant le corpus, ce qui peut être ramené ici à k pour les vecteurs de classes $\vec{A_k}$ et 1 pour le vecteur de document à classer $\vec{D}$
- df_{ti} étant le nombre de documents qui contiennent ti dans le corpus d'apprentissage, pour la classe k

Considérant que tf_{ti} est équivalent à la probabilité de voir apparaître le mot w dans un document sachant sa classe, soit $p(w|\vec{A_k})$, on en déduira qu'en sélectionnant dans une classe k les mots ou objets textuels les plus représentatifs de l'opinion qu'elle caractérise, on maximisera le caractère discriminant de son vecteur $\vec{A_k}$. En théorie, et dans le cadre d'un système de mesure de similarité cosine appliqué à la recherche d'opinion, l'objectif sera donc de sélectionner les termes w les plus fréquents dont la probabilité d'appartenir à une classe $\vec{A_k}$ et à aucune des autres classes est la plus élevée.
Pour mesurer le degré de similarité entre $\vec{A}$ et $\vec{D}$, on reprend le principe des calculs de similarités sur des données numériques qui peuvent être ramenés au calcul du produit scalaire sur les deux vecteurs $\vec{D}$ et $\vec{A_k}$. Lorsque la norme euclidienne est choisie pour normaliser les composants des vecteurs $\vec{A}$ et $\vec{D}$, le calcul du produit scalaire se ramène à celui du cosinus. Soit :

$$cosine(\vec{A}, \vec{B}) = \frac{\vec{A}.\vec{B}}{\|\vec{A}\|.\|\vec{B}\|}$$

Et dans le cadre de DEFT07, pour chercher la classe k qui correspond au document représenté par $\vec{D}$ à maximiser le calcul du cosinus soit :

$$\vec{A}(k) = Argmax_{A_i}\left(cosine(\vec{D}, \vec{A_i})\right)$$

Si l'efficacité des familles de mesures qui découlent de ces applications est démontrée dans le cadre de moteurs de recherche documentaire, se posait pour nous la question de savoir si une opinion bonne, mauvaise ou moyenne pouvait se mesurer avec elles.
Pour maximiser le caractère descriptif des valeurs de TF.IDF des mots représentés par les vecteurs $\vec{A_k}$ on cherche généralement à réduire l'espace de représentation du vocabulaire pour ne conserver dans cet espace que des objets textuels caractéristiques. On adopte ainsi un ensemble de possibilités de filtrage du texte que l'on activera ou non, en fonction de la qualité du F-Score obtenu avec les données de vérification :

- Réduction du langage par suppression des mots outils, non représentatifs d'une opinion, avec un anti-dictionnaire[3].
- Réduction des mots à leurs lemmes pour regrouper le vocabulaire porteur du même sens (ex "bon", "bonne") en utilisant un lemmatiseur[4].
- Suppression des noms propres, à priori non porteurs de sens pour décrire une opinion, en utilisant les lettres majuscules en tant que repères.
- Combinaison de mots consécutifs sous forme de n-grammes pour conserver des séquences porteuses d'opinions (ex "Article de qualité").

A cette réduction de l'espace de représentation par filtrage du vocabulaire on peut aussi adjoindre des méthodes de valorisation de certains mots ou groupes de mots qu'on espère porteurs d'informations. On proposera pour cela les possibilités suivantes :

- Utilisation d'une partie seulement du texte pour construire les classifieurs, et classer les documents : on retient ainsi le postulat que, dans un article journalistique, l'opinion est généralement exprimée dans le chapô, l'introduction ou la conclusion. On tente donc de réduire la portion de texte à sa partie significative, pour supprimer autant que possible les phrases non porteuses d'opinion, et donc susceptibles de bruiter le classifieur. Cette option est donnée sous forme de pourcentages du texte à prendre en tête et en fin de document[5].

[3] On utilisera ici l'antidictionnaire de Jean Veronis, présenté sur http ://www.up.univ-mrs.fr/veronis/logiciels/index.html.

[4] Le lemmatiseur mis en oeuvre est celui élaboré par Benoit Favre au LIA.

[5] On notera que nous avons complété cette option par un seuil minimal de mots à prendre dans le document pour éviter les réductions excessives pour les cas où les textes sont de petite taille.

– Nous intégrons la possibilité de suppression des intersections entre classes, soit $\forall i \neq j,\ Ci \cap Cj = \emptyset$ où Ci, Cj sont des classes. On retire tous les objets (mots ou n-grammes) présents dans plus d'une classe pour étendre le caractère discriminant des classifieurs.

Pour obtenir le meilleur compromis, nous combinons tous ces paramètres et mesurons pour chaque classifieur le résultat obtenu par F-Score.

2.2 Classification par régression logistique

Dans son expression mathématique, l'analyse par régression examine la relation entre une variable dépendante (la variable de réponse) et des variables indépendantes particulières (les prédicteurs). Dans notre modèle, la variable de réponse est binaire. Mais cette variable est égale à 1 si le document en cours de comparaison est bien celui correspondant au modèle de la classe en cours d'examen. La variable est égale à 0 dans tous les autres cas.

Nous pouvons donc dire en résumé que le coeur de la régression logistique réside dans la définition d'un jeu de variables indépendantes. Ces variables de prédiction doivent être calculées uniquement en utilisant le contenu du document. Nous utilisons d'ailleurs dans cette version de notre algorithme la fréquence de certains mots dans les différentes classes de corpus déjà étiquetées, mise en rapport avec le nombre total de mots contenus dans le texte. Considérant ces aspects, nous utilisons bien la régression logistique en tant que méthode centrale de notre algorithme.

Cette idée est venue des travaux de l'un des auteurs, qui met en oeuvre le modèle de régression dans les applications de recherche opérationnelle. La méthode est utilisée pour réduire de manière astucieuse le nombre de variables entières d'un problème de réorganisation d'horaires de transport ferroviaire. (Mages, 2006). L'état de l'art de cette méthode laisse apparaître un vaste champ d'application.

Au cours des années précédentes, la régression logistique à été mise en oeuve dans le cadre de très nombreuses applications, répondant à des secteurs d'activités très variés : en médecine, (T. Cleophas, 2006; G. Venkataraman, 2006; G. Wu, 2006) ; dans le cadre des sciences de la vie et biomédicales (Oexle, 2006; D. Testi, 2001) ; Sciences du comportement (A. Menditto, 2006; B. Rosenfeld, 2005) ; Sciences sociales et de la législation (S. Wasserman, 1996; G. Robins, 1999; S. Stack, 1997) ; les sciences de la terre et de l'environnement (N. Sahoo, 1999; K. Chau, 2005; W. Wilson, 1996) ; le monde des affaires et de l'économie (Rodriguez, 2001; N. Dolsak, 2006) ; les sciences de l'informatique et de l'ingénieur (M. Collins, 2002; B. Jiang, 2004; J. Colwell, 2005) ; et finalement dans l'industrie et les sciences de la matière (A. Marinichev, 2005; A. Valero, 2006).

Tous ces travaux utilisent la régression logistique pour produire des équations prédictives. Dans la plupart de ces applications, la finalité est de démontrer que certains facteurs sont significatifs et d'exprimer l'importance de ces facteurs par une variable de réponse binaire. L'analyse par régression obtient de très bons résultats dans tous les domaines que nous venons de présenter. A notre connaissance, elle n'a pas encore été mise en oeuvre dans les activités de fouille de textes et plus généralement de TALN, en particulier dans les problèmes de classification, pour lesquels elle nous semble pourtant particulièrement appropriée.

2.2.1 Modèle de régression logistique pour extraire et classer une opinion

Selon le modèle de la régression logistique, nous commençons par définir dans notre méthode des ensembles et des index :

i : Index des documents.
j : Index des classes
C : Ensemble de catégories.
T : Ensemble de documents d'apprentissages

L'analyse par régression logistique permet de rechercher et d'estimer des modèles de regression multiple quand la réponse attendue est dichotomique, et peut être de type booléen. On utilise généralement la régression logistique lorsque l'on souhaite modéliser une question statistique qui peut être résumée par une réponse de type *"l'évenement a eu lieu/l'évenement n'a pas eu lieu"* et de manière plus générale, toute question à deux issues. Dans cet esprit, notre modèle aura deux possibilités de réponses : lorsqu'il cherche

à affecter un texte à une classe d'opinion, il répondra soit 1 lorsque la classe en cours de test est la bonne, soit 0 pour tous les autres cas.

La dépendance dichotomique (résultat binaire d'après une expérience ou une observation), implique que la variable de dépendance peut prendre une valeur de 1 avec une probabilité de succès évaluée par θ, ou une probabilité de défaut de $1-\theta$ si la valeur de la variable est 1. Nous sommes donc ici dans le cadre applicatif d'une variable de Bernouilli. Ce qui conduit à définir pour nos besoins un estimateur θ_{ij} tel que :

$$\theta_{ij} : \text{ estime la probabilité que le document } i \text{ soit apparenté à la classe } j$$

On voit ainsi que les variables explicatives sont indépendantes et peuvent prendre n'importe quelle forme. La régression logistique ne propose aucune hypothèse quand à la distribution des variables indépendantes. Elles ne sont pas nécessairement distribuées selon une loi normale, reliées linéairement ou de variances équivalentes à l'intérieur de chaque groupe.

En conséquence, il n'existe aucune règle pour définir les facteurs, pas plus qu'il n'est possible d'affirmer que chaque document correspondant à une classe, possède une proportion identique ou proche de mots spécifiques (et donc critiques pour la segmentation). Nous prenons acte de cette particularité sous la forme d'une variable indépendante explicative.

Par ailleurs, nous incluons le nombre total de mots contenus dans le document dans une autre variable explicative, car nous postulons que la classification pourra être expliquée, au moins en partie, par la prise en compte du volume du document. Dans ce contexte, les variables explicatives sont :

$$z_{ij} : \text{Le nombre de mots critiques dans le texte } i \text{ de la catégorie } j$$
$$y_i : \text{Le nombre total de mots dans le texte } i$$

Nous savons aussi que la relation entre les variables explicatives et les variables de réponse binaires n'est pas une fonction linéaire en régression logistique. Nous utilisons donc la fonction de régression logistique qui est une transformation logit de :

$$\theta_{ij} = \frac{e^{\left(\alpha_j+\gamma_j y_i+\sum\limits_{k\in C}\beta_j^k z_{ik}\right)}}{1+e^{\left(\alpha_j+\gamma_j y_i+\sum\limits_{k\in C}\beta_j^k z_{ik}\right)}} \tag{1}$$

Où :

α_j : Constante de l'équation $\forall j \in C.$ (2)

β_j^k : Coefficient des variables de prédiction z_{ik}. Valide pour le modèle qui évalue la catégorie j $\forall j \in C.$ (3)

γ_j : Coefficient des variables de prédiction y_{ij} $\forall j \in C.$ (4)

Il en résulte que pour chaque corpus, il est nécessaire de calculer $|C|$ modèles de régression différents. Dès que tous les paramètres de chaque modèle de régression ont été estimés, il devient possible d'appliquer l'équation (1) pour évaluer la probabilité qu'un texte i appartienne à une classe j. Ainsi, pour un document donné, nous avons une probabilité calculée de son appartenance, pour toutes les classes possibles. Finalement, le critère pour assigner une classe à un document sera la plus grande probabilité obtenue d'appartenance du texte à cette classe.

$$j^* = \arg\{\max_{j\in C}\theta_{ij}\} \tag{5}$$

2.3 Classification par mesure de densité calcul de compacité

Les systèmes de question-réponse (QR) sont définis comme des systèmes de recherche orientés non plus pour fournir une réponse d'après une mesure de similarité entre un document et une requête, mais par une évaluation de l'adéquation entre le contenu sémantique d'une requête et des réponse possibles, extraites du corpus. On notera néanmoins que l'analyse du contenu sémantique est réduite à son strict minimum, c'est à dire aux structures morpho-syntaxiques et que le résultat de recherche repose quasi exclusivement sur des approches statistiques. On parle d'ailleurs de modélisation statistique du langage (Thierry Spriet, 1996).
En effet, pour répondre à une question, un système QR procède à une suite séquentielle de traitements qui sont autant d'enrichissements et de filtrages des questions et des réponses. L'enrichissement consistera dans un premier temps à étiqueter le plus finement possible la question et le corpus pour préparer la mise en relation de concepts similaires contenus à la fois dans la question et dans le corpus. Par "concepts", on entend des entités nommées, et par relation, on entend trouver dans un segment du corpus, une entité nommée cible susceptible de correspondre au concept formulé dans la requête. On peut imaginer par exemple que des questions, débutant par les termes *"Qui"*, *"A qui"* pris en tant que concepts, sont à mettre en relation avec des réponses contenant une entité nommée de type *"Personne"*. Le modèle d'algorithme utilisé dans les systèmes de question-réponse repose sur une mesure de densité pour la recherche de passages (Gillard *et al.*, 2006).
Le principe retenu est d'élaborer un score de densité qui permette d'identifier la zone d'un segment issu d'un texte qui présente le plus de similitudes avec une question. A l'intérieur de chaque document, une distance moyenne $\mu(o_i)$ est calculée entre l'occurence courante o_i et les occurences des autres objets de la requête. Le calcul de ce score est effectué pour chacun des "objets caractéristiques" o_i d'un document D. La pénalité p fixée empiriquement doit favoriser le score produit par une concentration (proximité forte) de quelques objets communs de la requête, plutôt qu'une proximité faible mais pour un grand nombre d'objets. Ainsi on a :

$$\textit{DensitéScore}(o_i, D) = \frac{log[\mu(o_i) + (card\{\bigcup_{o_i \in Q} o_i\} - card\{\bigcup_{o_i \in D} o_i\}) * p]}{card\{\bigcup_{o_i \in Q} o_i\}} \quad (6)$$

Suite à l'application de ce premier test, on obtient un ensemble d'entités réponses candidates (ERc) qui sont confrontées à des passages jugés informatifs, extraits du corpus. Ces passages sont le plus souvent obtenus en déplaçant une fenêtre de n mots sur le document dans lequel on recherche une réponse.
Il est proposé en tant que mesure de compacité pour la sélection de la réponse au sein de toutes les ERc proposées, d'associer à la mesure de densité, le critère du "Confidence Weighted Score" (Voorhees, 2006). L'idée est de considérer chaque occurence d'une ERc comme point zéro d'un repère (décrit également en tant que "centre" ou "centroïde" selon les auteurs), et la présence des mots de la question autour de ces ERc issues du corpus comme des indices de présence de réponses correctes. On cherche à retrouver ici des "sacs de mots" les plus compacts et complets provenant de la question, autour de l'ERc (Luhn, 1958). Ce calcul de compacité est défini par :

$$\textit{Compacité}(ER_{c_i}) = \frac{\sum_{X \in MQ} P_{X,ER_{C_i}}}{\mid MQ \mid} \quad (7)$$

où $p_{X,ERci}$ correspond à la précision mesurée à l'intérieur d'une fenêtre centrée sur l'ERc_i pour les mots non vides de la question, à l'intérieur d'un rayon R, fixé par l'occurence la plus proche X_p du mot X (Gillard *et al.*, 2006).
Notre idée est que les systèmes de question-réponse, en ce sens qu'ils peuvent êtres définis comme des systèmes de recherche d'information spécifiques, sont adaptable à une recherche d'opinion, si cette dernière est considéré comme une information spécifique, posée comme une question. On peut même envisager que, comme dans le cas des campagnes d'évaluation, la réponse fournie à une question posée puisse être de nature binaire (*"L'opinion de ce document est elle mauvaise"* $\Longrightarrow$ [oui/non]) ou factuelle (*"Cet article est il de qualité"* $\Longrightarrow$ [Bonne/Moyenne/Mauvaise]).
Dans le cadre applicatif qui nous intérèsse, le problème particulier posé est que la liste des questions qui permettrait de chercher une réponse, n'est pas préexistante, comme dans le cas des campagnes TREC par exemple. Nous devons donc adapter le fonctionnement du système de question-réponse pour qu'il soit en mesure de construire automatiquement et d'après le corpus, pour une classe d'opinion donnée, toutes les

questions qui pourraient être posées. Il faudra ensuite que ces questions soient formulées de telle manière que l'algorithme puisse fournir une réponse factuelle ou binaire.

2.3.1 Application d'un modèle de question réponse dégradé

La méthode que nous proposons est la suivante. On considère les questions comme n groupes de mots regroupés autour d'un objet M constitutif de l'expression d'une opinion O de classe k, O_k. Ces *"sacs de mots"* de chaque O_k, peuvent éventuellement être lemmatisés et traités par un anti-dictionnaire. On considérera ensuite l'ensemble des éléments contenus dans un sac O_{k_M} comme de possibles réponses caractéristiques d'une opinion donnée, lorsque l'on rencontrera l'objet M dans un segment de texte issu du corpus à classer. L'hypothèse que nous formulons est qu'un score de compacité calculé sur toutes les phrases sélectionnées, par ce que contenant M de O_k, permettra de localiser les meilleures réponses candidates à une question posée (par exemple *"Cet article est il bon ?"* pour localiser les documents de classes *"[Bon/Moyen/mauvais]"* du corpus relecture, pour l'objet M, *"article"*). On calculera le score de compacité pour chaque groupe de mot $x_i \in M$ dans chaque O_k, puis on considérera que la somme de tous les scores de compacité obtenus avec tout O_{k_M} pour chaque O_k indique à quelle classe appartient le document.

Pour classer un document Y, on recherchera dans ce documents tous les segments y d'une longueur de n mots, contenant en leur centre (soit $y_{n/2}$) une occurence de l'objet M pour une opinion O_k. Tous ces segments seront considérés comme autant d'entités réponses candidates y pour Y. Puis on calculera le score de compacité par :

$$\textit{compacité}(y \mid O_k) = \frac{\sum\limits_{X \in O_{k_M}} P_{X,y}}{\mid O_{k_M} \mid} \tag{8}$$

Ou pour tout mot X de O_{k_M} présent dans y on mesure la distance Δ qui le sépare du centre représenté par M. Dans notre application, Δ est égale au nombre de mots qui séparent X de M dans y. On notera que pour construire les listes de questions de O_{k_M}, nous avons besoin d'identifier les objets centroïdes, équivalent des entités nommées "source". Cet objet M est représenté par des mots tels que "article" ou "papier" dans le cas du corpus "relectures". Ces sources devront être accompagnées des cibles les plus porteuses de sens lorsqu'elles sont associées à l'objet M : le plus souvent des adjectifs qualificatifs dans le cas d'une opinion, par exemple "bien", "bon", "mauvais", "favorable" dans un modèle parfait. Ces cibles sont regroupées dans les sacs de mots O_{k_M}. Ce qui explique pourquoi nous avons qualifié ce modèle de "dégradé" (ou restreint) : il n'implique pas de procéder à un étiquetage du corpus pour définir les objets M qui serviront à construire les sacs de mots O_{k_M}. En effet, par défaut, deux étiquettes morpho-syntaxiques sont suffisantes dans notre modèle : *"Objet d'opinion qualifié par des adjectifs"*, et *"Adjectif qualificatif de l'objet d'opinion"*.

2.3.2 Localisation automatique des objets

Pour localiser les objets M caractérisant les sacs de mots, nous avons considéré que les mots les plus fréquents porteurs de sens sont ceux de plus forte occurence subsistant après application d'un antidictionnaire, pour chaque classe. Dans le corpus "Avoir à Lire" par exemple, les mots les plus fréquents seront notamment *"Film"* et *"Album"*. Dans le corpus *"Relectures"*, c'est le mot *"article"* qui ressort, *"loi"* pour les *"Débats parlementaires"*, etc. On notera que les mots objets peuvent être les mêmes d'un sous-corpus d'opinion à un autre.
Nous recherchons ensuite autour de ces objets M ou entités nommées "source", l'ensemble de mots non outils "cible" susceptible de qualifier la source. Cette recherche se fait automatiquement en définissant un périmètre exprimé en nombre de mots autour de l'entité nommée source (n mots précédent et n mots suivant), et en ne conservant dans ce périmètre que les mots "non outils". Cette opération est menée sur chaque sous-corpus correspondant à une classe d'opinion (exemple *"Relecture/favorable"*, *"Relecture/défavorable"*, *"Relecture/moyen"*). A la suite de cette opération nous possédons n listes de phrases "réponses possibles" associées à l'ensemble des objets M correspondants aux classes O_k de chaque corpus.

On remarque que la relation entre les objets et leurs qualificatifs n'étant pas d'ordre sémantique, mais exclusivement statistique (extraction par comptage d'occurences), il est fréquent qu'un même ensemble M_n soit présent simultanément dans deux classes (exemple "Une bonne idée mais un article de mauvaise qualité" devient l'élément M'_n "bonne idée article mauvaise qualité" dans $O_{k'_{M'}} =$ *défavorable*, et la

phrase "Une mauvaise idée pour un article de bonne qualité" est représenté sous la même forme M_n dans $O_{k_M} =$ *favorable*, ce qui revient dans ce cas à $P(M_n \mid O = favorable) = P(M'_n \mid O =$ *défavorable*$)$. On considérera que le système peut compenser de lui-même la présence de ces résultats antagonistes si les sacs de mots construits autours des M sont les plus exhaustifs possible pour chaque O_k. Nous avons également vérifié que l'on augmentait $p(M_n \mid O_k)$ en insérant dans les sacs de mots des groupes composés de bigrammes, qui restituent dans leur classe la localisation des mots les uns par rapports aux autres (exemple *"est_bon"* ou *"pas_bon"*).

3 Résultats obtenus

3.1 Resultats du modèle par similarité

L'aprentissage a été conduit par une séparation du corpus en deux. La première moitié du corpus a été utilisée pour l'entrainement par le classifieur[6]. La seconde partie a été utilisée pour calculer les F-Score. Pour les options optimalement choisies, les scores obtenus en phase d'entrainement et en phase d'évaluation DEFT07 sont présentés dans le tableau 2.

Corpus	Lem	Antidico	$\cup = 0$	npr	pct	ngrams	FS Train	FS DEFT07
Avoir à Lire	oui	oui	non	oui	20/30	3	0.50	0.37 (0.48)
Jeux vidéos	oui	oui	oui	non	20/30	3	0.73	0.62 (0.65)
Relectures	oui	oui	non	oui	50/50	3	0.37	0.43 (0.46)
Débats	non	non	oui	non	20/30	3	0.63	0.61 (0.64)

TAB. 2 – Résultats obtenus sur chaque corpus et options utilisées pour les obtenir

Le tableau 2 présente les options retenues pour obtenir les F-Scores indiqués. On trouve respectivement : lemmatisation des mots (options oui,non), application d'un antidictionnaire (oui, non), $\cup = 0$ supression des intersections (oui, non), *npr* correspond à la supression de noms propres (oui,non), *pct* décrit les pourcentages du document pris en tête du document à classer et à la fin, *ngrams* décrit le nombre de ngrammes ajoutés au texte. Dans la dernière colonne, nous indiquons le F-Score obtenu sur le corpus d'apprentissage après séparation en deux parts égales, le F-Score final obtenu sur les données de DEFT07 et entre parenthèses le F-Score moyen des participants de DEFT07.

Ces paramètres sont obtenus par une évaluation systématique conduite sous forme d'un processus de recherche du maximum de F-Score pour tous les corpus C :

$$MaxF - Score(C) \tag{9}$$

Par itérations successives pour toutes les combinaisons des valeurs de variables de décision suivantes :

- lemmatisation=$\{0, 1\}$
- antidictionnaire=$\{0, 1\}$
- supression des intersections=$\{0, 1\}$
- supression des noms propres=$\{0, 1\}$
- supression des noms propres=$\{0, 1\}$
- pourcentage de texte pris en tête=$[0, 50]$[7]
- pourcentage de texte pris en fin=$[0, 50]$
- ngrammes=$\{1, 3\}$

On note que les résultats de DEFT07 obtenus après reconstruction des classifieurs d'après le corpus d'aprentissage complet en conservant les meilleurs paramètres, sont relativement éloignés de ceux obtenus sur les jeux d'entrainements. Ceci est très probablement dù un surapprentissage, imputable au choix de 50% du corpus d'entrainement utilisé pour valider les paramètres. Nous renouvelerons prochainement nos

[6]Le classifieur utilisé, écrit en java, est disponible sur www.echarton.com/deft, associé au programme perl de traitement préalable

[7]Les valeurs de pourcentages sont discrètes et prises par incrément de 5

expériences en améliorant notre méthode (évaluation du classifieur sur un corpus d'entrainement segmenté en 4 éléments), pour confirmer nos meilleurs scores.
Ces tests nous ont confirmé que l'expression de l'opinion ne peut être évaluée statistiquement par similarité cosine ou mesure de probabilité, qu'en supprimant le bruit et en tentant de localiser au mieux les segments d'un document où s'exprime cette opinion. Ils nous incitent aussi à penser que les performances d'un classifieur par mesure de similarité ou de distance dans ce type de tâche, sont particulièrement dépendantes de la forme d'expression (complexité, richesse de vocabulaire) utilisée.

3.2 Resultats du modèle de régression logistique

Dans cette section, nous détaillons les résultats obtenus avec notre algorithme par régression logistique appliqué aux différents corpus proposés par DEFT07 (DEFT, 2007). Les outils utilisés pour expérimenter ce modèle sont :

- Microsoft Visual Studio 2005 et Visual C# 2005 [8] principalement pour extraires les mots et les insérer dans la base de données.
- Microsoft Access 2003 [9] en tant que moteur de base de données et pour exécuter les requêtes SQL.
- Statgrapichs 4.0 [10] utilisé pour l'analyse par régression.

3.2.1 Corpus 1 : Avoir A Lire, critiques de produits culturels

En utilisant les valeurs calculées des paramètres (voir la table TAB.3) du modèle le plus vraisemblable :

$$\theta_{i0} = \frac{e^W}{1+e^W}$$

où :

$$W = -0.27299 + 0.16782 z_{i0} - 0.23223 z_{i1} - 0.16084 z_{i2} + 0.00565 y_i$$

Ces valeurs s'inscrivent dans le cadre de notre hypothèse initiale. Il y a bien une corélation entre la probabilité de classifier en catégorie 0, avec la quantité critique de mots de catégorie 0 présents dans le document évalué (i.e. le parâmètre $\beta_0^0 > 0$). On constate également que dans le même temps, la quantité de mots critiques contenus par ailleurs dans d'autres classes fait décroitre la probabilité.

Paramètre	Estimation	Standard Error	Odds Ratio estimé
α_0	-0.272992	0.282966	
β_0^0	0.167818	0.0144647	1.18272
β_0^1	-0.232226	0.0273179	0.79276
β_0^2	-0.160838	0.0126263	0.85143
γ_0	0.005654	0.0016795	1.00567

TAB. 3 – Modèle de régression estimé pour la catégorie $j = 0$ dans le corpus Avoir A Lire, DEFT07

On déduit que si la "P-value" pour le modèle, dans l'analyse de la déviance (lire table TAB.4) est inférieure à 0,01, il existe une relation statistiquement significative entre les variables, dans un intervalle de confiance à 99% . Par ailleurs, la "P-Value" pour les données résiduelles est supérieure à 0,10, indiquant que le modèle n'est pas significativement plus mauvais que le meilleur modèle possible pour ces données, dans un intervalle de confiance de 90 % (ou supérieur).

Source	Deviance	Degrés de liberté	P-Value
Modèle	774.97	4	0.0000
Résiduel	971.12	2069	1.0000
Total	1746.1	2073	

TAB. 4 – Analyse de la déviance pour la catégorie $j = 0$ dans le corpus 1, DEFT07

[8] Informations sur http ://www.microsoft.com

[9] Informations sur http ://www.microsoft.com)

[10] Edité par StatPoint, Inc. link http ://www.statgraphics.com

En cherchant à évaluer comment le modèle peut être simplifié, nous notons que la plus haute valeur pour le test de vraisemblance (likelihood ratio test) est de 0,0007 (Voir table TAB.5), lorsqu'il est associé à y_i (le nombre total de mots contenus dans le texte). Sachant que la "P-value" est inférieur à 0,01, on considère que ce terme est statistiquement significatif à un intervalle de confiance de 99% . En conséquence, nous ne retirons aucune variable du modèle.

Facteur	χ_2	Degrés de liberté	P-Value
z_{i0}	170.08	1	0.0000
z_{i1}	87.97	1	0.0000
z_{i2}	231.91	1	0.0000
y_i	11.40	1	0.0007

TAB. 5 – Test de vraisemblance (Likelihood ratio tests) pour la catégorie $j = 0$ dans le corpus Avoir A Lire, DEFT07

Ces procédures de calcul et d'analyse du modèle de régression ont été répétées pour les classes 1 et 2 (*moyen* et *bon*). Le tableau TAB.6 montre la validité du modèle de régression appliqué au corpus Avoir A Lire. Le tableau TAB.7 présente les principaux aspects de ce corpus et nos scores finaux.

Categorie	P-Value du modèle	Pourcentage de déviance expliquée par le modèle
0	0.0000	44.38
1	0.0000	20.88
2	0.0000	30.80

TAB. 6 – Validité du modèle de régression pour le corpus Avoir A lire, DEFT07

Item	Valeur
$\|T\|$	2074
n	2074
$\|C\|$	3
$F - Score$ Corpus de test (cette méthode)	0.50
$F - Score$ Corpus de test (moyenne DEFT07)	0.48

TAB. 7 – Résultats finaux pour le corpus Avoir A Lire, DEFT07

3.2.2 Corpus 2 : JeuxVideo.com, Tests de jeux vidéo

Les résultats obtenus sur le corpus 2 sont présentés dans les tableaux qui suivent. Le tableau TAB.8 présente la validité du modèle de régression appliqué.

Category	P-Value du modèle	Pourcentage de déviance expliquée par le modèle
0	0.0000	88.02
1	0.0000	73.73
2	0.0000	90.30

TAB. 8 – Validité du modèle de régression pour le corpus JeuxVidéo.com, DEFT07

Le tableau TAB.9 présente les principales caractéristiques de notre modèle appliqué au corpus et le score final.

Item	Valeur
$\|T\|$	2537
n	60
$\|C\|$	3
$F - Score$ Corpus de test (cette methode)	0.46
$F - Score$ Corpus de test (moyenne DEFT07)	0.65

TAB. 9 – Résultats finaux pour le corpus JeuxVideo.com, DEFT07

3.2.3 Corpus 3 : Relectures d'articles de conférences

Les résultats pour le corpus "Relectures sont présentés dans les tables qui suivent. La table 10 présente la validité du modèle de régression appliqué.

Categorie	P-Value du modèle	Pourcentage de déviance expliquée par le modèle
0	0.0000	29.58
1	0.0000	24.11
2	0.0000	21.12

TAB. 10 – Validité du modèle de régression pour le corpus Relectures, DEFT07

Le tableau TAB.11 présente les principales caractéristiques de notre modèle appliqué au corpus et le score final.

Item	Value
$\|T\|$	881
n	881
$\|C\|$	3
$F - Score$ Corpus de test (cette méthode)	0.47
$F - Score$ Corpus de test (Moyenne DEFT07)	0.47

TAB. 11 – Résultats finaux pour le corpus Relectures, DEFT 2007

3.2.4 Corpus 4 : Débats parlementaires

Les résultats pour le corpus 4 sont présentés dans les tableaux qui suivent. Le tableau TAB.12 présente la validité du modèle de régression appliqué.

Categorie	P-Value du modèle	Pourcentage de déviance expliquée par le modèle
0	0.0000	24.20
1	0.0000	24.20

TAB. 12 – Validité du modèle de régression pour le corpus Débats, DEFT07

Le tableau 13 présente les principales caractéristiques de notre modèle appliqué au corpus, et le score final.

Item	Valeur
$\|T\|$	17299
n	1000
$\|C\|$	2
$F - Score$ Corpus de test (cette méthode)	0.55
$F - Score$ Corpus de test (moyenne DEFT07)	0.64

TAB. 13 – Résultats finaux pour le corpus Débats, DEFT07

3.3 Résultats du modèle par densité et compacité

Notre algorithme de recherche d'opinion par calcul de compacité repose sur la définition de sous-classes de mots centroïdes (ou "attracteurs"), c'est à dire susceptibles d'être entourés par un ensemble de mots porteurs d'une opinion. Ces centroïdes sont accompagnés par des sacs de mots que nous avons remplis de tous les mots qui les entourent (portée de $+/-$ 5 mots avant et après) dans le corpus d'aprentissage. Il existe un centroïde pour chaque sous corpus d'opinion (exemple *centroïde "article"* classe *bon*, *centroïde "article"* classe *mauvais*, etc).

Nous indiquerons ici les scores de précision et rappel obtenus pour chaque classe car ils soulignent un déséquilibre entre les performances des classifieurs pour chaque opinion : certaines classes sont très bien détectées et d'autres ont des scores extrèmement faibles. Il semble après quelques expériences complémentaires que ce déséquilibre soit lié au choix de sacs des mots non individualisés par classes. Ceci nous conduira à envisager l'hypothèse que les sacs de mots devraient être construits d'après les n mots centroïdes les mieux placés pour chaque classe (et non plus choisis sur tout le corpus, de manière inter-classes).
Les résultats F-Score obtenus ici sont ceux obtenus d'après le corpus d'entrainement, divisé en 50% pour l'apprentissage, et 50% pour le test. Nous n'avons pas soumis ces résultats lors de l'évaluation DEFT07. [11].

3.3.1 Résultats sur le corpus Débats

Les mots centroïdes retenus pour la constitution de sacs de mots, après suppression des mots outils par application d'antidictionnaire, et comptage des occurences de mots restants sont : $\{Loi, projet\}$.

Classe	0 (Défavorable)	1 (Favorable)	F-Score
Précision	0.55	0.48	0.51 (0.64)
Rappel	0.89	0.11	

TAB. 14 – Scores obtenus sur le corpus Débats (entre parenthèses, le score moyen obtenu par les participants de DEFT07

3.3.2 Résultats sur le corpus Relectures

Les mots centroïdes retenus pour la constitution de sacs de mots, après suppression des mots outils par application d'antidictionnaire, et comptage des occurences de mots restants sont : $\{Article, Papier\}$.

Classe	0 (Défavorable)	1 (Moyen)	2 (Favorable)	F-Score
Précision	0.54	0.09	0.66	0.42 (0.46)
Rappel	0.19	0.46	0.57	

TAB. 15 – Scores obtenus sur le corpus Relectures(entre parenthèses, le score moyen obtenu par les participants de DEFT07

3.3.3 Résultats sur le corpus Avoir à Lire

Les mots centroïdes retenus pour la constitution de sacs de mots, après suppression des mots outils par application d'antidictionnaire, et comptage des occurences de mots restants sont : $\{Film, Roman\}$.

Classe	0 (Défavorable)	1 (Moyen)	2 (Favorable)	F-Score
Précision	0.43	0.35	0.49	0.40 (0.48)
Rappel	0.14	0.10	0.86	

TAB. 16 – Scores obtenus sur le corpus Avoir à Lire (entre parenthèses, le score moyen obtenu par les participants de DEFT07

[11] Les logiciels utilisés pour réaliser ces expériences sont des prototypes écrits en perl, spécifiquement pour ce défi. Ils sont disponibles sur http :/www.echarton.com/deft)

4 Conclusions et perspectives

Nous avons considéré dès le début de nos expériences ce défi sous l'angle de son hétérogénéité. Tant dans la forme des corpus (quatre familles de textes issus de sources très variées), que dans la méthode d'évaluation des résultats (trois soumissions prises sous une forme atomique), ce défi invite à rechercher une solution neuve d'extraction d'une opinion, susceptible de généraliser ce type de tâche.
La confrontation des résultats obtenus avec une première méthode classique de la recherche documentaire - similarité cosine et probabilité de présence des mots - sur les corpus d'apprentissage, avec ceux obtenus par les deux autres méthodes - régression logistique et calcul de compacité - nous a semblé très encourageante, notamment si l'on considère que les résultats sont globalement proches de ceux obtenus en moyenne par l'ensemble des participants. Par ailleurs, les méthodes de régression logistique et de mesure de compacité n'ont encore - à notre connaissance - jamais été déployées pour classer des corpus d'après des "concepts" ou des "idées" (activité dont nous semble se rapprocher la thématique de l'opinion). De nombreuses possibilités d'améliorations restent à explorer.
Il reste par exemple possible d'améliorer considérablement le modèle de régression, en ajoutant d'autres variables explicatives, en passant d'une exploration des mots critiques à une étude des phrases critiques, ou encore en introduisant une utilisation des modèles n-grammes.
Il en va de même avec le modèle de mesure de densité et de calcul de compacité. Nous pourions enrichir le corpus avec un étiquetage morphosyntaxique orienté vers l'expression d'une idée, ou encore, affiner la recherche de mots centroïdes pour la construction des sacs de mots.
D'une manière générale, nous pensons que l'introduction d'un nouveau modèle statistique (la régression logistique), et l'adaptation d'un dispositif prévu à l'origine pour les systèmes de question-réponse à la classification par opinion, ouvre des pistes intéressantes dans le domaine de la classification par les idées.

5 Remerciements

Nous souhaitons adresser nos chaleureux remerciements à nos encadrants respectifs, Messieurs Philippe Michelon, et Jean-François Bonastre pour leurs encouragements et leur bienveillance dans le cadre de cette participation à DEFT07 qui s'éloigne singulièrement des recherches que nous sommes censés réaliser !
Nous remercions également Messieurs Frédéric Béchet et Juan Manuel Torres-Moreno, pour le temps qu'ils ont bien voulu nous consacrer.

Références

A. MARINICHEV, S. VYAZ'MIN I. D. (2005). A spectrophotometric study of solid. *Russian Journal of Applied Chemistry*, **vol 78 issue 10 p1662-1667**.

A. MENDITTO, D. LINHORST J. C.-N. B. (2006). The use of logistic regression to enhance risk assessment and decision making by mental health administrators. *The Journal of Behavioral Health Services and Research*, **vol 33 issue 2 p213-224**.

A. VALERO, E. CARRASCO E. A. (2006). Growth/no growth model of listeria monocytogenes as a function of temperature, ph, citric acid and ascorbic acid. *European Food Research and Technology*, **vol 224 issue 1 p91-100**.

B. JIANG, C. WANG P. C. (2004). Logistic regression tree applied to classify pcb golden finger defects. *The International Journal of Advanced Manufacturing Technology*, **vol 24 issue 7 p496-502**.

B. ROSENFELD C. L. (2005). Assessing violence risk in stalking cases : A regression tree approach. *Law and Human Behavior*, **vol 29 issue 3 p342-357**.

D. TESTI, A. CAPPELLO L. C. M. V. S. G. (2001). Comparison of logistic and bayesian classifiers for evaluating the risk of femoral neck fracture in osteoporotic patients. *Medical and Biological Engineering and Computing*, **vol 39 issue 6 p633-637**.

DEFT (2007). 3ème dÉfi fouille de textes, http ://deft07.limsi.fr/index.html.

FAVRE B., BECHET F. & NOCÉRA P. (2005). Robust named entity extraction from large spoken archives. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, p. 491–498, Vancouver, British Columbia, Canada : Association for Computational Linguistics.

G. ROBINS, P. PATTISON S. W. (1999). Logit models and logistic regressions for social networks : Iii. valued relations. *Psychometrika*, **vol 64 issue 3 p371-394**.

G. VENKATARAMAN, V. ANANTHANARAYANAN G. P. E. A. (2006). Morphometric sum optical density as a surrogate marker for ploidy status in prostate cancer : an analysis in 180 biopsies using logistic regression and binary recursive partitioning. *Virchows Archiv*, **vol 449**.

G. WU S. Y. (2006). Prediction of possible mutations in h5n1 hemagglutitins of influenza a virus by means of logistic regression. *Comparative Clinical Pathology*, **Vol 15**(issue 4), p255–261.

GILLARD L., BELLOT P. & EL-BÉZE M. (2006). Influence de mesures de densité pour la recherche de passages et l'extraction de réponses dans un système de questions-réponses. In *Actes de Coria 2006*, p. 193–204, Lyon, France.

J. COLWELL A. R. (2005). Hot surface ignition of automotive and aviation fluids. *Fire Technology*, **vol 41 issue 2 p105-123**.

K. CHAU J. C. (2005). Regional bias of landslide data in generating susceptibility maps using logistic regression : Case of hong kong island. *Landslides*, **vol 2 issue 4 p280-290**.

LUHN H. (1958). The automatic creation of literature abstracts. In *IBM Journal of research and Development*.

M. COLLINS, R. SCHAPIRE Y. S. (2002). Logistic regression, adaboost and bregman distances. *Machine Learning*, **vol 48 issue 1 p253-285**.

MAGES (2006). Mages (modules d'aide à la gestion des sillons), http ://awal.univ-lehavre.fr/ lmah/mages/.

N. DOLSAK M. D. (2006). Investments in global warming mitigation : The case of activities implemented jointly. *Policy Sciences*, **vol 39 issue 3 p233-248**.

N. SAHOO H. P. (1999). Integration of sparse geologic information in gold targeting using logistic regression analysis in the hutti maski schist belt, raichur, karnataka, india. a case study. *Natural Resources Research*, **vol 8 issue 3 p233-250**.

OEXLE K. (2006). Biochemical data in ornithine transcarbamylase deficiency (otcd) carrier risk estimation : logistic discrimination and combination with genetic nformation. *Journal of Human Genetics*, **vol 51 issue 3 p204-208**.

RODRIGUEZ A. A. (2001). Logistic regression and world income distribution. *International Advances in Economic Research*, **vol 7 issue 2 p231-242**.

S. STACK O. T. (1997). Suicide risk among correctional officers : A logistic regression analysis. *Archives of Suicide Research*, **vol3 issue 3 p183-186**.

S. WASSERMAN P. P. (1996). Logit models and logistic regressions for social networks : I. an introduction to markov graphs andp. *Psychometrika*, **vol 61 issue 3 p401-425**.

SALTON G. & BUCKLEY C. (1988). Term-weighting approaches in automatic text retrieval. In *Information Processing and Management*.

T. CLEOPHAS A. (2006). Post-hoc analyses in clinical trials, a case for logistic regression analysis. *Statistics Applied to Clinical Trials*, **p 187-191**.

THIERRY SPRIET, FRÉDÉRIC BÉCHET, MARC .EL-BÈZE. C. D. L. L. K. (1996). Traitement automatique des mots inconnus. Avignon, France.

VOORHEES E. (2006). Overview of the trec 2002 question answering track. In *Actes de "the 11th Text REtrieval Conference"*, Gaithersburg, Maryland, USA : TREC.

W. WILSON, K. DAY E. H. (1996). Predicting the extent of damage to conifer seedlings by the pine weevil (hylobius abietis l.) : a preliminary risk model by multiple logistic regression. *New Forests*, **vol 12 issue 3 p203-222**.

Approches statistiques et SVM

Approches naïves à l'analyse d'opinion

Eric Crestan, Stéphane Gigandet et Romain Vinot

Yahoo! Inc., Paris,
{ecrestan,sg,romainv}@yahoo-inc.com

Résumé : L'analyse d'opinion semble être une tâche simple pouvant se rapporter au simple paradigme de classification. Cependant, cela s'avère plus compliqué notamment à cause de la classe d'opinion *neutre* qui peut être à la fois une combinaison de positif/négatif ou aucun des deux. Nous montrons dans cet article que des systèmes rudimentaires, donc l'un est basé sur des modèles SVM entrainés sur une sélection de traits saillants et l'autre, basé sur une sommation d'indices pondérés, obtiennent des scores honorables dépassant la moyenne des systèmes participants. De plus, il est à noter, que ces systèmes n'utilisent aucunes ressources externes.

Mots-clés : Analyse d'opinion, Support Vector Machine, Critère d'impureté de Gini

1 Introduction

L'analyse d'opinion est depuis longtemps une composante importante des systèmes de veille technologique et stratégique sur Internet. On conçoit aisément le besoin des grandes sociétés de surveiller leur image de marque dans la presse quotidien, mais également à travers les multiples blogs et forums disponibles sur Internet.

Le domaine de l'analyse d'opinion sur le Web a connu un essor plus important ces dernières années avec la possibilité d'obtenir des corpus déjà annotés à travers les sites de ventes en ligne. Les travaux de (Schein *et al.*, 2002; Kushal *et al.*, 2003) utilisent les revues de produits comme un tout afin d'entrainer des classifiers. D'autres systèmes considèrent des fenêtres de taille fixe autour d'entités nommées (Grefenstette *et al.*, 2001) ou encore, travaillent au niveau de la phrase (Wilson *et al.*, 2005). Cependant, la majorité de ces travaux ont été effectués sur la langue anglaise par manque de corpus annotés dans d'autres langues. La compagne DEFT'07 est donc la première à offrir un cadre d'évaluation pour l'analyse d'opinion sur une large diversité de type de documents en français.

Notre participation à cette campagne a été motivée par l'intérêt de connaître à quel point des systèmes simples peuvent performés sur une tâche supervisé d'analyse d'opinion. Les approches que nous proposons dans ces travaux sont basées, pour la première, sur les méthodes SVM couplées avec une sélection de traits fondée sur le saillance ; alors que notre second système se base sur la sommation des scores attribués aux couples trait/classe. Ces deux systèmes, qui reposent principalement sur l'heuristique et l'empirique, ne prétendent pas révolutionner le domaine, mais montrent qu'il est possible d'arriver à des résultats acceptables avec peu d'efforts.

Cet article est découpé en trois parties principales : Dans un premier temps, les deux approches seront présentées en détail. La seconde partie sera consacrée à l'évaluation de nos systèmes dans le cadre de la campagne DEFT'07. Enfin, nous terminerons cet article par une analyse détaillée des résultats en proposant des exemples concrets issus des jeux de test.

2 Description des approches

Les différents systèmes présentés dans cette section, sont basés sur des approches supervisées n'utilisant aucun lexique externe, si ce n'est une courte liste de mots-outil.

2.1 Approche SVM et sélection de traits

Les *Support Vector Machines* (SVM) sont devenus très populaires pour les systèmes de classification supervisée depuis leur application à ce domaine par (Joachims, 1997). Bien que cela soit une approche binaire, elle peut tout à fait être appliquée sur de grande dimension en utilisant un mode d'apprentissage *un contre tous* et en construisant donc un classifier par classe.

2.1.1 Application des SVM à l'analyse d'opinion

Dans le cadre de l'évaluation DEFT'07, la plupart des corpus sont basés sur une classification ternaire (*positive*, *négative* et *neutre*), ce qui nous oblige à entrainer plusieurs modèles. Seul le corpus de *débats parlementaires*, requière une classification binaire (*pour* / *contre*).

La logique en classification de document par SVM est d'avoir un classifier par classe. Toutefois, notre application est quelque que peu différente des schémas classiques car les classes représentent des tonalités et non des thèmes. De plus, il convient de s'interroger sur la sémantique des classes proposées. En effet, il semble difficile de trouver des termes décrivant la neutralité. Pour cette raison, seuls 2 classifiers seront créés, dont un servira à la détection des traits *positifs* (*positif* contre *négatif*+*neutre*), l'autre à la détection des traits *négatifs* (*négatif* contre *positif*+*neutre*). La classe *neutre* sera, quand à elle, affectée dans le cas ou ni le classifier *positif*, ni le classifier *négatif* n'auront un score dépasseront le seuil requit.

Dans le cadre de cette évaluation, nous avons mis en œuvre l'outil *mySVM* de (Stefan Rüping, 2000). La particularité de cet outil est qu'il n'accepte pas de grande dimension et nous avons donc dû nous contenter de 80 dimensions pour ces travaux.

2.1.2 Sélection des traits

Une des phases les plus importants lors de l'entrainement des SVM est la sélection des traits (ou *features*). La grande dimensionnalité qu'offrent les corpus, se comptant en milliers, ne permet pas une utilisation totale de ces dimensions. Il est donc indispensable de réduire la dimensionnalité en faisant une sélection des traits les plus porteurs d'information pour la tâche. De nombreuses approches ont été proposées par le passé, dont les plus populaires et performantes sont le test du χ^2 (Schütze *et al.*, 1995) et le gain d'information (Yang & Pedersen, 1997).

Pour notre part, nous proposons l'utilisation d'une variante du critère de divergence de Kullback–Leibler (Kullback & Leibler, 1951), ce critère est également appelé critère du gain d'information. En effet, une divergence trop importante entre la distribution d'un mot entre les classes, constitue un indice sur l'important de ce terme à discriminer une classe par rapport à une autre. Nous ferons référence par la suite à ce critère de divergence comme *score de saillance*, définit par :

$$S(t, C_i) = \left[P(C_i / t) - P(t)\right] \times \log\left(\frac{P(C_i / t)}{P(t)}\right) \qquad (1)$$

Le score de saillance peut donc être calculé pour chaque terme *t* appartenant à la classe C_i. Dans notre évaluation, le nombre de classe est de 2 (*positif* et *négatif*), le cas de la classe *neutre* n'étant pas considéré.

Finalement, seuls les 40 termes ayant les plus hautes saillances par classe seront retenus pour l'apprentissage des modèles, correspondant au final à un vecteur de 80 dimensions.

2.1.3 Prétraitement et Apprentissage

Avant même de pouvoir identifier les termes saillants inhérents à une classe, les documents doivent être segmentés. Cependant, n'utilisant pas de dictionnaire externe, une segmentation uniquement effectuée sur les unités lexicales peut éventuellement générer une perte d'information. Ceci est encore plus vrai dans le cadre de cette évaluation car les opinions *positives* ou *négatives* sont souvent exprimées par des négations (*ne … pas*) comportant donc plusieurs mots. Par exemple, le verbe *parvenir* à une toute autre signification si celui-ci est précédé de la particule *ne* ou pas. Pour cela, les scores de saillance sont

calculés pour les uni-grammes, ainsi que pour les bi-grammes. Les *termes* définis comme les plus saillants, sont donc en fait composés à la fois d'unis et de bi-grammes. Des exemples de termes saillants seront présentés dans la Section 4 de cet article.

Le corpus d'apprentissage est constitué de vecteurs de traits de 80 dimensions pour chaque document, dans lesquelles apparaissent des 1 ou des 0 selon la présence ou non du terme. Les modèles SVM peuvent ensuite être entrainés sur ces vecteurs.

Le décodage se fait très simplement en créant les vecteurs par la même méthode pour les documents de test. Ensuite, la classe obtenant le score supérieur à zéro le plus élevé est choisie. Si aucun des modèles SVM ne donnent un score positif, la classe *neutre* est alors choisie par défaut.

2.2 Approche par maximisation d'indices

L'approche exposée dans cette section est des plus triviales et ne prétend pas être reconnue comme *approche scientifique* à proprement parlé. Principalement basée sur l'observation, le système proposé ici ne reste qu'une ébauche étant donné le peu de temps que nous lui avons consacrée. Cependant, les résultats obtenus sont des plus encourageants et nous invite donc à pousser plus avant notre analyse afin d'établir scientifiquement ce qui a été créé empiriquement.

Cette approche consiste à sommer pour chaque terme d'un document, les scores représentant les gains d'information observés sur le corpus d'apprentissage. Cette approche avait déjà été employée par (Crestan, 2004) dans le cadre de l'évaluation en désambiguïsation sémantique SENSEVAL-3 et avait montrée des résultats comparables à d'autres approches comme les arbres de classification sémantique.

2.2.1 Présentation de l'approche

Le grand problème des approches comme les modèles SVM ou d'autres approches, est qu'il est indispensable de réduire la dimensionnalité de la tâche. Cela engendre généralement une perte d'information, qui est toutefois nécessaire par le fait que ces approches opèrent généralement une recherche d'optimal qui peut être très coûteux en temps de calcul. D'autres approches, comme les arbres de classification, divisent les populations à chaque nœud et créé de nouvelles distributions avec des densités plus faibles rendant la généralisation difficile.

Cette approche simple, part de ce dernier constat. Lors de la construction d'arbre de classification, une fonction est utilisée afin de calculer le gain « d'ordre » que va procurer la réponse à une question. Dans le cadre des arbres de classifications sémantique, les questions portes sur la présence ou l'absence de certain terme en contexte. Suivant la réponse obtenue pour chacun des documents de l'apprentissage, deux populations se découpent : l'une pour laquelle la réponse a été affirmative et l'autre pour laquelle la réponse a été négative. Deux types de fonctions de gain sont communément utilisés afin de calculer le gain qu'apporte une question : Le gain d'entropie et le gain en impureté de Gini. Dans la présente approche, nous utilisons le gain en impureté de Gini afin de calculer l'apport d'un terme pour décrire une classe donnée.

Le coefficient de Gini a été créé par le statisticien Corrado Gini (1912) afin de mesurer le degré d'inégalité de la distribution des revenus dans une population. L'application de ce critère d'impureté dans notre cas se traduit par la formule suivante :

$$G(Q) = 1 - \sum_{c \in C} P(c/Q)^2 \qquad (2)$$

où *P(c/Q)* est la probabilité de la classe *c* sachant la distribution *Q*.

De par cette formule, nous pouvons connaître l'impureté d'une distribution initiale. De même, il est possible de calculer le gain d'impureté porté par un terme en faisant la différence entre la moyenne pondérée du gain d'impureté de la population contenant le terme et de celle ne le contenant pas avec l'impureté initiale. Cela se traduit par la formule suivante :

$$GI(t) = G(Q) - \frac{|Q_t| G(Q_t) + |Q_{\neg t}| G(Q_{\neg t})}{|Q|} \qquad (3)$$

où $G(Q)$ est l'impureté de Gini pour la population initiale, $G(Q_t)$ est l'impureté de Gini pour la distribution de document contenant le terme t et $G(Q_{\neg t})$, celle qui ne le contient pas.

Chaque terme a un impact différent suivant la classe dans laquelle il est observé. Par exemple, le terme *insipide*, ne va pas avoir le même poids dans la classe *positive* que dans celle *négative*. Pour cela, il est nécessaire de d'attribuer un score à chaque terme par rapport à une classe. La combinaison suivante a été trouvée de façon empirique sur le corpus d'apprentissage :

$$S(c_i, t) = \frac{GI(t) \times Q_{\neg t} \times Q_{t,c_i}}{Q_t^2} \quad (4)$$

où $Q_{t,ci}$ est la distribution des documents contenant le terme t qui appartiennent à la classe c_i.

Cette approche nous a permis d'obtenir les meilleurs résultats sur le corpus d'apprentissage et c'est donc celle-ci qui a été retenu pour l'évaluation. Cependant, plusieurs filtrages ont mis en place afin de ne conserver que certain trait répondant à plusieurs critères de seuil, qui n'ont d'ailleurs pas été optimisés jusqu'à maintenant. Ce filtrage a été mis en place dans l'objectif de ne conserver que les termes ayant le plus d'important pour la prise d'opinion. Typiquement, seuls les termes ayant une fréquence intra-classe supérieure à 50% de leur masse totale, ont été conservés. De plus, un seuil supplémentaire a été appliqué afin d'éliminer les termes ayant un score $S(c_i,t)$ faible.

2.2.2 Prétraitement et Apprentissage

Les mêmes prétraitements ont été appliqués pour cette approche, que celle présentée dans la section précédente. Il n'y a pas « d'apprentissage » à proprement dit, juste un calcul de poids pour chacun des termes du corpus d'entrainement. En plus des unis et bi-grammes, quelques traits supplémentaires ont été ajoutés comme celui indiquant qu'un terme est présent dans les deux premières ou les deux dernières phrases du document, ainsi qu'un trait additionnel pour les termes ayant une fréquence supérieurs à 3 dans un même document.

Le décodage se fait très simplement en sommant sur chaque classe, le score des termes contenus dans les documents.

3 Evaluation

Les corpus d'évaluation sont au nombre de quatre et couvrent les mêmes domaines que les corpus d'apprentissage présentés précédemment. Le tableau suivant donne le nombre d'exemples par classe pour l'apprentissage et l'évaluation.

Table 1 - *Distribution des exemples d'apprentissage et d'évaluation selon les corpus*

	Corpus d'apprentissage				***Corpus d'évaluation***			
	Positif	**Neutre**	**Négatif**	***Total***	**Positif**	**Neutre**	**Négatif**	***Total***
Film/Livre	1150	615	309	*2074*	768	411	207	*1386*
Jeux	874	1166	497	*2537*	583	779	332	*1694*
Relecture	376	278	227	*881*	256	190	157	*603*
Débat	6899		10400	*17299*	4961		6572	*11533*

On remarquera que la distribution des fréquences par classe est assez bien respectée entre apprentissage et évaluation pour tous les corpus.

3.1 Scores

Lors de l'évaluation, chaque équipe avait la possibilité de présenter jusqu'à 3 systèmes différents. Pour notre part, nous n'en avons proposé que deux. Au lieu de calculer la précision et le rappel en ne tenant pas compte des classe, ceux-ci sont calculés classe par classe, puis ces scores sont moyennés pour donner une précision et un rappel moyen.

N'ayant donné qu'une seule suggestion de classe à chaque fois avec un indice de confiance de 1, nos scores pondérés et non-pondérés sont identiques. Pour cette raison, nous ne considéreront pas le premier cas dans notre analyse.

3.1.1 Exécution 1

Notre premier système utilisant des SVM avec une sélection des 80 traits *positifs/négatif* les plus saillants obtient des scores dans la moyenne des autres participants, comme le montre la colonne Δ de la Table 2.

Table 2 - Scores obtenus pour l'exécution 1

	Précision	**Rappel**	**F-Score**	**Δ**
Film/Livre	0.542	0.517	0.529	+0.029
Jeux	0.678	0.662	0.670	+0.009
Relecture	0.458	0.424	0.441	-0.030
Débat	0.692	0.616	0.652	+0.010

Le corpus de *Relecture* semble être celui présentant le plus de difficulté pour notre système, mais également pour les autres systèmes si l'on en juge par la F-Score moyen (47,1%). A l'opposé, le corpus de *Débat* obtient le meilleur score, ce qui n'est pas étonnant étant donné la bipolarité des notes à attribuer (*pour* ou *contre*).

3.1.2 Exécution 2

Notre second système basé sur une sommation des scores individuels d'appartenance à une classe de chaque terme qui compose les documents, a obtenu, de manière surprenante, de meilleurs résultats que notre première approche. Seul le score obtenu sur le corpus de revue de *Films* et *Livres* est très légèrement inférieur au précédent (Table 3).

Table 3 - Scores obtenus pour l'exécution 2

	Précision	**Rappel**	**F-Score**	**Δ**
Film/Livre	0.500	0.547	0.523	+0.022
Jeux	0.718	0.633	0.673	+0.013
Relecture	0.507	0.425	0.462	-0.008
Débat	0.715	0.691	0.703	+0.061

Le gain le plus significatif est à observer sur le corpus de *Débat* pour lequel notre système obtient un F-Score supérieur à 70% avec +5% par rapport à notre premier système et +6% par rapport à la moyenne des systèmes.

4 Analyse des résultats par système

Il est très difficile de comparer les systèmes en se basant uniquement sur les scores bruts, sachant que ceux-ci sont en fait une moyenne de scores par classe. De plus, il est intéressant de décomposer les scores selon les classes afin d'appréhender la difficulté de chaque tâche selon le corpus. Nous nous efforcerons dans cette section d'effectuer une analyse détaillée de chaque système en présentant des exemples issus directement des corpus de test et d'apprentissage.

4.1 Analyse du système à base de SVM : *Exécution 1*

L'aspect le plus important et déterminant pour cette approche est la sélection des traits à utiliser pour entrainer les modèles SVM. La restriction des 80 traits est une contrainte supplémentaire qui impose une grande rigueur pour cette phase de l'apprentissage. Il ne saurait être question de sélectionner ces termes au hasard ou même, de prendre les termes les plus fréquents car beaucoup d'entre eux n'ont qu'une fonction de support, ou pire encore, sont présents dans une portion descriptive du document n'ayant pas pour but de donner une opinion sur le sujet (comme par exemple les synopsies de films). Cependant, il faudrait pourvoir détecter automatiquement ce qu'il retourne de l'opinion et ce qui est purement narration, ce qui n'est pas une chose aisée à effectuer de manière automatique. De plus, certaine fois, le ton employé pour la narration peut être porteur d'information quand à l'opinion de la personne qui a rédigée le commentaire, surtout dans le cas du corpus de *Relecture* où l'auteur est obligé de faire une synthèse de lui-même.

La Table 4 contient une série de mots utilisés pour l'apprentissage des modèles SVM sur le corpus *Film/Livre*.

Table 4 - *Top des termes saillants pour le corpus de critique de Film/Livre*

Saillance négative		**Saillance positive**	
malheureusement	0.483	écriture	0.033
._malheureusement	0.368	chef-d'	0.025
spectateur	0.366	mots	0.024
acteurs	0.321	chef-d'_oeuvre	0.024
ennui	0.307	magnifique	0.021
se_contente	0.286	roman	0.020
contente	0.269	artiste	0.020
comédie	0.257	sens	0.019
bons	0.234	oeuvre	0.019
intérêt	0.230	plume	0.018
cette_production	0.228	porte	0.017
hélas	0.215	personnel	0.017
décevant	0.211	maître	0.017
ridicule	0.201	littéraire	0.017
réalisation	0.199	évidemment	0.016
._hélas	0.196	rêves	0.016
Insipide	0.182	auteur	0.016
…	…	…	…

Plusieurs observations peuvent être émisses à partir de cet exemple. Dans un premier temps, on remarquera un problème dû à l'utilisation des unis et bi-grammes dans le même modèle. Ainsi, le terme *malheureusement* semble être le mot le plus saillant pour représenté un avis *négatif*. Mais, ce même mot est également listé en seconde position lorsqu'il est précédé d'un point (ce qui établi ça présence en début de phrase). Cela remet quelque peu en cause condition d'indépendance des variables.

Autre remarque, cette liste paraît contenir des termes semblant tout à fait dénués de tonalité comme par exemple les mots *spectateur* et *acteurs*. La raison en est que leur distribution au sein de la classe *négative* est incohérente avec celle de l'ensemble du corpus. Apparaissant dans 100 documents *négatifs* sur les 309 que comporte cette classe (environ 32% des critiques *négatives*), le mot *spectateur* à une distribution « diverge » par rapport à l'observation faite sur l'ensemble du corpus car n'apparaissant que 324 dans l'ensemble du corpus sur 2974 documents (donc, dans moins de 15% du corpus). A noter qu'aucune lemmatisation n'a été effectué sur les données, donc pluriels et singuliers sont traités comme des mots différents. Bien que ce mot ne soit pas teinté d'opinion, il est néanmoins une sorte de marqueur indiquant une prise de position souvent *négative* comme le montre les exemples suivant :

...s'abîme dans des considérations besogneuses et installe le ***spectateur*** *dans une torpeur sourde et désagréable.*

Réussite totale sur ce point, mais la formule choisie, véritable carcan, emprisonne le ***spectateur****.*

...la complaisance de cette chronique désenchantée épuisent les résistances du ***spectateur****.*

Le critique, prend souvent le *spectateur* à partie en décrivant les effets *négatifs* que tel ou tel film à sur lui ou elle. Ces termes ne sont toutefois pas porteurs d'opinion à eux seuls et bien qu'il augment la probabilité d'être face à une opinion *négative*, ils doivent cependant être secondés pas d'autres indices afin de réellement discriminer une classe par rapport à une autre. C'est d'ailleurs ce que nous pouvons observer à travers le poids qu'attribut l'apprentissage SVM à ces termes. Par exemple, le terme *spectateur* à un poids de 0.279, ce qui est positive, mais bien en dessous du *prior* qui est de -0,833. Donc, un document ne contenant que ce terme là, ne sera pas considéré comme *négatif* par le modèle.

Sur les 1386 documents que contient le corpus de test *Film/Livre*, 46 documents ne contenaient aucun des 80 termes sélectionnés lors de la construction des modèles. Le nombre de traits moyen non-nul pour ce même corpus est de 4,2 termes. Ce qui dénoté qu'en moyenne 4 termes sur les 80 sélectionnés ont été utilisés lors du décodage pour les documents de test. Ce chiffre passe à 5,3 pour le corpus de *Jeux* (et 28 documents vides sur 1694), 5,6 pour les *Relectures* (et 40 documents vides sur 603) et 0,97 pour le corpus de Débats (et 6068 documents vides sur 11533). On remarquera que plus de la moitié des documents du test sur les textes de *Débat*, ne contiennent aucun des termes sélectionnés. Cela veux implique donc que la moitié des jugements sont données grâce au *prior*, qui est de 1,0 pour le modèle *négative*, donc une opinion *négative* par défaut. Ce qui est bien en accord avec la distribution des éléments *positif/négatif* dans le corpus où les exemples *négatifs* sont plus nombreux que ceux *positifs*.

Enfin, il convient d'analyser la capacité du système à détecter plus une classe par rapport à une autre. La **Fig. 1** présente les F-Scores pour chaque corpus et chaque classe. Le corpus *Débat* étant bipolaire, il ne comprend pas de score pour la classe *neutre*.

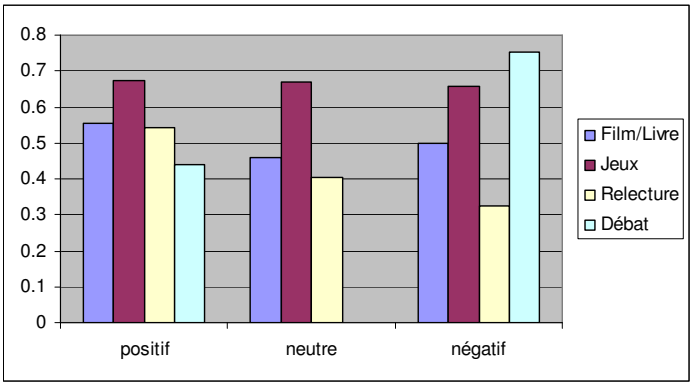


Fig. 1 - *F-Scores par classe et par corpus pour l'exécution 1*

Alors que le corpus de *Jeux* concède un score quasiment identique sur les trois classes, ce n'est pas le cas pour les autres corpus. Ainsi, on observe les plus grandes variations pour les corpus *Relecture* et *Débat* avec une préférence inversée quand à la classe la mieux identifiée. Cependant, il est difficile d'effectuer une analyse fine en utilisant le F-Score qui est une combinaison de précision et rappel. La Table 5 donne donc les scores de précision et de rappel pour chaque classe et pour chaque corpus de l'évaluation.

Table 5 - *Précision et Rappel par classe et par corpus pour l'éxécution 1*

	Précision			**Rappel**		
	positif	**neutre**	**négatif**	**positif**	**neutre**	**négatif**
Film/Livre	0.733	0.356	0.536	0.444	0.642	0.464
Jeux	0.745	0.630	0.658	0.617	0.710	0.660
Relecture	0.621	0.327	0.427	0.480	0.532	0.261
Débat	0.744	/	0.639	0.313	/	0.919

On observera en premier lieu que la précision de la classe *positive* est toujours la plus élevée que celle de la classe *négative*. Cela semble donc indiquer que ce système à tendance à n'affecter la classe *positive*

que lorsque la certitude est élevée. Par contre, la classe *neutre* semble être la classe par défaut pour tous les corpus à trois états et la classe *négative* pour le corpus *Débat*.

Cela semble traduire en fait l'incapacité du système à généraliser afin d'avoir assez de matière pour effectuer la classification. Le nombre réduit de traits utilisés pénalise grandement les classes d'apprentissage des SVM, car le jugement doit généralement se faire sur trop peu de termes.

4.2 Analyse du système à sommation de score : *Exécution 2*

Cette méthode, très empirique, à l'avantage de fonder sa décision sur un nombre beaucoup plus important de termes et donc d'indices. Par exemple, *1586* indices ont été extraits du corpus d'apprentissage *Film/Livre*. Ce nombre d'indice est très variable selon les corpus, il est de *5531* pour le corpus de *Jeux*, *1942* pour le corpus de *Relectures* et seulement *531* pour le corpus de *Débats*. Cette variation est due à plusieurs facteurs dont l'un des principaux est la mise en dur des seuils de filtrage dans le système.

La Table 6 présente des exemples de trait avec leur score respectif sur le corpus *Film/Livre*. Les traits précédés de *#LAST* et *#FIRST* correspondent à une présence de ces termes dans les deux dernières ou premières phrases du document. *#HF* indique, quant-à lui, une fréquence d'apparition supérieure ou égale à trois.

Table 6 - *Exemples de traits les plus saillants pour le corpus Film/Livre*

Négatif		**Neutre**		**Positif**	
Trait	**Score**	**Trait**	**Score**	**Trait**	**Score**
#LAST:ennuyeux	0.8150	ses_élèves	0.5193	oubli	0.1860
mollement	0.7697	agréable_à	0.5193	autre_côté	0.1764
remake#HF	0.7140	narre	0.4852	poignante	0.1692
lourdingue	0.7140	petite-fille	0.4852	#FIRST:chef-d'	0.1651
fade_.	0.7140	assez_bien	0.4852	plus_beau	0.1651
gâché_par	0.7140	est_dommage	0.4852	avancer	0.1588
consternant	0.7140	jeune_public	0.4852	#FIRST:magnifique	0.1579
décevant_.	0.6810	prof_de	0.4852	#LAST:photographie	0.1579
molle	0.6439	#LAST:jolis	0.4436	après-guerre	0.1579
vite_dans	0.6439	sérieuses	0.4436	#LAST:vérité	0.1553
scènes_sont	0.6439	face_.	0.4436	dépouillé	0.1533
insignifiante	0.6439	#LAST:en_attendant	0.4436	sans_oublier	0.1519
scène_sans	0.6439	#LAST:attendant	0.4436	plus_vite	0.1519
médiocrité_.	0.6439	#LAST:en_face	0.4436	pleines	0.1482
navet	0.6215	nos_héros	0.4436	narratrice	0.1482
ennuient	0.5536	#LAST:au_charme	0.4436	névroses	0.1482
…	…	…	…	…	…

Comme cela était le cas dans l'approche par sélection de traits pour les modèles SVM, tous les traits présentés ici ne semble pas tous pertinents par rapport à la classe. De plus, la description de la classe *neutre* est très abstraite dans le sens où il est difficile de définir la notion de neutralité pour un trait.

Le comportement du deuxième système est assez différent du premier pour certain corpus si l'on considère les *F-Scores* pour chaque classe. Ainsi, comme nous le montre la **Fig. 2**, la classe *neutre* obtient de moins bons scores, saufs pour le corpus de *Relectures*.

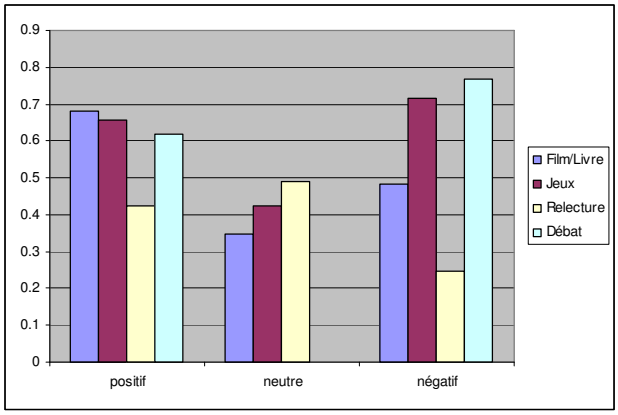


Fig. 2 - *F-Scores par classe et par corpus pour l'exécution 2*

Cela traduit directement la difficulté de la tâche de modélisation de cette classe. Par contre, la classe *positive* obtient de meilleurs scores (sauf pour le corpus de *Relectures*). Comme cela a été fait pour le système précédent, il faut analyser la précision et le rappel au niveau de chaque classe afin d'avoir une image claire de l'impact du système. La Table 7 nous montre une bonne précision du système en classe *positive*, sauf pour le corpus de *Jeux* pour lequel les classes *neutre* et *négative* dégagent une forte progression au détriment de la classe *positive*.

Table 7 - *Précision et Rappel par classe et par corpus pour l'éxécution 2*

	Précision			**Rappel**		
	positif	**neutre**	**négatif**	**positif**	**neutre**	**négatif**
Film/Livre	**0.708**	0.419	0.374	0.659	0.297	**0.686**
Jeux	0.490	0.826	**0.838**	**0.990**	0.286	0.623
Relecture	**0.696**	0.351	0.473	0.305	**0.805**	0.166
Débat	**0.721**	/	0.708	0.541	/	**0.842**

En regardant le nombre de traits utilisés afin de décrire chaque classe, on s'aperçoit qu'il y a une corrélation forte avec la précision sur cette classe. Les classes ayant le plus grand nombre de traits sont celles qui obtiennent les moins bons scores en précision. Par exemple, 54,3% des traits sélectionnés pour le corpus de *Jeux* sont dédiés à la classe *positive*. Un déséquilibre trop important dans le nombre de traits pour une classe plébiscite grandement celle-ci au détriment des autres, d'où une précision plus faible (peut être perçu comme la classe pas défaut).

Bien que ce système donne de meilleurs résultats que le précédent, il reste cependant très imparfait de par ses aspects empiriques. Il demande notamment un ajustement des paramètres de filtrage au niveau de chaque corpus. Pour cela, une technique de *N-Folds* devrait permettre une plus grande souplesse d'adaptation.

5 Conclusion

Nous avons montré à travers cette évaluation que des « systèmes simples », peuvent obtenir des résultats honorables, dans la moyenne des autres participants. Notre système à base de SVM et de sélection de traits basée sur la saillance des termes, permet d'atteindre une bonne précision sur la classe *positive* avec des scores supérieurs à 60%. Cependant, cette approche souffre de la limitation du nombre de traits utilisables en entrée, qui est de 80 dans notre cas. Les perspectives pour ce système seraient d'utiliser un plus grand nombre de trait avec leur score de saillance comme pondération.

Notre seconde approche semble quant-à elle prometteuse car donnant de meilleurs résultats que la précédente. De plus, une marge de progression important peut sens doute être réalisée en affinant le

filtrage des termes selon le corpus considéré. Cela permettrait notamment de limiter les pertes en précision due à une surreprésentation d'une classe dans la liste des traits retenus.

Dans tous les cas, il serait probablement bénéfique de pouvoir travailler au niveau de la phrase, plutôt que de considérer le document comme un tout. Ceci nous donnerait en sus la possibilité de rejeter les phrases de narration, comme celles contenus dans les synopsies de film, afin de ne pas comptabiliser des indices *positifs* ou *négatifs* n'indiquant pas de prise d'opinion de l'auteur.

Références

CRESTAN E. (2004). *Contextual semantics for WSD*, dans les Actes de Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, p. 101-104, Barcelone, Espagne.

GINI C. (1912). *"Variabilità e mutabilità"* Reimprimé dans Memorie di metodologica statistica (Ed. Pizetti E, Salvemini, T). Rome: Libreria Eredi Virgilio Veschi (1955).

GREFENSTETTE G., QU Y., SHANAHAN J. G. & EVANS D. A. (2001). *Coupling niche browsers and affect analysis for an opinion mining application.* Dans les Actes de RIAO-2004.

JOACHIMS T. (1997). *Text categorization with support vector machines: Learning with many relevant features*. Technical Report 23, Universitat Dortmund, LS VIII.

KULLBACK S. & LEIBLER R. A. (1951). *On information and sufficiency*, Annals of Mathematical Statistics 22: 79-86.

KUSHAL D., LAWRENCE S. & PENNOCK D. (2003). *Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews*. Dans les Actes de WWW2003, Budapest, Hungary, 20-24 Mai 2003, p 519–528.

RÜPING S. (2000). mySVM-Manual, University of Dortmund, Lehrstuhl Informatik 8, http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/.

SCHEIN A. I., POPESCUL A. & UNGAR L. H. (2002). *Methods and Metrics for Cold-Start Recommendations*. Dans les Actes de the XXV Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland.

SCHÜTZE H., HULL D. & PEDERSEN J. (1995). *A comparison of Classiers and document representations for the routing problem*. In International ACM SIGIR Conference on Research and Development in Information Retrieval.

WILSON T., WIEBE J. & HOFFMANN P. (2005). *Recognizing contextual polarity in phrase-level sentiment analysis.* Dans les Actes de Human Languages Technology Conference/EMNLP 2005.

YANG Y. & PEDERSEN J. (1997). *A comparative study on feature selection in text categorization*. In International Conference on Machine Learning (ICML).

Défi DEFT07 : Comparaison d'approches pour la classification de textes d'opinion

Michel Plantié[1], Gérard Dray[1], Mathieu Roche[2]

[1] Laboratoire LGI2P, Laboratoire de Génie informatique et d'ingénierie de la production,
Ecole des Mines d'Alès, Site EERIE –parc scientifique Georges Besse, 30035 – Nîmes,
(michel.plantie, gerard.dray)@ema.fr

[2] Laboratoire LIRMM, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier,
UMR 5506, 161 rue Ada, 34392 Montpellier Cedex 5 - France,
mathieu.roche@lirmm.fr

Résumé : Nous exposons dans cet article, les méthodes utilisées pour répondre au défi DEFT 2007. Après une présentation succincte de la méthode générale incluant les différents types de classifications utilisés, les résultats obtenus sont détaillés et analysés. Plusieurs tentatives d'améliorations des résultats initiaux sont enfin proposés.

Mots-clés : Classification, fouille de texte, Machine à Vecteurs Support, SVM, Naïve Bayes, Loi Multinomiale, sélection d'attributs, validation croisée, Apprentissage machine.

1 Introduction

Le défi consiste comme il est indiqué sur le site : http://deft07.limsi.fr/ à évaluer en différentes opinions des textes d'opinion différents corpus en français de style et de domaine différents.

Les corpus et leurs catégories d'évaluation sont :

- Corpus 1 : critiques films, livres, spectacles et bandes dessinées,
  - Trois catégories : bon, moyen, mauvais
- Corpus 2 : critiques de tests de jeux vidéo,
  - Trois catégories : bon, moyen, mauvais
- Corpus 3 : Commentaires de révision d'articles de conférences scientifiques,
  - Trois catégories : acceptation, acceptation sous condition, rejet
- Interventions des parlementaires et du gouvernement dans les débats sur les projets de lois votés à l'Assemblée nationale
  - Deux catégories : pour, contre

Afin de pouvoir trouver les méthodes de traitements toutes les équipes avaient accès à quatre corpus d'apprentissage. Les corpus de test ont ensuite été fournis par les organisateurs du défi. Ainsi, le résultat de chaque équipe sur les données test a été évalué.

Un tel défi permet d'estimer globalement la qualité des méthodes de classifications à partir de textes spécifiques (ici, des textes d'opinions). Précisons que notre approche dans le cadre de DEFT'07 n'utilise aucun traitement spécifique propre aux corpus. En effet, le but du challenge est d'avoir des approches génériques de classifications adaptées à des textes d'opinion. Notre approche générale a donc été intégralement appliquée sur chacun des corpus. Ainsi, la spécificité, notamment linguistique, de chacun des textes d'opinion (tournures de phrases, richesse du vocabulaire, etc) n'a pas réellement été prise en compte dans notre approche.

Cet article qui se veut assez technique dans la présentation des résultats développe succinctement les méthodes appliquées et les résultats obtenus avec ces dernières. Le détail des approches utilisées n'est pas donné dans cet article qui a pour but d'analyser la performance et également les contre performances des différents traitements appliqués. Bien que nos résultats soient globalement satisfaisants (situés dans la moyenne des résultats des participants), les résultats négatifs que nous avons obtenus ont été volontairement présentés dans cet article. En effet, nous estimons que ceux-ci peuvent être particulièrement intéressants pour la communauté « fouille de texte ».

Après une pésentation de notre méthode générale détaillée en section 2, la section suivante décrit les résultats obtenus. Enfin, la section 4 propose des méthodes additionnelles qui ont également été testées dans le cadre du

défi mais qui n'ont malheureusement pas été toujours satisfaisantes. Enfin, la section 5 développe quelques perspectives à notre travail.

2 Méthode générale

Dans ce défi nous avons considéré que le problème posé relevait de la problématique de la classification. Chaque opinion possible représente une catégorie et la tâche se traduisait donc en une procédure pour attribuer des candidats à une catégorie prédéfinie.

La méthode de traitement générique que nous avons utilisée comprend cinq étapes détaillées ci-après.

Étape 1 : prétraitement linguistique : recherche des unités linguistiques du corpus.

Étape 2 : prétraitement linguistique et représentation mathématique des textes du corpus

Étape 3 : sélection des unités linguistiques caractéristiques du corpus.

Étape 4 : choix de la méthode de classification

Étape 5 : Évaluation des performances de la classification par validation croisée

Cette méthode de traitement a été utilisée telle quelle et également nous avons ajoutés dans certains cas des traitements supplémentaires afin de tenter d'améliorer les résultats.

2.1 Recherche des unités linguistiques de chaque corpus

Ce prétraitement consiste à extraire du corpus toutes les unités linguistiques utilisées pour la représentation des textes de ce corpus.

Dans notre méthode, une unité linguistique est un mot lemmatisé ou lemme.

Cependant certains types grammaticaux sont éliminés : les articles indéfinis et la ponctuation faible.

Nous extrayons donc tous les mots lemmatisés pour chaque corpus. Cela donne pour chaque corpus :

Corpus	Nombre d'unités linguistiques (lemmes)
Corpus 1	36214
Corpus 2	39364
Corpus 3	10157
Corpus 4	35841

Cette opération est effectuée avec l'outil d'analyse syntaxique « Synapse » (Synapse, 2001).

Cette liste de lemme pour chaque corpus constituera donc ce que nous nommerons un « index ».

Chaque texte sera représenté par un vecteur de « compte ». L'espace vectoriel de représentation est constitué par un nombre de dimension égal au nombre de lemmes du corpus. Chaque dimension représente un lemme. Ainsi chaque coordonnée d'un vecteur représentera le nombre d'occurrence du lemme associé à cette dimension dans le texte.

2.2 prétraitement linguistique et représentation mathématique des textes du corpus

- Lemmatisation : Chaque texte subit le même prétraitement linguistique que précédemment c'est-à-dire une lemmatisation. Ainsi chaque texte est transformé en une suite de lemme.

- Filtrage grammatical : Dans une deuxième étape certains types grammaticaux sont éliminés. Dans un processus où il s'agit de différentier des appréciations positives et négatives nous avons choisi de conserver tous les lemmes exceptés : les articles indéfinis et la ponctuation faible. Nous pensons que ces deux éléments n'ont pas d'incidence sur la tonalité du texte. Et surtout tous les autres types grammaticaux sont succeptibles d'exprimer des nuances d'opinions ou des contributions à des opinions. Nous avons donc conservé les lemmes associés à tous ces types grammaticaux.

- Vectorisation : Enfin la dernière étape consiste à transformer en vecteur d'occurrence chaque texte. Les dimensions de l'espace vectoriel étant l'ensemble des lemmes du corpus. Chaque coordonnée d'une dimension représente donc le nombre d'apparition dans le texte considéré du lemme associé à cette dimension.

2.3 Sélection des unités linguistiques caractéristiques du corpus

L'ensemble des textes d'un corpus et donc les vecteurs associés constituent dans notre approche l'ensemble d'apprentissage qui permettra de calculer un classifieur associé. L'espace vectoriel défini par l'ensemble des lemmes du corpus d'apprentissage et dans lequel sont définis ces vecteurs comporte un nombre important de dimensions. Par suite, les vecteurs de chaque texte de l'apprentissage peuvent avoir de nombreuses composantes toujours nulles selon certaines de ces dimensions. On peut donc considérer que ces dimensions n'ont aucune incidence dans le processus de classification et peuvent même ajouter du bruit dans le calcul du classifieur entraînant des performances moindres de la classification.

Pour pallier cet inconvénient, nous avons choisi d'effectuer une réduction de l'index afin d'améliorer les performances des classifieurs. Nous utilisons la méthode très connue présentée par Cover qui mesure l'information mutuelle associée à chaque dimension de l'espace vectoriel (Cover & Thomas, 1991).

Cette méthode expliquée en détail dans (Plantié, 2006) permet de mesurer l'interdépendance entre les mots et les catégories de classement des textes.

Dans le tableau suivant nous voyons les diminutions de la dimension de l'espace vectoriel associée à chaque corpus.

Corpus	**Nombre initial d'unités linguistiques**	**Nombre d'unités linguistiques Après réduction**
Corpus 1	36214	704
Corpus 2	39364	2363
Corpus 3	10157	156
Corpus 4	35841	3193

2.4 Construction des vecteurs réduits de l'ensemble des textes de chaque corpus

Une fois les « index » de chaque corpus obtenus, nous considérons chaque mot clé sélectionné dans cet index comme les nouvelles dimensions des nouveaux espaces vectoriels de représentation des textes de chaque corpus. Les expaces vectoriels en question comporteront donc un nombre de dimensions largement réduit. Ainsi pour chaque corpus nous calculerons les vecteurs d'occurrence de chaque texte associé à l'index du corpus considéré. Nous nommerons les vecteurs ainsi calculés : les vecteurs « réduits ».

L'utilisation de cette réduction d'index permet d'améliorer grandement les performance des classifieurs.

2.5 Choix de la méthode de classification

Une fois réduit l'espace vectoriel nous procédons au calcul du modèle de classification. Ce modèle sera ensuite utilisé pour l'évaluation des textes du jeu de test.

Nous avons utilisé plusieurs méthodes de classification. Elles sont fondées sur quatre méthodes principales.

Nous avons également tenté d'autres procédures de classification dont les performances se sont révélées moins intéressantes.

Le choix de la procédure de classification s'est fait sur chaque ensemble d'apprentissage ou corpus. La sélection fut très simple, nous avons conservé la méthode de classification la plus performante pour un corpus donné. Les mesures de performances sont décrites ci après.

Nous décrivons brièvement ci-après les cinq méthodes de classification. Notons que la plupart de ces méthodes est décrite de manière précise dans (Plantié, 2006).

En voici la liste :

- La classification probabiliste utilisant la combinaison de la loi de Bayes et de la loi multinomiale,
- La classification par les machines à vecteurs support S.V.M.
- La classification par la méthode des réseaux RBF (Radial Basis Function)
- La classification par arbre de décision de type C4.5
- La classification par la méthode probabiliste fondée sur la loi de Dirichlet lissée

2.5.1 Classifieur de Bayes Multinomial

Cette technique (Wang, Hodges, & Tang, 2003) est classique pour la catégorisation de textes nous l'avons décrite dans (Plantié, 2006). Elle combine l'utilisation de la loi de Bayes bien connue en probabilités et la loi

multinomiale. Nous avons simplement précisé le calcul de la loi à priori en utilisant l'estimateur de Laplace pour éviter les biais dus à l'absence de certains mots dans un texte.

2.5.2 Classifieur par la méthode des Machines à Vecteurs Support (S.V.M.)

Cette technique (Joachims, 1998) a été décrite dans (Plantié, 2006). Elle consiste à délimiter par la frontière la plus large possible les différentes catégories des échantillons (ici les textes) de l'espace vectoriel du corpus d'apprentissage. Les vecteurs supports constituent les éléments délimitant cette frontière.

Plusieurs méthodes de calcul des vecteurs support peuvent être utilisées comme indiqué dans (Platt, 1998) :

- une méthode linéaire
- une méthode polynomiale
- une méthode fondée sur la loi gaussienne normale
- une méthode fondée sur la fonction sigmoïde

Nous avons effectués des essais avec plusieurs de ces méthodes.

2.5.3 Classifieur par la méthode des réseaux RBF (Radial Basis Function)

Cette technique implémente un réseau de neurones à fonctions radiales de base. Elle utilise un algorithme de « clustering » de type « k-means » (MacQueen., 1967) et utilise au dessus de cet algorithme une régression linéaire. Les gaussiennes multivariables symétriques sont adaptées aux données de chaque « cluster ». Toutes les données numériques sont normalisées (moyenne à zéro, variance unitaire). Cette technique est présentée dans (Parks & Sandberg, 1991).

2.5.4 Classifieur par la méthode des arbre de décision de type C4.5

Cette technique utilise l'approche bien connue de (Quinlan, 1993) et qui est également présentée dans (Plantié, 2006). Elle permet d'élaborer un arbre de décision sur l'ensemble des mots clés constituant l'espace vectoriel de représentation des textes. nous l'avons décrite dans (Plantié, 2006).

2.5.5 Classifieur par la méthode probabiliste de Bayes combiné à la loi de Dirichlet

Cette technique utilise le même principe que la méthode probabiliste de Bayes/multinomiale décrite précédemment mais remplace la loi de Bayes et la loi Multinomiale par une loi de Dirichlet lissée comme précisé dans (Nallapati Ramesh, 2006).

2.6 Évaluation des performances de la classification par validation croisée

La validation croisée est une technique d'évaluation permettant de valider une méthode de classification en particulier. Cette approche ne construit pas de modèle utilisable mais sert à estimer l'erreur réelle d'un modèle selon l'algorithme suivant (figure 1) :

Validation croisée (S;x) *: // S est un ensemble, x est un entier*
Découper S en x parties égales S1, … , Sx
Pour i de 1 à x
Construire un modèle M avec l'ensemble S - Si
Evaluer une mesure d'erreur ei de M avec Si
Fin Pour

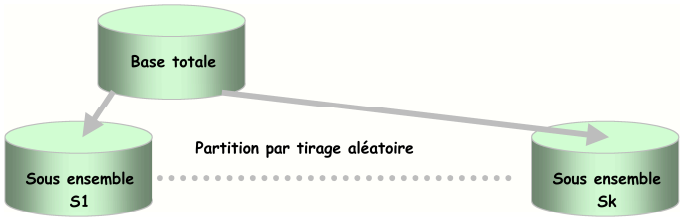


Figure 1 : Processus de validation croisée

Si la taille des *Si* est de un individu, on parle alors de validation par « leave one out ».

En général le nombre *x* de parties est fixé à 10.

Dans notre approche nous avons utilisée la méthode de validation croisée sur l'ensemble des vecteurs « réduits » d'un corpus. L'objectif que nous nous sommes fixés dans le cadre du défi est d'évaluer nos résultats à partir du seul corpus d'apprentissage disponible. Ceci nous a aidé à adapter les paramètres les plus pertinents.

2.7 Mesure de performances de la classification

Pour évaluer la performance d'un procédé de classification nous utilisons la mesure préconisée dans le cadre du défi DEFT07 c'est à dire le « fscore ». Il s'agit de la moyenne harmonique de la précision et du rappel. Ces deux mesures sont bien connues, et une explication complète de ces mesures est écrite dans (Plantié, 2006).

3 Résultats obtenus avec la méthode générale

Nous allons présenter ici les résultats obtenus tout d'abord en validation croisée sur le corpus d'apprentissage, puis les résultats sur les corpus de tests fournis dans le cadre du défi DEFT.

Nous allons présenter ces résultats par corpus.

Dans les tableaux présentés ci-dessous, il existe un différence notable entre ceux obtenus par la méthode de validation croisée et ceux obtenus sur les corpus de test. Cette différence est expliquée à la fin de ce chapitre.

3.1 Corpus 1

En utilisant la méthode générale présentée précédemment nous avons sélectionné plusieurs classifieurs performants.

Le corpus d'apprentissage comporte 2074 textes dont :

309 textes classés : 0 (*mauvais*)

615 textes classés : 1 (*moyen*)

1150 textes classés : 2 (*bien*)

Ce corpus est déséquilibré, la dernière catégorie comporte deux fois plus d'individus que les autres. Le déséquilibre entre les tailles des catégories pose souvent des difficultés pour obtenir de bons scores de classement. En effet si la performance sur la classe la plus volumineuse est faible en pourcentage de fscore le nombre d'échantillons mal classés devient important et les performances sur les autres classes deviennent bien plus faibles.

Dans le cas d'un corpus déséquilibré la performance de l'ensemble dépend en grande partie de la performance obtenue sur la catégorie comportant le plus grand nombre d'échantillons.

Voici le tableau des résultats obtenus.

		Validation Croisée			**Jeu de test**
Type de classifieur		Précision	Rappel	Fscore	Fscore
RBF-Network	Classe 0	0.927	0.737	0.821	**0.4715**
	Classe 1	0.713	0.704	0.708	
	Classe 2	0.835	0.887	0.86	
Naive Bayes Multinomial	Classe 0	0.735	0.776	0.755	0.5902
	Classe 1	0.635	0.56	0.595	
	Classe 2	0.806	0.845	0.825	
SVM	Classe 0	0.729	0.708	0.718	0.6102
	Classe 1	0.602	0.575	0.588	
	Classe 2	0.796	0.821	0.808	
Dirichlet	Classe 0	0.4224	0.9545	0.5857	
	Classe 1	0.7034	0.4365	0.5387	
	Classe 2	0.8581	0.7431	0.7965	

Comme indiqué précédemment nous constatons une chute importante des résultats sur le corpus de test. Nous pouvons tirer quelques enseignements des résultats précédents :

- le classifieur RBF-Network est plus performant sur le jeu d'apprentissage mais son résultat chute très fortement sur le jeu de test, plus fortement que les autres classifieurs. Il est donc plus sensible à l'apparition de nouvelles données.
- Le classifieur de Bayes Multinomial est très sensible au déséquilibre de population des individus. Il est cependant plus robuste sur les données de test.
- Le classifieur SVM donne les meilleurs résultats.
- Le classifieur Dirichlet n'est pas meilleur que les deux précédents.

Remarque : Les résultats sur les jeux de test pour les classifieur de Bayes et SVM ne sont pas officiels, ils ont été effectués après l'échéance. Compte tenu du nombre très limité de soumissions possibles nous avons préféré soumettre d'autres résultats fondés sur des méthodes combinés.

Les résultats sur ce corpus sont moyens (62% de fscore au maximum). Nous avons tenté d'améliorer ce score par une méthode fondée sur les synonymes (voir ci-après).

Le résultat du classifieur RBF-Network a été publié dans le jeu 1 de nos soumissions.

3.2 Corpus 2

Le corpus d'apprentissage comporte 2537 textes dont :
497 textes classés : 0 (*mauvais*)
1166 textes classés : 1 (*moyen*)
874 textes classés : 2 (*bien*)

Ce corpus est légèrement déséquilibré, la catégorie 1(moyen) comporte deux fois plus d'individus que la première. Voici le tableau des résultats obtenus.

		Validation Croisée			**Jeu de test**
Type de classifieur		Précision	Rappel	Fscore	Fscore
SVM	Classe 0	0.825	0.774	0.799	**0.7829**
	Classe 1	0.799	0.842	0.82	
	Classe 2	0.866	0.834	0.849	
RBF-Network	Classe 0	0.912	0.789	0.846	**0.5475**
	Classe 1	0.782	0.927	0.849	
	Classe 2	0.906	0.751	0.821	
Naive Bayes Multinomial	Classe 0	0.792	0.819	0.805	0.7416
	Classe 1	0.812	0.815	0.814	
	Classe 2	0.862	0.841	0.851	
Dirichlet	Classe 0	0.5490	0.9671	0.7004	
	Classe 1	0.9319	0.4957	0.6472	
	Classe 2	0.7653	0.9185	0.8349	

La chute des résultats sur le corpus de test est moins importante pour la méthode SVM.

Nous pouvons tirer quelques enseignements des résultats précédents :

- le classifieur RBF-Network est équivalent à SVM sur le jeu d'apprentissage mais son résultat chute très fortement sur le jeu de test.
- Le classifieur de Bayes Multinomial donne de bons résultats malgré le déséquilibre de population des individus.
- Le classifieur SVM donne les meilleurs résultats.
- Le classifieur Dirichlet n'est pas meilleur que les deux précédents.

Remarque : Les résultats sur les jeux de test pour le classifieur de Bayes ne sont pas officiels, ils ont été effectués après l'échéance, nous avons préféré soumettre d'autre résultats fondés sur des méthodes combinés.

Le résultat du classifieur SVM a été publié dans le jeu 1 de nos soumissions.

Le résultat du classifieur RBF-Network a été publié dans le jeu 3 de nos soumissions.

Notons par ailleurs, que les résultats que nous avons obtenus pour la soumission 1 sur ce corpus sont de très bonne qualité (0.78) comparativement au fscore moyen du défi sur ce corpus (0.6604 +/- 0.086). Une analyse plus approfondie serait nécessaire pour expliquer un tel résultat très positif sur le corpus des critiques de tests de jeux vidéo qui est d'une taille conséquente.

3.3 Corpus 3

Le corpus d'apprentissage comporte 881 textes dont :
227 textes classés : 0 (*rejet*)
278 textes classés : 1 (*acceptation sous conditions*)
376 textes classés : 2 (*acceptation*)
Ce corpus est assez équilibré, la catégorie 2(*acceptation*) comporte 50% d'individus en plus que la 1.
Voici le tableau des résultats obtenus.

		Validation Croisée			**Jeu de test**
Type de classifieur		Précision	Rappel	Fscore	Fscore
SVM	Classe 0	0.733	0.604	0.662	**0.4782**
	Classe 1	0.645	0.57	0.605	
	Classe 2	0.673	0.803	0.732	
RBF-Network	Classe 0	0.503	0.758	0.605	
	Classe 1	0.91	0.44	0.594	
	Classe 2	0.645	0.693	0.668	
Naive Bayes Multinomial	Classe 0	0.6452	0.819	0.805	0.4914
	Classe 1	0.812	0.815	0.814	
	Classe 2	0.862	0.841	0.851	
Dirichlet	Classe 0	0.5985	0.696	0.6436	
	Classe 1	0.6234	0.5035	0.5571	
	Classe 2	0.6768	0.7093	0.6927	

La chute des résultats sur le corpus de test est conséquente pour tous les classifieurs.
Nous pouvons tirer quelques enseignements des résultats précédents :

- le classifieur RBF-Network est presque équivalent à SVM sur le jeu d'apprentissage mais son résultat chute très fortement sur le jeu de test.
- Le classifieur de Bayes Multinomial donne les meilleurs résultats.
- Le classifieur Dirichlet n'est pas meilleur que les deux précédents.

Remarque : Les résultats sur les jeux de test pour le classifieur de Bayes ne sont pas officiels, ils ont été effectués après l'échéance, nous avons préféré soumettre d'autre résultats fondés sur des méthodes combinés.
Le résultat du classifieur SVM a été publié dans le jeu 1 de nos soumissions.

3.4 Corpus 4

Le corpus d'apprentissage comporte 17299 textes dont : 10400 textes classés : 0 (*contre*), 6899 textes classés : 1 (*pour*).
Ce corpus est un peu déséquilibré, la catégorie 1 comporte 30% d'individus en moins que la première.
Voici le tableau des résultats obtenus.

		Validation Croisée			**Jeu de test**
classifieur		Précision	Rappel	Fscore	Fscore
RBF-Network	Classe 0	0.822	0.61	0.701	**0.6179**
	Classe 1	0.577	0801	0.671	
C4.5 Quinlan	Classe 0	0.503	0.758	0.605	**0.5940**
	Classe 1	0.503	0.758	0.605	
SVM	Classe 0	0.806	0.874	0.839	0.6907
	Classe 1	0.782	0.684	0.73	
Naive Bayes Multinomial	Classe 0	0.8	0.813	0.806	0.6855
	Classe 1	0.711	0.694	0.702	
Dirichlet	Classe 0	0.855	0.7375	0.7919	
	Classe 1	0.6727	0.8118	0.7357	

La chute des résultats sur le corpus de test est conséquente pour les classifieurs SVM et Naïve Bayes Multinomial. Par contre le classifieur RBF chute peu sur le corpus de test.

Nous pouvons tirer quelques enseignements des résultats précédents :

- Le classifieur SVM donne les meilleurs résultats.
- Le classifieur de Bayes Multinomial est très proche du meilleur.
- le classifieur RBF-Network chute un peu sur le jeu de test.
- Le classifieur Dirichlet est le meilleur sur une des classes.
- Le classifieur utilisant les arbres de décisions (C4.5) a des performances inférieures à tous les autres (nous avons constaté ce phénomène sur l'ensemble des corpus).

Remarque : Les résultats sur les jeux de test pour les classifieurs de Bayes et SVM ne sont pas officiels, ils ont été effectués après l'échéance, nous avons préféré soumettre d'autre résultats fondés sur des méthodes combinés.

Le résultat du classifieur RBF-Network a été publié dans le jeu 1 de nos soumissions.

Le résultat du classifieur C4.5 Quinlan a été publié dans le jeu 3 de nos soumissions.

3.5 Explication de la différence entre résultats en validation croisée et sur le corpus de test

La réduction de la taille de l'index est fondée sur l'appartenance de textes à des classes. Si nous réduisons l'index sur l'ensemble du corpus d'apprentissage, alors cela suppose que le vocabulaire utilisé dans les jeux d'apprentissage est exhaustif ou presque et que les jeux de tests n'utiliseront pas de mots différents de ceux appartenant à l'ensemble d'apprentissage.

Cette hypothèse est généralement fausse.

La procédure de validation croisée que nous avons utilisée utilise des vecteurs qui sont calculé sur l'index réduit de tout l'ensemble d'apprentissage.

Cette procédure est incorrecte car la réduction d'index doit être effectuée uniquement sur la partie de l'ensemble d'apprentissage qui est utilisée pour calculer le modèle de chaque sous ensemble utilisé dans chaque itération de la validation croisée.

Ainsi la procédure de validation croisée se transforme comme suit :

// S est un ensemble, x est un entier
Découper S en x parties égales S1, … , Sx
Pour i de 1 à x
Réduire l'index sur l'ensemble S- Si
Calculer les vecteurs avec l'index réduit obtenu
Construire un modèle M avec l'ensemble S - Si sur les vecteurs réduits
Evaluer une mesure d'erreur ei de M avec Si
Fin Pour

Dans notre cas x vaut 10.

En utilisant cette procédure nous avons constaté que les résultats en fscore sur l'ensemble d'apprentissage sont quasiment identiques à ceux obtenus en test et cela sur les quatre corpus testés.

4 Méthodes additionnelles pour améliorer les résultats

Nous avons tentés plusieurs approches pour améliorer les résultats. Elles sont de cinq types :

- Filtrage préliminaire des textes
- Ajout de synonymes
- Procédure de vote de classifieurs
- Utilisation des fonctions grammaticales des mots pour le calcul de l'index.
- Utilisation de bi-grammes en lieu et place de lemmes.

4.1 Filtrage préliminaire des textes

Ce traitement a été tenté uniquement sur le corpus 1. Nous avons effectué une procédure préliminaire pour élaguer les phrases considérer comme inutiles dans le corpus 1.

En lisant les textes du premier corpus nous avons constaté que une part non négligeable de ceux-ci contenaient à la fin du texte une partie commençant par l'expression : « Notre avis : ».

Nous avons considéré que la partie de texte qui suivait était uniquement constitué de phrases de jugement de valeur.

Ainsi nous avons utilisés ces phrases pour extraire dans tous les textes de l'ensemble d'apprentissage les phrases exprimant un jugement de valeurs.

Nous avons déjà traité cette problématique et montré notre approche de traitement dans (Plantié, 2006). Elle consiste à considérer un ensemble d'apprentissage contenant deux catégories :

- les phrases exprimant un jugement de valeur
- les phrases n'exprimant pas un jugement de valeur.

Le problème revient alors d'éliminer dans un texte les phrases n'exprimant pas un jugement de valeur, que l'on pourra considérer comme des phrases inutiles ou non pertinentes. Cette problématique reprend le thème du défi DEFT 2005 pour séparer des phrases de « Chirac » de celles de « Mitterand ».

Nous constituons donc un nouvel ensemble d'apprentissage constitué uniquement des textes et contenant l'expression « Notre avis : ». On constitue alors deux ensembles :

- classe 1 : les phrases de ces textes hors jugement de valeurs(n'appartenant pas à la rubrique « Notre avis : »)
- classe 2 : les phrases « jugement de valeur » (appartenant à la rubrique débutant par « Notre avis : »)

Un fois constitué cet ensemble d'apprentissage, nous calculons alors un nouveau classifieur. Ce classifieur interviendra sur les autres textes du corpus 1 (hors ceux contenant « Notre avis : ») pour éliminer les phrases étant classées « hors jugement de valeurs » ou classe 1. Le meilleur classifieur dans cette tâche a été le « Naïve Bayes Multinomial ». Nous avons obtenu pour ce classifieur un fscore de 84% par procédure de validation croisée.

Avec cette procédure nous obtenons donc un **nouveau** corpus 1, dans lequel chaque texte est une réduction du texte initial. Chaque texte ne contient que la partie considérée comme jugement de valeurs.

Puis nous appliquons sur ces textes la méthode générale présentée au chapitre précédent.

Hélas tous les tests que nous avons effectués en utilisant les différents classifieurs présentés précédemment donnent des résultats fscore inférieur d'environ 5 à 10%. Nous n'avons donc pas présenté de résultats pour cette méthode.

4.2 Ajout de synonymes

Comme nous l'avons expliqué dans la section précédente les textes du corpus de test comportent certains mots de vocabulaire qui ne sont pas obligatoirement présent dans le corpus d'apprentissage. Afin de prendre en compte ces nouveaux mots nous proposons d'utiliser les synonymes.

Nous avons déjà dans nos travaux rencontré cette problématique et montré notre approche de traitement dans (Plantié, 2006).

Voici l'algorithme :

```
Programme synonyme (texte)
    Établir la liste des lemmes (texte)
    Pour chaque lemme du texte à analyser :
           Rechercher dans l'index du corpus le lemme
           Si le lemme est présent : ne rien faire
           Sinon(le lemme est absent) :
                  Rechercher la liste des synonymes de (lemme)
                  Pour chaque synonyme de la liste
                     Si le synonyme est présent dans la liste des lemmes du corpus
                                Associer le lemme à l'indice de ce lemme du corpus
                                Fin du pour
                     Sinon rien
                  Fin Pour
           Fin si
    Fin Pour
```

Ainsi chaque mot inconnu est associé à un mot de l'index. Les vecteurs de chaque texte sont alors calculés selon la procédure ci-dessus. Nous pouvons alors utiliser la méthode du chapitre précédent.

Hélas tous les tests que nous avons effectués en utilisant les différents classifieurs présentés précédemment donnent des résultats fscore inférieur d'environ 10 à 15%. Nous n'avons donc pas présenté de résultats pour cette méthode.

4.3 Procédure de vote

Afin d'améliorer les scores obtenus précédemment nous avons tenté d'utiliser des procédures de vote.

Nous avons tentés deux approches :

- le vote à la majorité simple
- le vote tenant compte du fscore de chaque classifieur

4.3.1 Vote à la majorité simple

Nous avons appliqué cette procédure pour les quatre corpus.

Le principe est le suivant :

Nous prenons les résultats de trois classifieurs. Pour chaque texte évalué nous retenons la réponse qui emporte la majorité de 2 au moins sur 3.

Dans le tableau qui suit nous montrons les classifieurs retenus et les résultats obtenus :

Corpus	Classifieur 1	Classifieur 2	Classifieur 3	Résultat Fscore Sur jeu de test
Corpus 1	RBF-Network 8 clusters par classe	Naïve Bayes Multinomial	RBF-Network 6 clusters par classe	**0.4231**
Corpus 2	SVM	RBF-Network 4 cluster par classe	Naïve Bayes Multinomial	**0.7325**
Corpus 3	SVM	RBF-Network 4 clusters par classe	Naïve Bayes Multinomial	**0.4421**
Corpus 4	Naïve Bayes Multinomial	RBF-Network 8 cluster par classe	C4.5 Quinlan	**0.6706**

Ces résultats sont inférieurs à nos meilleurs résultats sur chacun des corpus. Les procédures de vote à la majorité sont donc peu convaincantes. Ces résultats ont été publiés dans le jeu 2 de nos soumissions.

4.3.2 Vote tenant compte du fscore de chaque classifieur

Nous avons tenté d'utiliser les résultats du rappel et de la précision pour chaque classifieur afin de trouver une procédure de vote.

Nous avons utilisé cette procédure sur le corpus 1 et sur le corpus 3.

Dans le corpus 1 nous avons sélectionné pour chaque classe le classifieur ayant le meilleur résultat de précision sur cette classe.

Ainsi à chaque classe correspondait un classifieur. Nous avons utilisé deux classifieurs pour cette procédure de vote : RBF-Network, et Naïve Bayes Multinomial.

Algorithme :

```
Affectation d'un classifieur à une classe, en prenant le classifieur donnant max de la
précision sur le corpus d'apprentissage.
Programme votеprecision (texte)
Si le classifieur en question affecte le texte en cours de traitement à cette classe
    alors nous prenons ce résultat.
Sinon nous prenons le classifieur suivant et l'on recommence la procédure.
Si aucun classifieur n'affecte le document en question à sa classe de prédilection,
    alors nous prenons le meilleur des deuxième choix.
```

Dans le corpus 3 nous avons sélectionné pour chaque classe le classifieur ayant le meilleur résultat de rappel sur cette classe.

Ainsi à chaque classe correspondait un classifieur. Nous avons utilisé deux classifieurs pour cette procédure de vote : RBF-Network, et SVM.

Algorithme :

```
Affectation d'un classifieur à une classe, en prenant le classifieur donnant max du rappel
sur le corpus d'apprentissage.
Programme voterappel (texte)
Si le classifieur en question affecte le texte en cours de traitement à cette classe
    alors nous prenons ce résultat.
Sinon nous prenons le classifieur suivant et l'on recommence la procédure.
Si aucun classifieur n'affecte le document en question à sa classe de prédilection,
    alors nous prenons le meilleur des deuxième choix.
Fin Si
```

Voici les résultats obtenus sur le jeu de test :

Corpus	Type de vote	Résultat Fscore Sur jeu de test
Corpus 1	Classe affectée au Classifieur ayant le Maximum de précision	**0.4715**
Corpus 3	Classe affectée au Classifieur ayant le Maximum de rappel	**0.4416**

Ces résultats sont inférieurs à nos meilleurs résultats sur chacun des corpus. Les procédures de vote à en tenant compte de la précision et du rappel sont donc peu convaincantes. Ces résultats ont été publiés dans le jeu 3 de nos soumissions.

4.4 Utilisation des fonctions grammaticales des mots pour le calcul de l'index

Dans le cadre de DEFT'07, nous avons appliqué un prétraitement supplémentaire. Ainsi, avant d'effectuer la classification des textes, nous avons cherché à améliorer les traitements « linguistiques » des textes. Dans ce but, les termes (groupes de mots respectant des patrons syntaxiques spécifiques) ont été extraits et exploités.

Avant d'extraire la terminologie, une première étape consiste à apposer des étiquettes grammaticales à chacun des mots du corpus. Une telle tâche a été effectuée avec l'étiqueteur de Brill (E. Brill, 1994) qui utilise des règles lexicales et contextuelles apprises à partir d'un corpus annoté.

Le tableau ci-dessous montre un fragment du corpus de relectures d'articles issu de DEFT'07 qui a été étiqueté avec (E. Brill, 1994).

Texte d'origine : **L'auteur devrait essayer de cibler...**	
Texte étiqueté : **L'/DTN :sg auteur/SBC :sg devrait/VCJ :sg essayer/VNCFF de/PREP cibler/VNCFF...**	
DTN : Déterminant de groupe nominal **SBC** : Substantif, nom commun **VCJ** : Verbe, conjugué	**VNCFF** : Verbe, non conjugué, infinitif **PREP** : Préposition **sg** : singulier

A partir des quatre corpus de DEFT'07 étiquetés, des patrons syntaxiques spécifiques peuvent être utilisés afin d'extraire les termes nominaux propres à chacun des corpus (termes de type "Nom Nom", "Adjectif Nom", "Nom Adjectif", "Nom Préposition Nom"). L'extraction de la terminologie a été menée avec le système(Roche, Heitz, Matte-Tailliez, & Kodratoff, 2004). Précisons que de nombreux travaux s'appuient sur l'utilisation de patrons syntaxiques pour extraire la terminologie (Bourigault & Fabre, 2000; Daille, 1994; Jacquemin, 1999).

Outre les termes nominaux qui ont été extraits et utilisés lors de l'étape de classification, nous nous sommes également intéressés à l'extraction des termes adjectivaux et adverbiaux. En particulier, les termes de type "Adverbe Adjectif" peuvent se révéler particulièrement pertinents pour certains corpus. A titre d'exemple, les termes *encore préliminaire*, *encore insuffisant*, *très significatif*, *difficilement compréhensible* obtenus à partir du corpus de relectures d'articles peuvent être assez discriminants pour classifier certains textes. Notons cependant que le corpus de relectures contient de nombreuses fautes d'orthographe (fautes d'accents, caractères manquants, etc.). Une telle situation peut expliquer les résultats décevants en utilisant uniquement la terminologie pour les tâches de classification. Une correction de ces fautes aurait pu significativement améliorer les tâches d'extraction de la terminologie et donc de classification. De plus, le nombre de termes extraits peut se révéler assez faible pour certains textes. Ceci met alors en défaut les méthodes statistiques qui ont été mises en oeuvre dans le cadre du défi.

Nous avons ensuite utilisé la méthode générale présentée dans la section précédente sur le corpus 1. Ainsi, nous avons considéré la liste des termes extraits comme l'index du corpus à partir duquel tous les textes ont été vectorisés. Puis la procédure classique a été implémentée : réduction d'index, classification, validation croisée. Le nombre de termes extraits peut se révéler assez faible pour certains textes ce qui réduit significativement la taille de l'index. Ceci met alors en défaut les méthodes statistiques qui ont été mises en oeuvre dans le cadre du défi. Egalement les termes sélectionnés après réduction d'index ne sont pas suffisamment significatifs pour représenter la diversité des textes. Ceci peut expliquer les résultats fortement dégradés avec cette méthode.

4.5 Utilisation de bi-grammes en lieu et place de lemmes

Nous avons tenté d'extraire les bi-grammes du corpus mais uniquement dans un premier temps les bi-grammes : Adjectif-Adverbe ou inversement. Cette extraction s'est effectuée avec la même méthode qu'au paragraphe précédent.

Nous avons ensuite utilisé la méthode générale présentée au chapitre précédent sur le corpus 1. C'est-à-dire que nous avons considéré la liste des bi-grammes extraits comme l'index du corpus à partir duquel tous les textes ont été vectorisés. Puis la procédure classique a été implémentée : réduction d'index, classification, validation croisée.

Hélas tous les tests que nous avons effectués en utilisant les différents classifieurs présentés précédemment donnent des résultats fscore fortement inférieur. Nous n'avons donc pas présenté de résultats pour cette méthode.

Nous aurions souhaité poursuivre cette expérience par l'extraction de tous les bi-grammes du corpus et ensuite appliquer la procédure de réduction d'index. Hélas le temps nous a manqué.

5 Conclusion et perspectives

Ce défi fut passionnant. Nous avons cependant manqué de temps et surtout de machines très performantes en terme d'espace mémoire notamment. Certains algorithmes en effet ont pris plusieurs heures pour donner des résultats.

Nous avons passé en revue plusieurs méthodes de classification. Mais nous devons approfondir nos essais sur les différentes méthodes de classification. Pour améliorer nos résultats nous devons également effectuer des prétraitements plus poussés, car les classifieurs montrent leurs limites sur la plupart des corpus. En particulier nous fondons de nombreux espoirs sur :

- une application plus approfondie de la méthode de Dirichlet,
- la combinaison de classifieur plus précise et plus adaptée que les procédures de vote simple présentées ici,
- L'utilisation plus généralisée des bi-grammes,
- La combinaison de méthodes fondées sur les lemmes et sur les bi-grammes.

Références

Bourigault, D., & Fabre, C. (2000). Approche linguistique pour l'analyse syntaxique de corpus. *Cahiers de Grammaires, 25*, 131-151.

Cover, & Thomas. (1991). *Elements of Information Theory*: John Wiley.

Daille, B. (1994). *Approche mixte pour l'extraction automatique de terminologie : statistiques*

lexicales et filtres linguistiques. Unpublished Ph. D. thesis, Université Paris 7.

E. Brill, I., Vol. (1994). Some advances in transformation-based part of speech tagging. *AAAI, 1*, 722-727.

Jacquemin, C. (1999). *Syntagmatic and paradigmatic representations of term variation.* Paper presented at the 7th Annual Meeting of the Association for Computational Linguistics (ACL'99).

Joachims, T. (1998). *Text Categorisation with Support Vector Machines : Learning with Many Relevant Features.* Paper presented at the ECML.

MacQueen., J. B. (1967). *Some Methods for classification and Analysis of Multivariate Observations.* . Paper presented at the 5th Berkeley Symposium on Mathematical Statistics and Probability.

Nallapati Ramesh, M. T., Robertson Stephen. (2006). *The Smoothed-Dirichlet distribution : a new building block for generetive topic models*.

Parks, J., & Sandberg, I. W. (1991). « Universal approximation using radial-basis function networks ». In *Neural Computation* (Vol. 3, pp. 246-257).

Plantié, M. (2006). *Extraction automatique de connaissances pour la décision multicritère.* Unpublished Thèse de Doctorat, Ecole Nationale Supérieure des Mines de Saint Etienne et de l'Université Jean Monnet de Saint Etienne, Nîmes.

Platt, J. (1998). Machines using Sequential Minimal Optimization. . In *Advances in Kernel Methods - Support Vector Learning*: B. Schoelkopf and C. Burges and A. Smola, editors.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. (Morgan Kaufmann ed.). San Mateo (CA US) Morgan Kaufmann.

Roche, M., Heitz, T., Matte-Tailliez, O., & Kodratoff, Y. (2004). *EXIT : Un système itératif pour l'extraction de la terminologie du domaine à partir de corpus spécialisés.* Paper presented at the JADT'04.

Synapse. (2001). Synapse Analyser: Synapse. "Synapse Analyser." http://synapse-fr.com, .

Wang, Y., Hodges, J., & Tang, B. (2003). Classification of Web Documents using a Naive Bayes Method. *IEEE*, 560-564.

Classification de texte et estimation probabiliste par Machine à Vecteurs de Support.

Anh-Phuc.Trinh[1]

[1] Laboratoire d'Informatique de Paris 6, Université de Pierre et Marie Curie,
104, avenue du Président Kennedy, 75016 Paris.
anh-phuc.trinh@poleia.lip6.fr

Résumé : La classification de documents D en classes pré-déterminées Y est simplement présentée comme le problème d'estimation probabiliste de la probabilité a posteriori $P(Y/D)$. Nous présentons ici une méthode basée sur le modèle de Machine à Vecteurs de Support afin de réaliser cette tâche. Il y a deux approches correspondant à deux niveaux de présentation des corpus : en documents et en phrases, que nous avons analysées dans ce défi.

Mots-clés : Machine à Vecteur de Support, Classification de Texte, Estimation Probabiliste

1 Introduction

La classification de texte est une des tâches primitives dans le domaine du Traitement Automatique des Langues (TAL), de la Recherche d'Information et des Algorithmes d'Apprentissage (AA). Les premiers efforts sont venus de la Recherche d'Information, avec (Salton et al., 1983), qui ont transformé leurs documents en vecteurs de termes. Le poids Tf*Idf (Salton et al., 1988) a été approuvé et utilisé comme la formule « standard » pour un traitement de documents.

De plus en plus, nous avons besoin de représenter les documents à un niveau plus profond, comme dans le cas des documents structurés ou semi-structurés et les documents XML qui ont une structure arborescente. Cependant, les chercheurs continuent de s'intéresser aux documents en texte simple car ils sont plus proches des langues naturelles, même si les documents structurés sont plus faciles à représenter sur les machines. Si les documents en plein texte peuvent être représentés à un niveau plus profond, nous estimons que la classification peut être améliorée par rapport aux documents entiers. Ainsi, allons-nous diviser les documents en phrases puis les analyser.

En matière de modèles d'apprentissage (MA) pour classifier des documents, nous pouvons choisir soit le modèle d'apprentissage non supervisé, soit le supervisé. Nous décidons de prendre le modèle d'apprentissage supervisé appelé Machine à Vecteurs de Support (Vapnik, 1998). Sur le problème d'estimation probabiliste par la MVS, les articles pertinents sont (Vapnik, 1998), (Platt, 2000), (Friedman et al., 1996), (Hastie et al., 1996), (Tax et al., 2002) et (Wu et al., 2004). Les deux premiers articles décrivent la classification binaire, les quatre derniers traitent du sujet de la classification de multi-classes. Nous avons également trouvé une thèse relative à la classification des documents en texte par la MVS (Joachim, 2002).

La tâche proposée par DEFT07 consistait à classifier les documents d'un corpus en deux classes (corpus de débats parlementaires) ou en plusieurs (les trois autres corpus). Nous avons donc décidé d'utiliser les méthodes de (Platt, 2000) et de (Wu et

al., 2004). En réalité, le dernier article poursuit le travail du premier, donc leurs formules sont identiques.

2 Présentation des corpus

Dans cette section, nous avons étudié les différences entre les corpus pour obtenir leurs propriétés statistiques. Afin de les pré-traiter, la technique de sac de mots a été appliquée, et nous donnons également au fur et à mesure les résultats de cette technique avec des commentaires.

2.1 Représentation générale des quatre corpus du défi

Il y a quatre corpus différents dans le cadre du défi : « Critiques de films », « Tests de jeux vidéo », « Relectures d'articles » et « Débats parlementaires ». Tous sont au format XML, donc ils ont besoin d'être traités correctement, tout en gardant les balises qui contiennent des informations indispensables. L'outil XML de Java a été utilisé lors du traitement de ces quatre corpus, cela nous permet d'extraire un sac de mots pour chaque corpus et de saisir des informations d'évaluation primaires de chaque document.

Les données des quatre corpus viennent de sources très variées. Les deux premiers semblent venir de deux sites Web de divertissement (vidéo, livres, spectacles, bandes dessinées, jeux vidéo). Les textes qui les constituent sont très variés, se composant souvent d'un petit résumé du contenu du jeu ou du livre, puis de commentaires personnels. A première vue, le niveau "mauvais", "moyen" ou "bon" semble facile à classifier en tenant compte du sens des phrases. Les deux autres, "débats parlementaires" et " relectures d'articles" semblent issus de sources plus officielles. D'une part, les débats parlementaires comprennent toujours des phrases courtes avec des termes politiques et très formels. Nous pouvons facilement définir l'intention du locuteur et ainsi classer son discours dans la catégorie "pour" ou " contre". D'autre part, nous remarquons que, dans le corpus de relectures d'articles, les textes sont plus riches et souvent il y a beaucoup de commentaires. Au niveau de la structure, les documents sont composés de phrases constituant un en-tête et sont structurés jusqu'à l'identifiant n° 275 . A partir de cet identifiant, les documents n'ont plus de phrase d'en-tête. Ce corpus nous a donné des résultats différents par rapport aux autres.

Parmi les corpus donnés, celui des relectures d'articles est le plus petit : 1471 Ko, le plus grand corpus est celui des débats parlementaires : 24 620 Ko. Par rapport aux autres défis que nous avons rencontrés, ces données sont assez grandes et en même temps assez variées. Le nombre de phrases de chaque document est très variable : le corpus "débat parlementaire" a souvent une phrase pour chaque avis ; le corpus "jeux vidéo" a en moyenne un paragraphe (de chaque avis) assez long par rapport aux autres documents. Le corpus le plus particulier est "relectures d'article" : il commence par des paragraphes assez structurés : originalité, commentaire ... puis finit par supprimer toute structure et n'a plus qu'une phrase courte pour chaque avis. Ce sont des données brutes et sans pré-traitement. Cela oblige les participants à choisir les techniques pour les pré-traiter. Ce fait augmente le temps de pré-traitement des données. Nous avons choisi la technique de sac de mots. C'est une technique classique et simple pour traiter les données textuelles que nous vous présentons maintenant.

Corpus d'apprentissage	Taille	Nombre de phrases	Nombre de documents
Critiques de cinéma, spectacles, livres, BD et CD	4Mb	34622	2074
Test de jeux vidéo	17Mb	145543	2537
Relectures d'articles de conférences	1,4Mb	11473	881
Débats parlementaires	24Mb	160399	17299

Tableau 1 – Les corpus d'apprentissage

2.2 Sac de mots

Nous pouvons constater que le sac de mots (William et al, 1995) est un dictionnaire de vocabulaire particulier. C'est une technique qui filtre les mots d'un document, puis ajoute les informations sur chaque mot filtré telles que : les indices de fréquence du mot dans le document ou dans les autres documents qui constituent le corpus.

Avec les indices d'un sac de mots, nous pouvons convertir un document en un vecteur, ainsi nous pouvons représenter un document dans un espace vectoriel. Après avoir constitué le vecteur qui correspond au document étudié, nous pouvons utiliser les formules mathématiques pour calculer puis définir la classification du document. D'après nos calculs, le sac de mots du corpus de "relectures d'articles" est plus grand par rapport à la taille de ce corpus, ce qui veut dire que les mots qui le constituent sont très variés et très diversifiés : un mot n'apparaît souvent qu'une seule fois dans le document et parfois n'existe pas dans les autres documents du corpus. Les trois corpus qui restent ont un sac de mots assez proportionnel à leur taille.

Comme nous avons montré ci-dessus, la technique du sac de mots est une technique très simple, qui récupère les mots d'un document sans prendre en compte le sens de ces mots. Faute d'outils logiciels qui puissent traiter les mots d'une façon plus approfondie : diviser les mots selon les thèmes, leur signification, ou leur domaine, etc ... nous avons adopté une solution statistique pour traiter le corpus : prendre uniquement en compte la fréquence d'un mot dans un document et dans le corpus. Une des limites de cette technique réside dans le fait d'être très sensible à la casse : par exemple, le sac de mots va prendre "BOn" et "bon" pour deux mots, ce qui augmente sans raison pertinente la taille du sac de mots. C'est la raison pour laquelle, en utilisant cette technique, nous avons tendance à diminuer le nombre de mots sans trop perdre d'information (voir la section 4.2). Le sac de mots est également utilisé afin de représenter une phrase du document (voir la section 3.2), nous constatons que l'apparition d'un mot est de la même importance que la fréquence de ce mot dans une phrase.

Corpus d'apprentissage	Nombre total de mots	Taille du sac de mots
Critiques de cinéma, spectacles, livres, BD et CD	792214	48489
Test de jeux vidéo	3084878	56185
Relectures d'articles de conférences	218588	13395
Débats parlementaires	3738562	46654

Tableau 2 – Les sacs de mots de chaque corpus d'apprentissage

Nous avons ci-dessous un ensemble des accents et numéraux enlevés du corpus, le reste constitue le sac de mots. Nous utilisons la classe « Tokenizer » de Java pour récupérer les instances. Les sacs de mots sont ensuite vérifiés plusieurs fois manuellement pour assurer que les mots filtrés gardent bien leur sens initial. Comme nous avons précisé dès le début, le sac de mots est une technique simple, utilisé dans le cas où nous n'avons ni dictionnaire, ni thésaurus. D'autre part, l'un des avantages de cette technique est que l'information est conservée telle quelle : si l'utilisateur utilise des mots spécifiques, le sac de mots les ramasse quand même et les considère comme les autres mots.

. \(\) :,?! +-%&°$'0123456789*[]/<>;\n§=@#|{}} {}

3 Approches

Nous avons étudié deux approches sur les quatre corpus d'apprentissage. La première est de représenter le document comme l'unité à analyser, et l'autre est de représenter la phrase comme l'unité à analyser. La MVS est le modèle d'apprentissage et nous avons utilisé sa fonction d'estimation probabiliste pour rétablir la distribution a posteriori $P(Y/D)$

3.1 Représentation des documents

3.1.1. Définition et formules

$$P(Y/D) \text{ avec } D=\left(mot_1:poids_1, mot_2:poids_2, \ldots, mot_n:poids_n\right)$$

Où

Y correspond aux classes pré-déterminées

mot_i correspond à chaque entrée du sac de mots qui a été présenté en section précédente.

La probabilité a posteriori $P(Y/D)$ présente une mise en correspondance (mapping) de documents avec leur propre classe. La probabilité de chaque classe Y sachant le document D nous permet de dire à quelle classe ce document D appartient. Notre stratégie est la suivante : nous avons d'abord essayé de représenter des documents de la manière où ils sont les plus séparables possible, ensuite le modèle d'apprentissage MVS (Machine à Vecteur de Support) va les discriminer par son algorithme d'apprentissage non linéaire qui permet de séparer des données bruitées. Enfin, une estimation probabiliste de la classification qui vient d'être construite va conclure notre tâche, et cette estimation probabiliste a besoin de prendre en compte la capacité de discrimination des données par le modèle d'apprentissage MVS.

3.1.2. Le poids Tf*Idf

Le poids Tf*Idf (Salton et al., 1988) a été largement utilisé dans le domaine de la Recherche d'Information et de la Fouille de données Textuelles. Ce poids mesure statistiquement l'importance d'un mot d'un document par rapport au corpus entier. Cette importance est représentée à la fois par son nombre d'apparitions dans ce document et par l'inverse du nombre de documents contenant ce mot dans le corpus.

Il y a beaucoup de formes du poids Tf*Idf (Church et al., 1995), nous avons décidé d'employer celui qui suit

$$\text{TfxIdf} = \text{Term\_Frequency X Inverse\_Document\_Frequency}$$

avec

$$Idf = \ln\left(\frac{D}{(d_i \text{ contient } m_j)}\right)$$

Nous prenons le logarithme naturel du nombre de documents contenant ce mot, dans la mesure où la valeur de cette fonction varie entre $(0, +\infty)$. Grâce à ce choix, le poids Tf*Idf devient une valeur réelle et chaque document va se transformer en un vecteur réel $\Re^n$. Quand le poids Tf*Idf d'un mot est grand, cela nous indique que ce mot apparaît de nombreuses fois dans ce document et rarement dans le corpus entier, autrement dit, ce mot est très valable, parce qu'il permet de discriminer son document d'origine par rapport aux autres. Nous pouvons également mesurer la proximité ou l'éloignement entre deux documents par le cosinus (ou le produit scalaire normalisé) de l'angle que forment les vecteurs qui les représentent. L'apprentissage non supervisé basé sur cette mesure peut placer un document dans une classe en utilisant simplement un seuil adapté aux données. Cette approche classique n'assure pas que des données bruitées soient correctement classifiées par le modèle d'apprentissage. Le bruit des données existe toujours pour des raisons subjectives, par exemple, des mots que nous avons choisis ne sont pas représentatifs, leur fréquence dans chaque document ne suffit pas pour éloigner ce document des autres dans l'espace vectoriel.

La sélection des mots réduit la dimension de l'espace vectoriel dont chaque document est un vecteur et en même temps augmente la capacité de discrimination entre les documents. La dimension de l'espace vectoriel correspond à la complexité du modèle d'apprentissage, c'est-à-dire que plus il y a de mots à prendre en compte, plus il y a de variables à évaluer. Nous devons tenir compte d'un ensemble de mots qui sont souvent utilisés en français : des pronoms, des prépositions, des conjonctions etc... Leurs valeurs de poids sont toujours très faibles parce qu'ils apparaissent dans tous les documents, donc nous pouvons les enlever du sac de mots. Cependant cette technique ne s'applique que si la taille du corpus est assez grande et le vocabulaire du sac de mots est large. Les résultats de cette technique vont également être présentés (voir la section 4.2). Néanmoins, nous ne les avons pas soumis dans le cadre de ce défi.

Débats parlementaires		
mot	nombre total	idf
de	205135	17105
la	111334	16126
l	88199	15754
à	76529	15557
le	75384	15577
les	72674	14753
des	71309	14544
et	67158	14702
que	51388	14375
du	39091	12114

Les dix mots les plus fréquents dans le corpus de débats parlementaires

3.1.3. Estimation probabiliste par MVS (Machine à Vecteur de Support)

a) Définition et formules (Vapnik, 1998) (Muller et al., 2001)

Une Machine à Vecteur de Support est une technique de discrimination. Elle consiste à séparer deux (ou plusieurs) ensemble de points de données par un (ou plusieurs) hyperplan(s). Considérons un ensemble de points de données $\{(D_1,Y_1),(D_2,Y_2),\ldots,(D_l,Y_l)\}$ avec $Y_i=\{-1,+1\}$. Un hyperplan $w\cdot D-b=0$, qui sépare cet ensemble des points de données, possède également deux hyperplans parallèles $w\cdot D-b=+1$ et $w\cdot D-b=-1$ tels que la marge qui les sépare est égale à $2/\|w\|^2$. Le problème est de minimiser l'inverse de cette marge, c'est-à-dire,

$$\min\frac{1}{2}\|w\|^2 \text{ soumis à } Y_i\times(w\cdot D_i-b)\geq 1 \quad \forall i=\overline{1,l} \quad (3)$$

La fonction de signe $f(D)=signe(w\cdot D-b)$ nous indique de quel côté de la marge un document D se trouve, nous l'appelons une fonction de décision. En cas de données bruitées, l'hyperplan ne permet pas de séparer totalement certains points de données $\{D_i,Y_i\}_1^l$. En présentant des pénalités d'erreurs non nulles $\xi_i\geq 0$, chaque point de données D_i possède une valeur de pénalité, donc l'optimisation devient un compromis entre une grande marge et des pénalités d'erreurs.

$$\min\frac{1}{2}\|w\|^2 \text{ soumis à } Y_i\times(w\cdot D_i-b)\geq 1-\xi_i \quad \forall i=\overline{1,l} \quad (4)$$

b) Estimation probabiliste en cas de classes multiples

Le cas le plus simple est la classification binaire $Y_i=\{-1,+1\}$, correspondant au corpus de débats parlementaires, la distribution a posteriori $P(Y/D)$ est estimée par la fonction sigmoïde, c'est-à-dire :

$$P(Y=+1/D)=\frac{1}{(1+\exp(A\times f(D)+B))} \quad (5)$$

où A et B sont estimés en réduisant au minimum la fonction négative de vraisemblance. La vraisemblance de cette distribution des données positives est la fonction

$$L(z)=\sum_{i=1}^{l}\left(t_i\log(p_i)+(1-t_i)\log(1-p_i)\right) \quad (6)$$

Avec $z=(A,B)$ et $t_i=\frac{Y_i+1}{2}$ est la probabilité de la cible.

(Platt et al., 2000) nous propose deux techniques afin de bien estimer les A et B : la première en partageant le corpus d'apprentissage en deux (70% pour entraîner la MVS avec la fonction de décision $f(D)$, les 30% restants pour estimer A et B dans la fonction sigmoïde), la deuxième est d'utiliser la validation croisée en n-partitions.

Dans un cas plus compliqué, la classification en k-classes (k>2), correspondant aux autres corpus, nous avons normalement deux stratégies pour appliquer la technique de discrimination : "un-contre-les autres" ou "un-contre-un". Il s'agit de construire plusieurs hyperplans (k avec celle de "un-contre-les autres" et k(k-1)/2 avec celle de "un-contre-un"). A chaque hyperplan, nous avons une distribution proportionnelle de la classification binaire entre deux classes différentes

$$rij \approx P(Y=i/Y=i \text{ ou } j,D) \text{ avec } i \neq j \quad (7)$$

sur un sous-ensemble de documents dans le corpus. Basé sur la stratégie "un-contre-un", (Wu et al., 2004) ont proposé une méthode assez coûteuse pour reconstituer la distribution de multi-classification $P(Y/D)$ à partir de cet ensemble $R=\left|rij\right|_{i\neq j}$ en résolvant le problème d'optimisation suivant

$$\frac{1}{2}\sum_{i=1}^{k}\sum_{j\neq i}\left(r_{ji}P(Y=i/D)-r_{ij}P(Y=j/D)\right)^2 \quad (8)$$

$$\text{soumis à } \sum_{i=1}^{k}P(Y=i/D)=1, P(Y=i/D)\geq 0, \forall i$$

La fonction est obtenue en introduisant (7) dans (8)

$$P(Y=i/Y=i \text{ ou } j,D)\cdot P(Y=j/D)=P(Y=j/Y=i \text{ ou } j,D)\cdot P(Y=i/D)$$

Certainement, nous avons également d'autres méthodes pour estimer la probabilité a posteriori $P(Y/D)$ comme (Friedman et .al, 1996), (Price et al. 1995), (Hastie et al., 1998). Alors que les deux premières sont simples, la dernière est proche du problème d'apprentissage statistique. Les distributions proportionnelles $R=\left|rij\right|_{i\neq j}$ jouent le rôle des données d'apprentissage et les $P(y=i/D)$ sont des variables qui doivent être évaluées en minimisant la distance de Kullback-Leiber (KL).

Corpus d'apprentissage	A1	B1	A2	B2	A3	B3
Critiques de cinéma, BD, livres, spectacles et CD	-44,275	38,623	-13,75	11,289	-17,165	-14,213
Test de jeux vidéo	-13,266	9,42	-4	1,87	-20,59	-11,33
Relectures d'articles de conférences	-1,205	0,558	-1,161	-0,47	-3,062	-1,771
Débats parlementaires	-4,176	-2,689				

Tableau 3 Les paramètres A et B de la fonction sigmoïde.

Le tableau ci-dessus montre trois paires de $\left(A_i,B_i\right) i=1,3$ correspondant à trois hyperplans qui sont également nos trois classificateurs. Ces paramètres $\left(A_i,B_i\right)$ ont été estimés par la validation croisée avec 5-partitions. A chaque nouvelle donnée d'entrée D, par exemple une donnée de test, nous avons besoin de résoudre le problème (8) afin de trouver les informations probabilistes $P(Y=i/D)$. La complexité de (8) est proportionnelle au nombre de classes pré-déterminées $(k=3)$, néanmoins le problème (8) est un problème d'optimisation, ainsi donc nous avons eu besoin de beaucoup temps pour prédire D. Cela est peut-être un inconvénient de cette méthode, en particulier, si l'ensemble de données de test est large.

La Figure 1 illustre une représentation partielle des données de test du corpus Critique, le trait continu représente la probabilité proportionnelle entre les deux classes « bon » et « mauvais », autrement dit r_{ii}, le trait discontinu représente cette probabilité a posteriori : $P(Y=bon/Y=bon \text{ ou } mauvais,D)$. Nous pouvons observer que la variabilité de cette probabilité a posteriori est souvent plus basse que la probabilité proportionnelle après avoir résolu le problème (8).

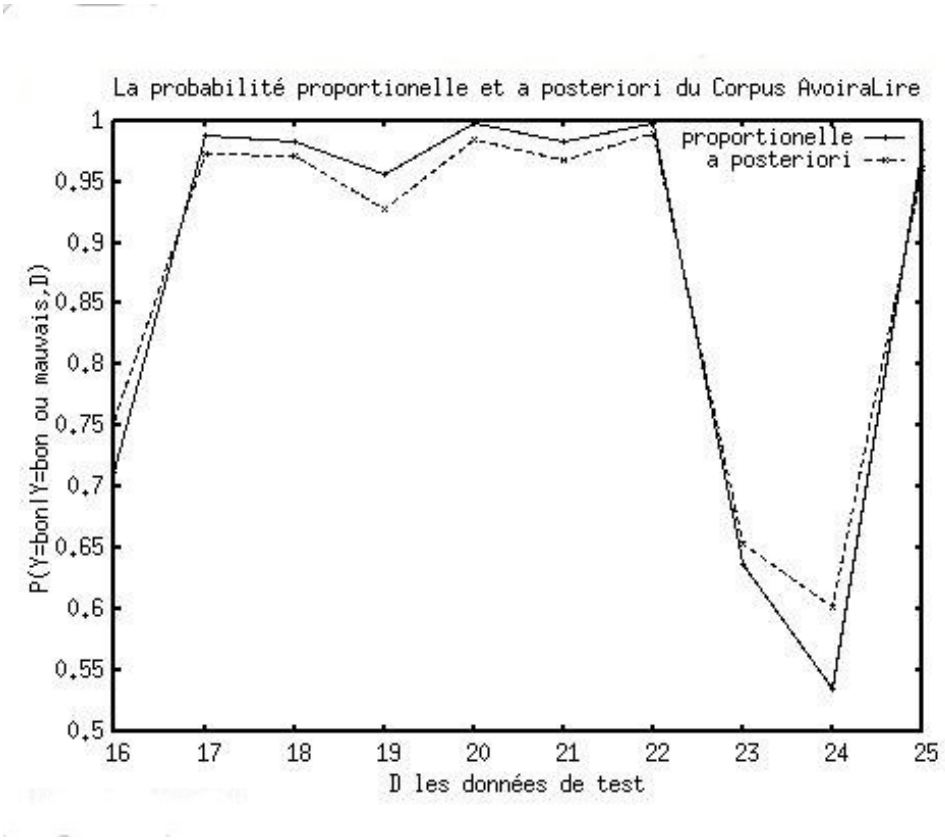


Figure 1 – Les courbes des probabilités a posteriori et proportionnelle du corpus des Critiques

3.2. Extraction locale de documents

3.2.1. Définition et formules

$$P(Y/D)=P\left(Y/phrase_1,phrase_2,\ldots,phrase_m\right)=\sum_{j=1}^{m}P\left(Y/phrase_j\right)\ avec\ \ j=\overline{1,m} \quad (9)$$

Où

$$phrase=\left(mot_1:poids_1,mot_2:poids_2,\ldots,mot_n:poids_n\right)\ avec\ poids=\left[0,1\right]$$

Jusqu'ici, nous avons considéré le document comme l'unité à analyser. Un document peut être autrement décrit comme une série de phrases, chaque phrase présentant sémantiquement une idée complète de l'auteur. Cette hypothèse nous conduit à la deuxième approche de notre travail. Comme chaque document se compose de plusieurs phrases, nous avons besoin de déterminer une technique pour réunifier les informations des phrases. Nous avons décidé simplement de prendre en charge des informations qui sont représentées par la probabilité a posteriori de chaque phrase $P\left(Y/phrase_j\right)$ dans un document. Enfin, nous procédons à une normalisation afin d'obtenir la probabilité a posteriori totale $P\left(Y/D\right)$. Le but de cette approche est de découvrir l'extraction locale de documents.

3.2.2. Représentation de la phrase par un vecteur binaire

D'abord les phrases sont extraites des documents, elles sont ensuite converties en un vecteur binaire. Nous constatons que l'apparition d'un mot est aussi importante que la fréquence de ce mot dans une phrase, ainsi elles sont représentées par les vecteurs binaires. Le vecteur binaire est souvent très faible en matière de nombre d'indices, c'est-à-dire, nous avons vu que des phrases ne contiennent qu'un seul mot. Nous espérons qu'au niveau de la phrase, les documents seront bien caractérisés et que la MVS pourra séparer facilement nos documents. En cas de documents spéciaux, par exemple, le corpus de relectures d'articles, il y a plusieurs phrases qui n'ont qu'un mot d'en-tête (originalité, commentaire...), ainsi la MVS vise à ranger toutes ces phrases dans une seule classe (voir la section 4.2).

Corpus d'apprentissage	Phrase la plus longue	Longueur moyenne de phrases	Phrase la plus courte
Critiques de cinéma, spectacles, livres, BD et CD	200	22	1
Test de jeux vidéo	171	21	1
Relectures d'articles de conférences	206	19	1
Débats parlementaires	324	23	1

Tableau 4 Les phrases pour chaque corpus d'apprentissage.

3.2.3. Discrimination locale et vraisemblance globale

La Figure 2 illustre une représentation partielle des données de test du corpus Critiques. Le trait discontinu représente la probabilité a posteriori $P(Y=mauvais|D)$ dans le cas des phrases binaires, le trait continu représente la même probabilité dans le cas de documents. Comme dans la figure précédente (voir la section 3.1.3.b), nous pouvons constater que la variabilité de la probabilité a posteriori de phrases est souvent plus basse que celle de documents : nous pouvons dire d'une autre façon que celle des documents est plus « pointue » que celle des phrases. La probabilité des documents est plus discriminante que la probabilité des phrases. Nous répétons encore une fois que le fait de distinguer les données ne signifie pas que le modèle a une bonne valeur de F-score. On arrive fréquemment à une valeur de précision plus grande en diminuant la valeur de rappel.

La MVS est un modèle de discrimination. Il en résulte que sa valeur de précision est en général plus élevée que celle du rappel. Si nous décidons de discriminer localement des documents, la valeur de précision semble encore plus élevée (voir la section 3.1.3.b et 4.2), néanmoins la valeur moyenne entre eux (F-score) est réduite. Nous estimons qu'il existe toujours un compromis entre la tendance de discrimination locale et celle de vraisemblance globale. Pour revenir sur la section d'estimation probabiliste précédente, après l'étape de discrimination locale entre sous-ensemble de données, nous avons besoin de rétablir la probabilité a posteriori totale P(Y|D) en minimisant la distance de Kullback-Leiber K(z||D).

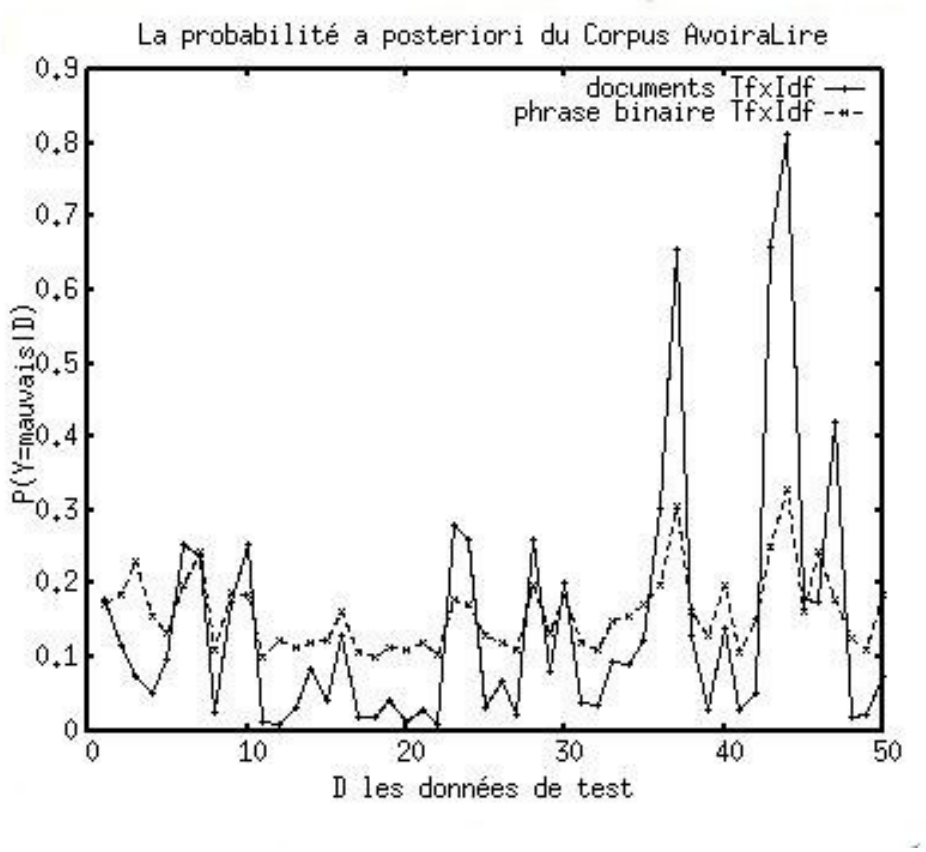


Figure 2 : La probabilité a posteriori du corpus Critiques en deux approches.

4 Résultats

Dans cette section, nous donnons les résultats définitifs des deux étapes d'apprentissage et de test. Les résultats entre les deux étapes sont assez cohérents et stables, la valeur d'exactitude (A) que nous présentons vient de (Wu et al., 2004). Les trois valeurs de précision (P), de rappel (R) et de F-score (F) sont venues de DEFT07. Les corpus d'apprentissage ont été répartis pour 60% en données d'entraînement et 40% en données de test. Et voici la liste des valeurs d'évaluation :

$$(A)\text{ccuracy} = \frac{\text{nombre correct}}{\text{nombre total}}$$

$$(P)\text{récision} = \frac{\text{nombre correct}}{\text{nombre total de cible}}$$

$$(R)\text{appel} = \frac{\text{nombre correct}}{\text{nombre total de prédict}}$$

$$(F)\text{score} = \frac{2 \times P \times R}{P + R}$$

4.1 Résultats soumis

Nous remarquons que les résultats dans les données de test et les données d'apprentissage sont assez homogènes. Premièrement, la valeur de précision (P) est souvent un peu plus élevée que la valeur de rappel (R) quand on utilise MVS. Deuxièmement, la valeur moyenne entre ces deux valeurs (F) diminue quand on considère le niveau local de la phrase. Quand on analyse la phrase, la valeur de précision augmente considérablement mais ne compense pas la perte sur le rappel.

Diviser le corpus d'apprentissage 60% et de test 40%		Les résultats de test		
Corpus d'apprentissage	(A)ccuracy	(P)récision	(R)appel	(F)score
Critiques de cinéma, spectacles, livres, BD et CD	0,60576	0,553125	0,53125	0,541954
Test de jeux vidéo	0,70866	0,678847	0,640234	0,658975
Relectures d'articles de conférences	0,42937	0,458529	0,400009	0,427275
Débats parlementaires	0,70885	0,684401	0,667297	0,675741

Tableau 6 (F-score, Précision, Rappel) Les résultats de test dans le cas des documents

Regardons de plus près le cas où l'unité à analyser est un document. Les corpus des articles à lire ont des valeurs assez petites car ils contiennent peu d'information et ces informations sont répétitives. Nous mettons particulièrement l'accent sur ce corpus pour deux raisons. La première raison est qu'il est petit, les informations sont homogènes donc non condensées, ce qui conduit à la réduction des valeurs (P, R, F). Ainsi nous ne pouvons pas représenter séparément les documents. La meilleure solution est de donner des caractéristiques aux documents en les décrivant davantage. Par exemple, nous pouvons ajouter les caractéristiques concernant la longueur du document en comptant les mots de celui-ci, ou même ne pas éliminer les accents dans le sac de mots, c'est-à-dire considérer un accent comme un mot. La deuxième raison est l'utilisation de masse de ce type de document. Avec l'expansion de l'Internet, les utilisateurs ont de plus en plus tendance à remplir les formulaires préexistants qu'à écrire un long paragraphe. L'utilisateur trouve que c'est plus facile de remplir un formulaire, mais rechercher et/ou prédire l'information dans un formulaire constitue un défi. D'une autre façon, on peut considérer le corpus des articles comme un échantillon de textes, avec une petite quantité de textes appartenant à un grand ensemble de textes, et notre mission est de reconstituer ce grand ensemble.

Au départ, nous avons obtenu une valeur d'exactitude (A) très significative avec les deux corpus Débats et Jeux vidéo (les valeurs A de ces deux corpus sont homogènes et sont légèrement supérieures à 70%). Nous nous concentrons ensuite sur l'analyse des deux corpus moins volumineux qui sont celui des Relectures et celui des Critiques. Ce qui est intéressant se trouve justement dans le volume de ces deux corpus. Comme ils sont tous les deux de petite taille, le filtrage des documents prend beaucoup moins de temps par rapport aux deux premiers corpus. Ce qui reste à faire est de diviser les documents en phrases. De cette manière on augmente considérablement les données à apprendre, autrement dit, on augmente les données du défi. Un exemple simple : le corpus Relectures d'articles contient 881 documents qui sont composés de 11473 phrases (voir le tableau 1). Ainsi, le nombre de vecteurs augmente de 881 à 11473.

Diviser le corpus d'apprentissage 60% et de test 40%		Les résultats de test		
Corpus d'apprentissage	(A)ccuracy	(P)récision	(R)appel	(F)score
Critiques de cinéma, spectacles, livres, BD et CD	0,55528	0,819650	0,349448	0,489993
Test de jeux vidéo	0,56102	0,788606	0,458202	0,579625
Relectures d'articles de conférences	0,45197	0,508588	0,432168	0,467274
Débats parlementaires	0,67360	0,683378	0,647842	0,665136

Tableau 7 (F-score,Précision,Rappel) Les résultat des tests dans le cas des phrases

Le tableau ci-dessus contient les résultats de la représentation au niveau local. Pour les corpus auxquels on s'intéresse particulièrement, tels que les Relectures d'articles, la remarque est que toutes les valeurs augmentent en même temps (A, P, R, F) par rapport à la représentation au niveau des documents. Ce résultat renforce la validité de notre principe de calcul des valeurs. Nous attirons l'attention sur ce fait car ces mêmes valeurs, dans les autres corpus, sont moindres. Ce n'est pas surprenant vis-à-vis des trois autres. Nous pouvons expliquer cette différence par la figure ci-dessous. En fait, un seul corpus, Relectures d'articles, possède une variabilité de la probabilité a posteriori de phrases qui est souvent plus haute que celle de documents. Cela nous confirme que ses documents sont bien distingués au niveau des phrases.

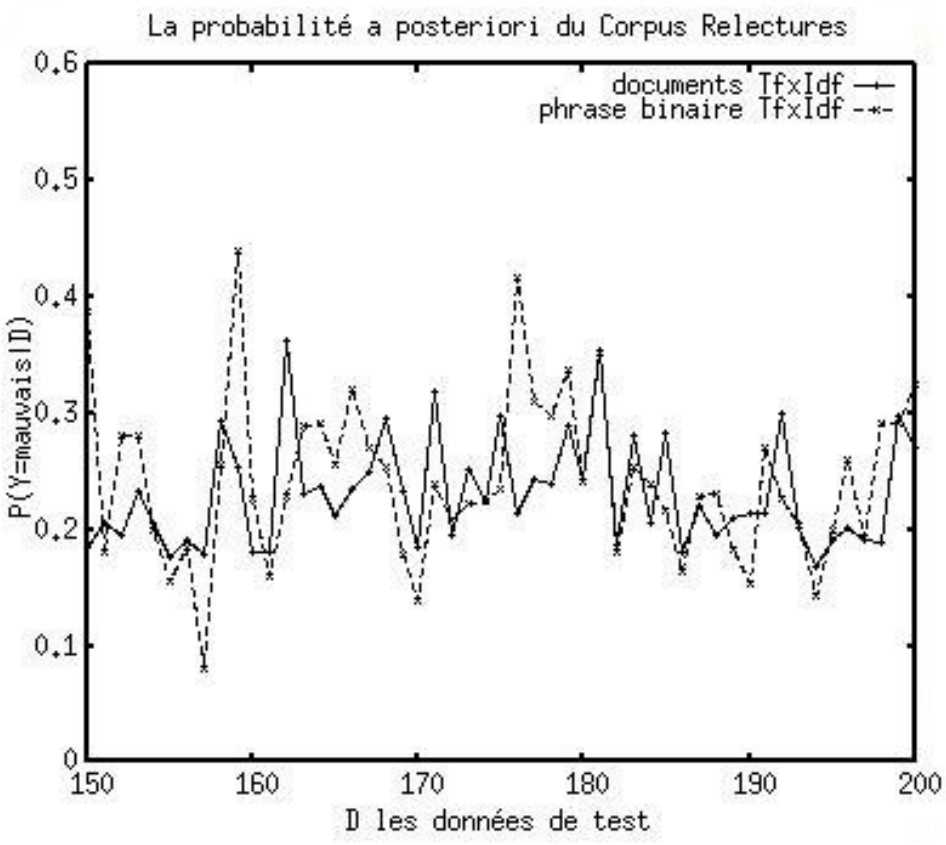


Figure 3 – La probabilité a posteriori du corpus Relectures d'articles

4.2 La réduction de mots dans le sac de mots

Comme nous l'avons précisé dans la section 2.2, nous adoptons pour méthode d'éliminer les mots les plus utilisés de la langue française afin de réduire le nombre de mots à analyser. Il y a 13 mots à éliminer lors des analyses comme suit :

la:75384:15577,*la*:111332:16126,*les*:72674:14753,*un*:36345:12358,

une:34421:11765,*l*:88199:15754,*et*:67158:14702,*Une*:1192:1059,

Un:1144:1021,*Les*:6619:4272,*L*:7312:5122,*La*:9299:5566,*Le*:9534:5851,

Avec mot :nombre_total_de_ce_mot : nombre_de_document_contient_ce_mot dans le corpus débats parlementaires

Ces mots sont éliminés des sacs de mots et naturellement n'apparaissent pas dans les vecteurs de documents ou de phrases. En réalité, l'ensemble des mots les plus utilisés peut très bien augmenter en y ajoutant les prépositions « à » ou « de », pourtant ces prépositions jouent un rôle important dans les expressions telles que « à travers » ou « par rapport à ». Nous constatons le même phénomène dans les cas de « ne .. que » (l'expression de l'unique) ou « ne .. pas » (le négatif), etc. Nous avons choisi d'éliminer les 13 mots ci-dessus pour la raison qu'ils sont plutôt souvent liés au genre du mot (masculin/féminin), donc au mécanisme du vocabulaire, qu'au sens apporté à la phrase. Ainsi l'information de la phrase est conservée le mieux possible. Une autre raison est que, si on élimine un grand nombre de mots, on risque de se retrouver devant une phrase, ou même un document « vide » car aucun mot n'est retenu, comme dans le cas du corpus de relecture d'articles.

Diviser le corpus d'apprentissage 60% et de test 40%	
Corpus d'apprentissage	(A)ccuracy
Critiques de cinéma, spectacles, livres, BD et CD	0,60096
Test de jeux vidéo	0,70767
Relectures d'articles de conférences	0,42372
Débats parlementaires	0,70697

Tableau 8 Les résultats des corpus d'apprentissage avec la réduction de mots

Le nombre de mots ne diminue pas d'une façon brutale quand on compare ce résultat au celui du sac de mots complet (voir le tableau 6). Notre but est de trouver une méthode pour augmenter la valeur d'exactitude (A), mais cette méthode la diminue dans l'ensemble des corpus. C'est pour cette raison que nous ne soumettons pas cette méthode mais la remplaçons par l'analyse des documents en les divisant en phrases comme expliqué ci-dessus.

5 Conclusions

La méthode que nous avons présentée ici repose sur la saisie des informations probabilistes à partir du modèle d'apprentissage discriminant. Nous avons besoin de nous appuyer sur deux principes, le premier est qu'il n'est pas nécessaire de calculer, à chaque itération dans l'algorithme d'apprentissage MVS, une estimation probabiliste. Le deuxième est qu'une étape supplémentaire est réalisée pour rétablir

cette estimation après l'avoir apprise, enfin, nous obtenons les informations désirées.

Corpus de test	(P)récision	(R)appel	(F)score
Critiques de cinéma, spectacles, livres, BD et CD	0,5276 ± 0,0982	0,4829 ± 0,0683	0,5004 ± 0,0668
Test de jeux vidéo	0,6925 ± 0,0996	0,6367 ± 0,0921	0,6604 ± 0,0864
Relectures d'articles de conférences	0,4804 ± 0,0490	0,4614 ± 0,047	0,4706 ± 0,0468
Débats parlementaires	0,6545 ± 0,0564	0,6298 ± 0,0645	0,6416 ± 0,0594

Tableau 9 Les résultats des tests (moyenne, écart-type) de toutes les équipes

En générale, la méthode a donné des résultats supérieurs à la moyenne de l'ensemble des participants de DEFT07 sur les deux corpus de Débats et Critiques. Cependant ce résultat reste inférieur à la moyenne sur les deux autres. La difficulté se situe dans l'augmentation du nombre de phrases lorsque nous divisons chaque document en phrases individuelles en représentant localement le document. Nous avons passé environ deux heures à simplement créer les deux corpus d'apprentissage et de test pour chacun des deux corpus « Test de jeux vidéo » et « Débats parlementaires ». Notre principe de calcul des valeurs est bien confirmé par l'usage, en plus, nous voulons rappeler un inconvénient de cette méthode, il faut résoudre le problème (8) à chaque donnée d'entrée (voir la sélection 3.1.3.b). Le problème (8) semble correspondre à la vraisemblance globale après la discrimination locale qui a été créée par la MVS. Nous désirons appliquer ces résultats aux modèles d'apprentissage probabiliste et espérons que les résultats vont s'améliorer.

Remerciements

Nous remercions sincèrement le comité de DEFT07 pour leur effort d'organisation, de collecte des données ainsi que pour les explications qui nous ont été fournies.

Bibliographie

Salton, G. and McGill, M. J. (1983). Introduction to modern information retrieval. McGraw-Hill, ISBN 0070544840.

Salton, G. and Buckley, C. (1988).Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 24(5): 513–523.

F. Vallet, P. Réfrégier, J. G. Cailton (1991). Linear discrimination: explicit and iterative solutions. In Pattern recognition and neural networks Vol. 2, Pages: 91 – 114

Kenneth W. Church and William A. Gale (1995). Inverse Document Frequency (IDF): A Measure of Deviation from Poisson. In Proceedings of the Third Workshop on Very Large Corpora,pp 121--130.

Corinna Cortes and V. Vapnik,(1995). Support-Vector Networks. Journal of Machine Learning,vol. 20.

William W. Cohen. (1995) Learning to classify English text with ILP methods. In Luc De Raedt, editor, Advances in ILP. IOS Press.

D. Price, S. Knerr, L. Personnaz, and G. Dreyfus (1995). Pairwise neural network classificateurs with probabilistic outputs. In Neural Information Processing Systems, volume 7, pages 1109-1116. The MIT Press.

T. Hastie and R. Tibshirani, (1996). Classication by pairwise coupling.Technical report, Stanford University and University of Toronto.

N. Friedman and M. Goldszmidt (1996). Building classificateurs using Bayesian networks. In AAAI '96.

Vapnik, V. (1998). Statistical Learning Theory. Wiley-Interscience, New York.

J. Platt (1998). Fast training of support vector machines using sequential minimal optimization. In B. Scholkopf, C. Burges, and A. Smola, editors, Advances in kernel methods - support vector learning. MIT Press.

J. Platt, (2000).Probabilistic outputs for support vector machines and comparison to regularized likelihood methods, in *Advances in Large Margin Classificateurs*, A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press.

Müller, K.-R.; Mika, S.; Rätsch, G.; Tsuda, K.; Schoelkopf, B. (2001). An Introduction to Kernel-Based Learning Algorithms. IEEE transactions on neural networks 12, Nr.2, pp.181-201/ISSN 1045-9227

D. Tax and R. Duin (2002). *Using two-class classificateurs for multi-class classification*. In International Conference on Pattern Recognition, Quebec City, QC, Canada, August.

T. Joachims (2002)**.** Learning to Classify Text using Support Vector Machines, Kluwer Academic Publishers, May 2002,ISBN 0-7923-7679-X.

Sarah Zelikovitz and Haym Hirsh (2002). Integrating Background Knowledge into Nearest-Neighbor Text Classification.*Proceedings of the 6th European Conference on Case Based Reasoning*. Springer Verlag.

H.-T. Lin, C.-J. Lin, and R. C. Weng (2003). A note on Platt's probabilistic outputs for support vector machines. Technical report, Department of Computer Science, National Taiwan University.

T.-F. Wu, C.-J. Lin, and R. C. Weng (2004). Probability estimates for multi-class classification by pairwise coupling. Journal of Machine Learning 5:975-1005.

Summary

The text classification is simply represented by an estimation of the posterior probability $P(Y/D)$. We shall present a method based on the SVM to realize this task. There are two analyze approaches: the first represent each document as unit, the second tries to divide document into sentences. We have also described a relationship between the local discrimination and the global likelihood.

Keywords: SVM, Probability Estimation, Text Classification, Multi-class classification.

Approches hybrides, vocabulaire d’opinion

Actes du troisième défi fouille de texte, DEFT2007, Grenoble, France, 3 juillet 2007.
Proceedings of the Third DEFT Workshop, DEFT2007, Grenoble, France, 3rd July 2007.
Pages 95-104

Défi: Classification de textes français subjectifs

Michel Généreux et Marina Santini

Natural Language Technology Group
University of Brighton, United Kingdom
{M.Genereux,M.Santini}@brighton.ac.uk

Résumé : Dans cet article, nous présentons le résultat de la classification de textes selon des critères subjectifs. La méthode proposée n'est pas nouvelle en soi, mais elle présente une brochette de traits et méthodes de normalisation des vecteurs de traits qui eux constituent une approche originale. Après une phase de réglage durant laquelle nous mettons au point une combinaison de traits et méthodes de normalisation susceptible de fournir les meilleurs résultats, nous soumettons les corpus de test à notre système. Les résultats obtenus, bien que modestes, nous permettent de tirer des conclusions intéressantes sur la validité et l'utilité d'une telle approche.

Mots-clés : Classification, Subjectivité, Traits, Normalisation

1 Introduction

La tâche demandée aux participants de DEFT 2007 était de classifier des textes selon qu'ils ont un argumentaire plutôt *positif, négatif* ou *neutre*. Les textes proviennent de divers domaines : des critiques de de films, livres, spectacles et bandes dessinées (corpus 1), des tests de jeux vidéo (corpus 2), des relectures d'articles de conférences (corpus 3) et des débats parlementaires (corpus 4). Les méthodes permises sont libres (supervisées, non supervisées), mais les ressources mises en oeuvre doivent se limiter aux corpus d'entraînement fournis par le comité DEFT 2007. Cet article décrit les techniques que nous avons utilisées pour la classification. L'article est organisé comme suit : la section 2 présente la méthode d'apprentissage automatique, les traits, le dictionnaire bilingue et les méthodes de normalisation utilisés. La section 3 est dédiée à la phase de mise au point alors que la section 4 présente les résultats de la tâche proprement dite. Nous discutons et concluons l'article aux sections 5 et 6 respectivement.

2 Méthodologie

2.1 Méthode d'apprentissage automatique

Nous avons arrêté notre choix sur la méthode d'apprentissage automatique dites *Support Vector Machine (SVM)*. Cette méthode fût utilisée pour la première fois dans la classification de texte par (Joachims, 1997). Elle a fait ses preuves dans la classification de documents d'opinion, incluant le style (Diederich *et al.*, 2000), et elle a le grand avantage de pouvoir prendre en compte une grande quantité de traits, caractéristique essentielle en ce qui concerne notre approche. Durant la phase d'entraînement, l'algorithme construit un hyperplan qui sépare de façon maximale les exemples positifs et négatifs. La classification de nouveaux exemples consiste à trouver de quel côté du plan cet exemple se trouve. Cette méthode peut être adaptée pour plus de 2 classes. Le logiciel Weka[1], disponible gratuitement, fût utilisé.

2.2 Traits

On peut diviser notre répertoire de traits en 3 groupes : catégories grammaticales (*Adjectifs, Noms, Verbes* et *Adverbes*), facettes linguistiques fonctionnelles (*Facettes*) et groupes de termes à connotation émotive (*WordNet-Affect* et *Big-Six*) :

[1]Le logiciel est disponible gratuitement à `http://www.cs.waikato.ac.nz/ml/weka/`. La méthode utilisée fût SMO avec les paramètres suivants : -C 1.0 -E 1.0 -G 0.01 -A 250007 -L 0.0010 -P 1.0E-12 -N 0 -M -V -1 -W 1

Groupe 1-Adjectifs, Noms , Verbes et Adverbes Ces catégories grammaticales ont la capacité d'exprimer une émotion ou un jugement subjectif (Turney, 2002).

Groupe 2-Facettes linguistiques fonctionnelles Dans la classification de documents selon leur *genre* (Santini, 2007), ces facettes ont donné de bons résultats. Elles sont données en Annexe A.

Groupe 3-Termes à connotation émotive Ces termes ont été classifiés par d'autres chercheurs comme ayant une composante émotive particulière. WordNet-Affect (Strapparava & Valitutti, 2004) est une extension affective de WordNet[2]. Les termes sont divisés en *positifs*, *négatifs* et *neutres*[3]. Le groupe de *Big-Six* (Ekman, 1972) se base sur des études en psychologie et réorganise WordNet-Affect selon les six émotions de base suivantes : *colère, joie, tristesse, dégoût, peur* et *surprise*. Un extrait de ce groupe est donné en Annexe B.

Chaque terme appartenant à un des trois groupes et qui se qualifie comme trait se voit assigné une catégorie grammaticale à l'aide de Tree-Tagger[4]. Pour éviter de compter les négations (e.g. «Ce n'est pas un bon film.»), nous avons éviter de comptabiliser tout terme (ici «bon») entre la particule négative *ne* et le délimiteur de fin de phrase.

2.3 Dictionnaire bilingue

En raison du manque de ressources en français pour certains calculs de normalisation nécessitant l'accès à un corpus ou lexique en anglais, certaines manipulations ont nécessité la création d'un dictionnaire bilingue anglais-français. Ce dictionnaire se compose de 1244 termes traduits manuellement et provenant des groupes de traits 2 et 3. Ce dictionnaire est nécessaire pour le calcul des facteurs de normalisation **pmi**, **sim**, **senti** et **vrai**, décrits dans la section suivante.

2.4 Méthodes de normalisation du nombre de traits

Chaque vecteur représentant un document attribue une valeur numérique à chacun des traits. La méthode de comptage la plus simple est dite *binaire*, où seulement la présence (valeur 1) ou l'absence (valeur 0) est prise en compte. Une autre façon simple de comptage est la *fréquence*, où le nombre d'apparitions du trait dans le document est directement pris en compte, souvent normalisé à une longueur de document fixe (dans notre cas, 1000 mots). Nous avons considéré dans nos expériences d'autres façons de normaliser la *fréquence*, en multipliant celle-ci par l'un ou plusieurs des facteurs suivants :

idf De l'anglais *Inverse Document Frequency*. Permet d'évaluer l'importance d'un terme *i*, la supposition étant que l'importance d'un terme diminue à mesure qu'il apparaît dans une proportion grandissante de documents faisant partie du corpus. La formule exacte est :

$$idf_i = log\frac{D}{d_i}$$

où D est le nombre total de documents et d_i est le nombre de documents dans lequel le terme *i* apparaît.

so-pmi-ir De l'anglais *Semantic Orientation - Pointwise Mutual Information - Information Retrieval*. Cette stratégie permet de calculer l'orientation sémantique (SO) de termes (textes) en calculant leur degré d'association (A) avec une liste de mots positifs et négatifs (P et N). Elle fût utilisée par (Turney, 2002) pour classifier des termes selon leur niveau de *sentimentalité*, qui peut être plus plus ou moins négative ou positive. Cette mesure, appelée SO-A, peut s'exprimer mathématiquement de la façon suivante :

$$\sum_{p}^{P} A(terme, p) - \sum_{n}^{N} A(terme, n)$$

Notons que la quantité de termes P doit être égale à la quantité de N. Pour calculer SO-A, (Turney, 2002) a recourt à la notion de PMI-IR. PMI (Church & Hanks, 1989) entre deux termes est définie comme :

$$\log_2 \frac{prob(terme_1\ est\ autour\ de\ terme_2)}{prob(terme_1) * prob(terme_2)}$$

[2] `http://wordnet.princeton.edu/`

[3] Il y a aussi une liste de termes *ambigus*.

[4] `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`

PMI est positif quand deux termes ont tendance à apparaître ensemble et négatif quand ils sont en distribution complémentaire. PMI-IR est indicatif du fait qu'en IR, les occurrences multiples d'un terme dans un même document ne compte que pour une seule occurrence ; selon (Turney, 2002), cela semble fournir une meilleure mesure de SO, plus résistante au bruit. En calculant les probabilités à l'aide du nombre de documents (nd) extraites tel que fournie en IR, cela nous donne, pour PMI-IR :

$$\log_n \frac{D * (nd(terme_1\ AUTOUR\ terme_2) + 1/D)}{(nd(terme_1) + 1) * (nd(terme_2) + 1)}$$

où D est le nombre total de documents dans le corpus. Les termes de références positifs P employés furent l'équivalent anglais de : *bon, gentil, excellent, positif, chanceux, correcte, supérieur* et les termes de références négatifs N *mauvais, méchant, pauvre, négatif, malchanceux, fautif, inférieur*. Les valeurs d'aplanissement (*smoothing*) (1/D et 1) sont choisies pour que PMI-IR soit zéro pour les termes qui ne sont pas dans le corpus, un terme est considéré comme étant *AUTOUR* d'un autre terme s'il est à l'intérieur d'une fenêtre de 20 mots et $\log_2$ a été remplacé par $\log_n$, puisque le logarithme naturel est plus commun dans la littérature et que cela ne fait aucune différence pour l'algorithme. Nous avons utilisé le corpus de Waterloo[5] contenant approximativement 46 millions de pages (documents). Chaque terme n'apparaissant pas dans le dictionnaire s'est vu attribué une valeur so-pmi-ir neutre (0).

so-sim Cette fois nous utilisons une mesure de similarité entre deux termes obtenue à l'aide de WordNet pour calculer SO-A. Cette approche est similaire à (Kamps & Marx, 2002), où la similarité est calculée en utilisant simplement un comptage des arcs séparant deux termes dans WordNet, une technique semblable au calcul effectué lorsqu'on veut connaître la relation génétique entre deux personnes à travers leurs ancêtres communs (Budanitsky & Hirst, 2001). Seuls les noms, verbes, adjectifs et adverbes peuvent avoir une affinité sémantique dans WordNet. Les termes de références positifs P employés furent l'équivalent anglais de : *bon-bonifier, gain-gagner, excellence-exceller, supériorité-surpasser* et les termes de références négatifs N *mauvais-empirer, perte-perdre, pauvreté-appauvrir, négation-nier*. Chaque terme n'apparaissant pas dans le dictionnaire s'est vu attribué une valeur **so-sim** nulle. Le «package» Perl WordNet : :Similarity[6] fût utilisé pour le calcul.

sen (Esuli & Sebastiani, 2006) fournit une ressource de valeur appelée *SentiWordNet* dans laquelle chaque *synset s* est associé à trois valeurs numériques décrivant le degré d'objectivité et de subjectivité (positif et négatif). La somme des trois valeurs doit être 1, ce qui veut dire que chaque terme peut posséder, à des degrés divers, plus d'une propriétés en même temps. Une mesure unique de subjectivité peut donc être obtenue pour chaque terme faisant partie de SentiWordNet. La méthode utilisée pour le développement de SentiWordNet est basée sur l'analyse quantitative des commentaires associés aux *synsets* en entraînant un ensemble de classeurs pour 3 classes (positif, négatif et objectif) (Esuli & Sebastiani, 2005). La valeur attribuée à chaque classe correspond à la proportion de classeurs qui ont choisi cette classe en particulier. SentiWordnet a été évalué favorablement à l'aide du *General Inquirer* (Stone *et al.*, 1966). Un extrait de SentiWordnet est fournie en annexe C. Chaque terme n'apparaissant pas dans le dictionnaire s'est vu attribué une valeur **sen** nulle.

hum Une liste de termes annotés manuellement comme étant soit positif (+1), soit négatif (-1) par (Turney, 2002). Un extrait de cette liste est fournie en annexe D. Chaque terme n'apparaissant pas dans le dictionnaire s'est vu attribué une valeur **hum** nulle.

binf Normalisation hybride, elle permet de faire une distinction entre le groupe de trait 1 (normalisé de façon *binaire*) et les groupes de traits 2 et 3 (normalisés selon la *fréquence*).

3 Mise au point

Dans la phase de mise au point, nous avons tenté d'établir une combinaison de traits et de méthodes de normalisation qui soit le mieux adapté pour l'ensemble des tâches de classification. L'ensemble du corpus d'entraînement de *Relectures* nous a servi d'étalon (227 textes classés *0*, 278 classés *1* et 376 classés *2*). Nous avons conservé les 500 traits les plus fréquents en ce qui concernent les traits de catégories grammaticales (adjectifs, noms, verbes et adverbes). Nous avons établi 18 combinaisons arbitraires de traits et

[5] `http://canola1.uwaterloo.ca/`

[6] `http://www.d.umn.edu/~tpedersen/similarity.html`

de méthodes de normalisation. Ces combinaisons se nomment *1*, *2*, *3*, *4*, *5*, *6*, *7*, *A*, *B*, *C*, *D*, *E*, *F*, *G*, *H*, *I*, *J* et *K*. Le tableau 1 illustre ces différentes combinaisons. Par exemple, la combinaison *J* est formées des 500 adjectifs et adverbes les plus fréquents tels que comptabilisés dans le corpus d'entraînement, ainsi que la fréquence totale des trois catégories de WordNet-Affect. Nous avons soumis les 18 combinaisons

Traits/Normalisation	binaire	fréquence	idf	so-pmi-ir	so-sim	sen	vrai	binf
Adjectifs	1ABHI	2 FG	3G	4	5	6	7	J
Noms	1 B	2	3	4	5	6	7	
Verbes	1 B I	2	3	4	5	6	7	
Facettes	1	2C	3	4	5	6	7	
WordNet-Affect	1	2 D K	3	4	5	6	7	J
Big-Six	1	2 E	3	4	5	6	7	
Adverbes	HI							J

TAB. 1 – Mise au point : 18 combinaisons de traits et méthodes de normalisation

à l'algorithme de classification pour le classement des textes de *Relectures*, chaque fois en utilisant une validation croisée (10-fois). Les résultats peuvent être visualisés en ordre décroissant d'exactitude à l'aide du tableau 2. Puisque DEFT 2007 permet de soumettre jusqu'à 3 exécutions différentes, nous avons donc

Combinaison	H	J	A	I	1	F	3	2	B
Exactitude	50.6	50.3	49.4	47.2	45.6	45.3	44.7	44.4	43.7
Combinaison	E	G	4	D	6	5	K	7	C
Exactitude	42.9	42.7	42.7	42.7	42.2	42.2	42.1	41.7	41.4

TAB. 2 – Mise au point : classification des *Relectures* avec différentes combinaisons de traits et méthodes de normalisation

choisi d'utiliser les combinaisons H, J et A pour la classification des corpus de test. Pour plus de clarté, nous les répétons ici :

Expérience 1 - Combinaison H Traits : adjectifs et adverbes.
Normalisation : binaire.

Expérience 2 - Combinaison J Traits : adjectifs, adverbes et Wordnet-Affect.
Normalisation : binaire pour adjectifs et adverbes, fréquence pour Wordnet-Affect.

Expérience 3 - Combinaison A Traits : adjectifs.
Normalisation : binaire.

4 Expériences

Les tableaux 3, 4, 5 et 6 détaillent les résultats de 3 expériences (H, J et A) sur les 4 corpus. Pour des raisons de rapidité de traitement, nous avons fait les choix suivants : à l'exception du corpus 3 (500 traits), le nombre maximum de traits utilisés fût 100, alors que le corpus 4 s'est vu amputé de 80% (le nombre exacte de texte d'entraînement est indiqué entre parenthèses). Chaque tableau est divisé de telle sorte que les validations croisées de la phase d'entraînement sont d'abord présentés, suivis par les résultats sur les fichiers de test. Chaque ligne de la matrice de confusion indique la distribution des textes parmi les classes. Par exemple, dans le tableau 3, combinaison H, des 309 textes étiquetés *zéro*, seulement 76 ont été classés correctement.

5 Discussion

À l'exception d'un sous-groupe particulier du groupe 3 (WordNet-Affect), nos résultats de la phase de mise au point montrent que, dans le cadre d'une approche supervisée avec SVM, la meilleure façon d'obtenir des taux d'exactitude raisonnable est de s'en tenir aux traits familiers (adjectifs, adverbes) avec normalisation binaire. En soi ce résultat est intéressant, quoiqu'un peu surprenant dans le cas de traits du

Combi-naison	Validation-Croisée 3-fois				Matrice de confusion			
	Exactitude	Précision	Rappel	F-score	zéro (309)	un (615)	deux (1150)	Classe
		0.425	0.246	0.311	76	61	172	zéro
H	56.2	0.415	0.270	0.327	50	166	399	un
		0.618	0.803	0.699	53	173	924	deux
		0.392	0.236	0.295	73	64	172	zéro
J	55.4	0.399	0.259	0.314	55	159	401	un
		0.615	0.797	0.695	58	175	917	deux
		0.382	0.220	0.279	68	34	207	zéro
A	55.7	0.392	0.115	0.178	53	71	491	un
		0.593	0.884	0.710	57	76	1017	deux
Moyenne	55.8	0.470	0.429	0.423	Fin de l'entraînement			
H avec données de test		0.48	0.43	0.45	Tel que communiqué par DEFT 07			
J avec données de test		0.49	0.44	0.46	Tel que communiqué par DEFT 07			
A avec données de test		0.50	0.39	0.44	Tel que communiqué par DEFT 07			
Moyenne Test		0.49	0.42	0.45				
Tous les participants		0.53	0.48	0.50	Tel que communiqué par DEFT 07			
		$\pm$0.10	$\pm$0.07	$\pm$0.07				

TAB. 3 – Corpus 1 (Critiques de films, livres, spectacles et bandes dessinées)

Combi-naison	Validation-Croisée 3-fois				Matrice de confusion			
	Exactitude	Précision	Rappel	F-score	zéro (497)	un (1166)	deux (874)	Classe
		0.610	0.453	0.520	225	216	56	zéro
H	64.6	0.639	0.703	0.699	113	820	233	un
		0.673	0.681	0.677	31	248	595	deux
		0.568	0.435	0.493	216	224	57	zéro
J	62.1	0.616	0.679	0.646	127	792	247	un
		0.651	0.649	0.650	37	270	567	deux
		0.551	0.467	0.505	232	231	34	zéro
A	63.2	0.617	0.669	0.642	160	780	226	un
		0.695	0.677	0.686	29	253	592	deux
Moyenne	63.3	0.624	0.604	0.613	Fin de l'entraînement			
H avec données de test		0.64	0.60	0.62	Tel que communiqué par DEFT 07			
J avec données de test		0.64	0.61	0.63	Tel que communiqué par DEFT 07			
A avec données de test		0.61	0.59	0.60	Tel que communiqué par DEFT 07			
Moyenne Test		0.63	0.60	0.62				
Tous les participants		0.69	0.64	0.66	Tel que communiqué par DEFT 07			
		$\pm$0.10	$\pm$0.09	$\pm$0.09				

TAB. 4 – Corpus 2 (Tests de jeux vidéo)

type *Big-Six*, où l'on aurait pu s'attendre à mieux. Dans le cas des méthodes de normalisation, force est d'admettre que dans certain cas, la faible dimension de notre dictionnaire bilingue n'a probablement pas permis à ces facteurs de jouer un rôle déterminant. Notons aussi que l'utilisation de bigrammes comme traits constitue une voie de recherche intéressante. À ce stade, l'approche qui semble la plus prometteuse compte adjectifs et adverbes de façon binaire et un groupe de termes (WordNet-Affect) selon leur fréquence globale.

D'autre part, nous avons essayé un type de traits jamais utilisé auparavant dans la classification de textes d'opinion, les facettes linguistiques. Ces facettes, telles que présentées dans (Santini, 2007), sont des macro-traits qui peuvent être *interprétés fonctionnellement*. Par exemple, la facette *première personne* inclut les pronoms personnels singuliers et pluriels. Cette facette indique que le contexte de communication est relié à celui qui produit le texte. Une fréquence élevée de cette facette dans un texte signale une position impressionniste ou subjective. Alors que dans la plupart des tâches de classification de textes les traits sont utilisés individuellement sans plus d'interprétation, avec les facettes le but est d'interpréter une vue particulière

Combi-	Validation-Croisée 3-fois				Matrice de confusion			
naison	Exactitude	Précision	Rappel	F-score	zéro (227)	un (278)	deux (376)	Classe
		0.435	0.445	0.440	101	70	56	zéro
H	50.2	0.439	0.482	0.460	63	134	81	un
		0.602	0.551	0.575	68	101	207	deux
		0.441	0.445	0.443	101	72	54	zéro
J	50.1	0.436	0.482	0.458	59	134	85	un
		0.597	0.548	0.571	69	101	206	deux
		0.427	0.436	0.431	99	76	52	zéro
A	47.9	0.397	0.442	0.418	68	123	87	un
		0.590	0.532	0.559	65	111	200	deux
Moyenne	49.4	0.475	0.485	0.484	Fin de l'entraînement			
H avec données de test		0.47	0.47	0.47	Tel que communiqué par DEFT 07			
J avec données de test		0.46	0.46	0.46	Tel que communiqué par DEFT 07			
A avec données de test		0.43	0.44	0.43	Tel que communiqué par DEFT 07			
Moyenne Test		0.45	0.46	0.45				
Tous les participants		0.48	0.46	0.47	Tel que communiqué par DEFT 07			
		± 0.05	± 0.05	± 0.05				

TAB. 5 – Corpus 3 (Relectures d'articles de conférences)

Combi-	Validation-Croisée 3-fois				Matrice de confusion		
naison	Exactitude	Précision	Rappel	F-score	zéro (2080)	un (1380)	Classe
H	62.7	0.643	0.852	0.733	1773	307	zéro
		0.564	0.288	0.381	983	397	un
J	63.4	0.650	0.849	0.736	1766	314	zéro
		0.576	0.309	0.403	953	427	un
A	64.6	0.651	0.886	0.751	1843	237	zéro
		0.623	0.284	0.390	988	392	un
Moyenne	63.6	0.618	0.578	0.566	Fin de l'entraînement		
H avec données de test		0.56	0.52	0.54	Tel que communiqué par DEFT 07		
J avec données de test		0.57	0.54	0.55	Tel que communiqué par DEFT 07		
A avec données de test		0.60	0.54	0.57	Tel que communiqué par DEFT 07		
Moyenne Test		0.58	0.53	0.55			
Tous les participants		0.65	0.63	0.64	Tel que communiqué par DEFT 07		
		± 0.06	± 0.06	± 0.06			

TAB. 6 – Corpus 4 (Débats parlementaires)

dans la communication. Par exemple, il est pris pour acquis qu'une fréquence élevée de la facette *première personne* indique qu'on est en présence d'un texte ARGUMENTAIRE comme des COMMENTAIRES ou des OPINIONS. (Santini, 2007) s'est servit de 100 facettes, divisées en plusieurs sous-types (e.g. fonctionnelles, syntagmatiques ou HTML). Pour DEFT nous n'avons utilisé que 14 facettes. Quoique plusieurs des 100 facettes utilisées dans (Santini, 2007) étaient de natures grammaticales basées principalement sur la sortie d'un parseur-étiqueteur (Tapanainen & Järvinen, 1997) pour l'anglais, dans ce défi nous avons sélectionné un petit sous-ensemble (principalement lexical) exploratoire : les facettes sont montrées en annexe A. La classification sémantique de verbes[7] en sept catégories est prise de (Biber *et al.*, 1999).

L'utilisation de facettes introduit deux innovations pour la classification de textes argumentaires. La première est reliée à la nature grammaticale des facettes. Alors que la plupart des recherches portant sur des textes affectifs (e.g. analyse de sentiments, classification d'opinion) sont basées sur des termes ayant une connotation affective (Hatzivassiloglou & Wiebe, 2000; Riloff & Wiebe, 2003), en l'occurrence des adjectifs et adverbes, les facettes mettent l'accent sur l'utilisation de signaux grammaticaux. Nous avons noté une variation dans l'utilisation de pronoms personnels à travers les différents types de textes. Par exemple, les textes de jeux vidéo réfèrent souvent directement aux joueurs en utilisant un pronom personnel de la

[7]Traduits en français pour ce défi.

deuxième personne, comme dans la phrase «Vous incarnez un guerrier [...]», alors que les débats parlementaires font souvent l'utilisation de pronoms personnels de la première personne qui mettent l'emphase sur la vue exprimées par l'auteur, comme dans «Nous avons passé des dizaines d'heures en juillet et en août [...] à analyser ce projet. Mon sentiment est qu'il répond bel et bien à l'évolution du monde [...]». De façon similaire, les verbes d'activité apparaissent plus souvent dans les textes sur les jeux vidéo, alors que les relectures d'articles se caractérisent plus par des verbes mentaux et de communication. Nous avons aussi fait l'hypothèse que les fréquences attribuées aux nominaux et prédicats (qui habituellement aident à faire la distinction entre l'écrit et le parlé) peuvent varier à travers les différents types de textes.

La deuxième innovation réfère à la nature composée des facettes ; en d'autres mots, les facettes sont des macro-traits, c'est-à-dire que chaque facette est composée d'un certain nombre de traits individuels qui partagent une interprétation sémantique et textuelle similaire. Nous avons défini les facettes comme des traits *interprétées fonctionnellement* parce qu'elles aident à l'interprétation et à la reconstruction du contexte de communication par l'intermédiaire de signaux linguistiques. L'utilisation de macro-traits comporte un avantage pratique. En fait, les facettes réduisent le risque d'*overfitting*, un phénomène qui apparaît habituellement quand un modèle statistique a trop d'attributs. Utilisées seules (expérience C), les facettes permettent de classifier correctement 41% des relectures, ce qui nous semble encourageant, d'autant plus si on compare avec une expérience (A, exactitude 50%) où un grand nombre (500) de traits (adjectifs) ont été utilisés (voir table 2).

6 Conclusion

Dans ce défi nous avons avons adopté une approche classique supervisée (SVM avec traits reliés aux catégories grammaticales - groupe 1) pour la classification de textes à teneur subjective, l'intention de base étant d'améliorer les performances rapportées dans la littérature (pour l'anglais) en faisant appel à d'autres types de traits, des facettes linguistiques fonctionnelles (groupe 2) et une liste de termes à connotation émotive (groupe 3), en plus de facteurs de normalisation diverses.

Pour ce qui est des résultats du défi en soi, nous nous en tirons honorablement, avec un F-score à l'intérieur de l'écart-type autour de la moyenne, sauf pour les débats parlementaires (2 classes). Nous avançons l'hypothèse que la mise au point, concoctée sur un corpus avec 3 classes (Relectures), n'est pas optimale pour la classification avec 2 classes (Débats). Les F-scores varient entre 54% et 57% pour 2 classes (Débats) et entre 43% et 63% pour 3-classes. Dans tous les cas, ces résultats sont largement au-dessus de ce que l'on pouvait s'attendre en choisissant au hasard (33% et 25%). La validation-croisée que nous avons effectuée à l'interne fût une bonne prédiction des résultats du test. Le meilleur résultat est pour les jeux vidéo (63%). La matrice de confusion pour le corpus 1 (critiques) révèle un lourd penchant en faveur des classes d'entraînement les plus nombreuses, et nous suspectons qu'un corpus de relectures d'articles de conférences (3), avec son langage plutôt neutre, présente une difficulté particulière pour ce genre de classification (d'où le faible F-score de l'ensemble des participants), surtout dans le cas de systèmes basés exclusivement sur le contenu lexical comme le nôtre. D'autre part, nous nous sommes limités à un nombre restreint de traits (les 100 les plus fréquent dans le cas des corpus 1,2 et 4, 500 dans le cas du corpus 3), ainsi que d'un nombre réduit de textes (1/5 du corpus original) pour l'entraînement de la tâche 4.

En ce qui concerne aux traits linguistiques fonctionnelles, ces premiers résultats nous permettent d'être encouragé. En les combinant avec d'autres traits, par exemple dans l'expérience 1, l'exactitude atteint environ 46%, un résultat compétitif. Le but à long terme est d'augmenter le nombre de facettes utilisé et de trouver une combinaison idéale de facettes et de traits plus traditionnels dans l'espoir d'augmenter la performance générale.

Summary

In this article, we present the results of a text classification task according to subjective criteria. The proposed method is not new as such, but it is based on a set of features and normalizing methods for the feature vectors that do constitute an original approach. After a tuning phase in which we investigate which combinations of features and normalizing methods are the best, we submit the testing data to our system. The accuracy of our results are modest, but allow us to draw interesting conclusions on the validity and utility of such an approach.

Keywords : Classification, Subjectivity, Features, Normalization

Références

BIBER, JOHANSSON, LEECH, CONRAD & FINEGAN (1999). *Longman Grammar of Spoken and Written English.* USA : Longman.

BUDANITSKY A. & HIRST G. (2001). Semantic distance in wordnet : an experimental, application-oriented evaluation of five measures. In *NAACL 2001 Workshop on WordNet and Other Lexical Resources*, Pittsburgh.

CHURCH K. W. & HANKS P. (1989). Word association norms, mutual information and lexicography. In *27th annual Conf. of the ACL*, p. 76–83 : New Brunswick, NJ :ACL.

DIEDERICH J., KINDERMANN J., LEOPOLD E. & PAASS G. (2000). Authorship attribution with support vector machines.

EKMAN P. (1972). Universal and cultural differences in facial expression of emotion. In J. COLE, Ed., *Nebraska Symposium on Motivation*, p. 207–282, Lincoln : University of Nebraska Press.

ESULI A. & SEBASTIANI F. (2005). Determining the semantic orientation of terms through gloss analysis. In *Proceedings of CIKM-05, 14th ACM International Conference on Information and Knowledge Management*, Bremen, DE. Forthcoming.

ESULI A. & SEBASTIANI F. (2006). Sentiwordnet : A publicly available lexical resource for opinion mining.

HATZIVASSILOGLOU V. & WIEBE J. (2000). Effects of adjective orientation and gradability on sentence subjectivity. In *International Conference on Computational Linguistics*.

JOACHIMS T. (1997). *Text Categorization with Support Vector Machines : Learning with Many Relevant Features.* Rapport interne LS8-Report 23, Universität Dortmund. LS VIII-Report.

KAMPS J. & MARX M. (2002). Words with attitude. In *1st International Conference on Global WordNet, Mysore, India*.

RILOFF E. & WIEBE J. (2003). Learning extraction patterns for subjective expressions. In *Conference on Empirical Methods in Natural Language Processing (EMNLP-03). ACL SIGDAT.*

SANTINI M. (2007). *Automatic Identification of Genre in Web Pages*. PhD thesis, University of Brighton.

STONE P. J., DUNPHY D. C., SMITH M. S. & OGILVIE D. M. (1966). *The General Inquirer : A Computer Approach to Content Analysis*. MIT Press. MIT Press.

STRAPPARAVA C. & VALITUTTI A. (2004). Wordnet-affect : an affective extension of wordnet. In *The 4th International Conference on Language Resources and Evaluation (LREC 2004)*, p. 1083–1086, Lisbon.

TAPANAINEN P. & JÄRVINEN (1997). A non-projective dependency parser. In *The 5th Conference on Applied Natural Language Processing*, Washington, DC, USA.

TURNEY P. D. (2002). Thumbs up or thumbs down ? semantic orientation applied to unsupervised classification of reviews. In *40th Annual Meeting of the Association for Computational Linguistics, Philadelphia*.

A Quatorze Facettes Linguistiques Fonctionnelles

NOMINAL : noms, adjectifs et nombres
PRÉDICAT : verbes
PASSÉ : verbes au passé
PASSIF : verbes au passif
PREMIÈRE PERSONNE : je moi mon mien nous notre
DEUXIÈME PERSONNE : tien tu vous ton votre
TROISIÈME PERSONNE : il elle son eux lui
VERBES D'ACTIVITÉ : faire saisir aller donner prendre venir utiliser quitter montrer essayer acheter travailler déplacer suivre mettre payer amener rencontrer jouer courir tenir tourner envoyer asseoir attendre marcher transporter perdre manger surveiller atteindre ajouter produire fournir choisir porter ouvrir gagner attraper passer secouer fixer vendre dépenser appliquer former obtenir réduire arranger battre vérifier couvrir diviser rapporter étendre réparer suspendre joindre étendre tirer recevoir répéter sauver partager sourir lancer visiter accompagner acquérir avancer comporter emprunter brûler nettoyer grimper combiner controler défendre délivrer creuser affronter engager exercer élargir explorer
VERBES DE COMMUNICATION: dire raconter appeller demander écrire parler énoncer remercier décrire réclamer offrir suggérer admettre annoncer répondre argumenter renier discuter encourager expliquer exprimer insister mentionner noter proposer publier citer répliquer reporter taire signer chanter affirmer enseigner avertir accuser reconnaître addresser aviser appeler assurer défier plaindre consulter convaincre déclarer demander stresser excuser informer inviter persuader téléphoner prier promettre questionner recommander remarquer répondre spécifier jurer menacer presser accueillir chuchoter
VERBES MENTAUX: voir savoir penser trouver vouloir signifier nécessiter sentir aimer entendre souvenir croire lire considérer supposer écouter désirer demander comprendre attendre espérer supposer determiner consentir porter soucier choisir comparer decider découvrir douter apprécier examiner confronter oublier haïr identifier imaginer soucier apprendre occuper manquer noter planifier préférer prouver réaliser réaliser rappeller reconnaître regarder souffrir souhaiter projeter accepter permettre apprécier approuver évaluer blâmer soucier calculer conclure célébrer confirmer compter oser mériter détecter écarter distinguer expérimenter craindre pardonner deviner ignorer fier interpréter juger justifier observer percevoir prédire prétendre compter rappeller satisfaire résoudre étudier suspecter impressionner
VERBES CAUSATIFS: aider laisser forcer exiger affecter causer activer assurer permettre prévenir assister garantir influencer permettre
VERBE D'OCCURENCE: devenir survenir changer mourir pousser développer survenir survenir émerger tomber augmenter durer élever disparaître couler briller couler glisser
VERBES EXISTENTIELS: devoir apparaître tenir rester vivre varier sonner inclure impliquer contenir exister indiquer représenter tendre insérer importer réfléchir associer rester révéler convenir mérter concerner constituer definir illustrer impliquer manque sembler devoir posséder
VERBES ASPECTUELS: débuter garder arrêter commencer continuer compléter terminer finir cesser

B WordNet-Affect et Big-Six (extrait)

```
WordNet-Affect
--------------
POSITIF   joie rayonné  exalté allègrement réjouir euphorisant triomphe
NÉGATIF   crainte effrayé terrible alarme impatient timide atroce
NEUTRE    apathie impassibilité rêveur langoureusement indifférence

Big-Six
-------
COLÈRE  ombrage offense folie irritation enragement indignation outrage
JOIE    culte adoration chaleur triomphe unité sympathie tendresse
TRISTESSE  ennui  poids  apitoiement douleur oppression misère punition
DÉGOÛT     répugnance  horreur  nausée  maladie  révulsion  revirement
PEUR       agitation effroi cercueil timidité suspens hésitation  ombre
SURPRISE   admiration  étonnement stupeur  terreur  merveille stupidité
```

C SentiWordNet (extrait)

TERME	SCORE POSITIF	SCORE NÉGATIF	SCORE NEUTRE
abandonné	0.000	0.000	1.000
accablant	0.250	0.250	0.500
adoration	0.625	0.000	0.375
affligé	0.125	0.625	0.250
affreux	0.000	0.625	0.375
agacer	0.000	0.625	0.375
agité	0.000	0.125	0.875
agressif	0.625	0.000	0.375
alarmé	0.000	0.500	0.500
aliéné	0.625	0.000	0.375
amoureux	0.500	0.125	0.375
apathique	0.000	0.750	0.250
appréhensif	0.000	0.625	0.375
approbation	0.500	0.125	0.375
ardent	0.375	0.125	0.500
atroce	0.000	0.500	0.500
attrister	0.000	0.750	0.250

D Termes classifiés manuellement (extrait)

```
NÉGATIF
 abandon  accablant  affligé  affligeant  affreux  aggravation aggraver
 agiter  agressif  agression  agressivité  alarme  aliénation  amertume
 anéantissement  animosité  antagonisme  antipathie  anxiété  apathie
 apathique  appréhensif  appréhension atroce avare aversion belligérant
POSITIF
 acclamation  accomplisser  admiration  admirer  affection  affectueux
 agréable  ajustement  alerte allégresse amical amour apaiser apprécier
 approbation  approuver  ardeur  attachement  avide  bienfaisant
 bienveillance  bienveillant  bon  bonheur  bouillant calme captivation
```

Classification de textes d'opinions : une approche mixte n-grammes et sémantique

Matthieu Vernier[1], Yann Mathet[1], François Rioult[1], Thierry Charnois[1], Stéphane Ferrari[1] et Dominique Legallois[2]

[1] Laboratoire GREYC, Université de Caen
Matthieu.Vernier@etu.info.unicaen.fr,
{Yann.Mathet, François.Rioult, Thierry.Charnois,
Stephane.Ferrari}@info.unicaen.fr

[2] Laboratoire CRISCO, Université de Caen,
dominique.legallois@unicaen.fr

Résumé : Cet article présente la participation de l'équipe du GREYC à DEFT'07, en détaillant les différentes approches mises en place ainsi que les résultats obtenus. Plusieurs techniques ont été mises en œuvre, notamment une approche à base de n-grammes, et une chaîne de traitement linguistique de production d'indices. L'approche de type n-grammes a bénéficié de traitements linguistiques complémentaires tels que la lemmatisation et la synonymie, et constitue à elle seule un classifieur autonome. La chaîne de traitements alimente quant à elle un classifieur supervisé en lui fournissant des indices s'appuyant en particulier sur un lexique. Enfin, un autre classifieur a pour vocation de conjuguer les résultats obtenus par les deux traitements précédents.

Mots-clés : Fouille de données, classification, n-grammes, lemmatisation, synonymie, sémantique, chaîne de traitements linguistiques, lexique, classification supervisée par règles d'association.

1 Introduction

Le laboratoire GREYC présente une équipe à DEFT pour la deuxième année consécutive. Pour cette édition 2007, sa composition ainsi que les techniques mises en œuvre sont pour partie issues de l'édition précédente, et pour partie nouvelles.

Le GREYC est impliqué à la fois dans la fouille de données et dans les techniques du TAL, notamment au sein de l'équipe DODOLA qui offre un carrefour idéal à ces axes de recherche. Parallèlement, une collaboration de longue haleine existe entre ce laboratoire d'informatique et le laboratoire de linguistique CRISCO. C'est donc avec un grand intérêt qu'une petite dizaine de chercheurs de Caen ont tenté de relever le défi.

En 2006, la technique mise en place consistait en une chaîne de traitement linguistique implémentée au sein de la plate-forme LinguaStream, produisant des indices pour chaque phrase du texte traité, et alimentant un classifieur (*via* un apprentissage supervisé).

Pour le présent défi, plusieurs voies ont été abordées parallèlement, et ont parfois fonctionné en synergie :

1) Un classifieur autonome basé sur une technique de n-grammes et adapté aux spécificités du langage naturel. Ce classifieur est capable de produire les fichiers de résultats DEFT. Cette partie que nous appellerons technique « n-grammes » dans la suite de cet exposé a mobilisé deux chercheurs.
2) Une chaîne de traitements linguistiques, mise en place au sein de la plate-forme LinguaStream, et produisant pour chaque texte un certain nombre d'indices. Cette chaîne alimente un classifieur supervisé, capable lui aussi de produire des résultats. Cette partie du projet est en quelque sorte un réinvestissement des idées de l'année passée, même si, bien sûr, son contenu effectif est totalement inédit, la tâche du présent défi étant singulièrement différente du défi précédent. Cette partie, que nous nommerons désormais « chaîne LinguaStream », a mobilisé trois chercheurs.

3) Enfin, un classifieur supervisé se basant sur les résultats croisés des deux traitements précédents a vocation à capitaliser ces derniers, le but étant bien sûr d'obtenir un score supérieur à chacun des deux scores indépendants. C'est un seul chercheur qui a géré l'intégralité de cette partie cette année.

En amont de l'élaboration de ces différentes approches, nous avons initié notre travail de réflexion par une étude « manuelle » des différents corpus. Pour rendre notre analyse plus pertinente, un premier travail a consisté à procéder à un découpage automatique des corpus en classes. Par exemple, pour le corpus 1 qui dispose de trois classes (correspondant resp. aux notes de 0 à 2), nous avons produit 3 fichiers, chacun desquels contient les seuls textes associés à une classe particulière. Nous avons implémenté ce découpage en Java, au moyen d'un parseur SAX.

C'est à l'issue d'un travail de réflexion d'une quinzaine de jours sur ces différents sous-fichiers des 4 corpus que les voies de recherche ont pu être définies. Nous présentons chacune de ces dernières dans les parties suivantes de cet article, puis aborderons une analyse comparative dans une ultime partie.

2 Un classifieur à base de n-grammes

La technique des n-grammes consiste à observer les collocations contiguës sur une fenêtre de n tokens consécutifs d'un flux, et à essayer de tirer de ces observations des régularités relatives à un aspect particulier de ce flux[1]. Par exemple, certains n-grammes seront caractéristiques de tel type de corpus car très récurrents dans ce dernier, et beaucoup plus rares ailleurs. En préambule à cette partie, nous devons annoncer clairement que notre équipe n'était pas du tout familière de ce type de technique, et qui si le principe nous en a paru pertinent, il s'agit pour nous d'un premier essai. En conséquence, le contenu de cet article qui y est relatif est vraisemblablement incomplet, et probablement quelque peu naïf.

Dans l'objectif de DEFT, le flux d'entrée est un matériau linguistique (textes écrits en français), et nous essayons de catégoriser les différents textes de ce flux selon le jugement porté par leur auteur. Pour illustrer de façon très simplifiée l'hypothèse de cette approche, nous espérons trouver des n-grammes caractéristiques d'un jugement favorable, défavorable, ou enfin, le cas échéant, neutre. Par exemple, pour des articles relatifs à des critiques de livres, et après analyse automatique des corpus d'apprentissage, nous pourrions avoir des tri-grammes caractéristiques tels que :

« une vraie catastrophe » : catégorie 0

« roman assez moyen » : catégorie 1

« très belle œuvre » : catégorie 2

Ainsi, lors de l'analyse d'un texte du corpus de test, si l'on tombe sur le tri-gramme « très belle œuvre », nous serons tentés de ranger ce texte en catégorie 2. Bien sûr, il y a un risque qu'au cours de l'analyse d'un même texte, nous trouvions des n-grammes appartement à différentes catégories, rendant le choix plus difficile. L'idée que nous mettons en œuvre pour pallier cette difficulté est de deux ordres :

- Ne retenir pour chaque catégorie que les n-grammes les plus discriminants, c'est-à-dire ceux étant le moins susceptibles d'apparaître dans des textes appartenant à d'autres catégories.

- Pondérer les n-grammes, c'est-à-dire associer à chacun un poids d'autant plus important qu'il apparaît fréquemment dans sa catégorie cible relativement aux autres catégories. Puis, lors de l'analyse d'un texte, tenir autant de comptes qu'il y a de catégories, et, pour chacune des catégories, sommer les poids[2] de tous les n-grammes (de cette catégorie) trouvés dans le texte. De la sorte, nous obtenons une note globale pour chacune des catégories, que nous pouvons mettre en balance avec les notes globales obtenues pour les autres catégories.

[1] Cf. Stubbs M. & Barth I. (2003).

[2] Il s'agit bien d'une somme et non d'un produit, car il ne s'agit pas ici à proprement parler d'un calcul de probabilité, mais d'une collecte d'indices concordants. Si par exemple pour un texte à classer nous obtenons les trois indices pondérés 10, 2 et 8 pour la catégorie 0 et les deux indices pondérés 12 et 12 pour la catégorie 1, le produit donnerait 10*2*8=160 contre 12*12=144, alors que la somme donnera 10+2+8= 20 contre 12+12=24. On voit au travers de cet exemple que le produit aurait tendance à favoriser de nombreux petits indices au détriment de gros indices moins nombreux.

2.1 Apprentissage

2.1.1 Collecte des n-grammes d'un corpus pour une catégorie

Étant donné que nous avons préalablement réalisé une application permettant de découper un corpus en autant de sous corpus qu'il y a de catégories, il nous suffit à présent de réaliser un traitement effectuant la collecte des n-grammes de n'importe quel corpus, et de l'appliquer ensuite sur chacun des sous-corpus.

Ce traitement prend donc en entrée un fichier corpus XML, et produit une instance de type NGramCorpus rendant compte de tous les n-grammes présents dans le texte, et de leur fréquence relative. Il est bien sûr paramétrable quant à la longueur des n-grammes à prendre en compte (monogrammes, bigrammes, trigrammes, etc.).

Par ailleurs, une méthode statique a été mise en place dans cette classe permettant de faire un « merge » des n-grammes de deux corpus distincts, à partir de deux de ses instances. Il sera ainsi possible d'obtenir le NGramCorpus des textes appartenant aux catégories 0 et 1 à partir de ceux des textes appartenant à la catégorie 0 et de ceux de la catégorie 1. Ceci s'avèrera pratique par la suite.

2.1.2 Création d'une collection de n-grammes discriminants

Conformément aux souhaits que nous avons formulés précédemment, un traitement ultérieur a vocation à déterminer quels sont les n-grammes discriminants d'un corpus vis-à-vis d'un autre. Cette notion de vis-à-vis est très importante pour la tâche que nous avons à réaliser. En effet, trouver des n-grammes représentant un corpus (donc, appliqué ici à un sous-corpus, à une catégorie de textes) en toute généralité (c'est-à-dire par rapport à un corpus générique) serait bien moins performant que de trouver des n-grammes opposant ce corpus à un certain autre corpus.

Ce traitement, réalisé par la classe CorpusDiscriminator prend donc en entrée deux instances de NGramCorpus, et fait ressortir les n-grammes révélateurs du premier corpus par rapport au second, selon le principe suivant :

- Considérer chaque n-gramme du premier corpus.
- Pour chacun d'entre eux, regarder s'il est présent ou non dans le second corpus
- S'il est absent du second corpus, et que son nombre d'occurrences dans le premier corpus est supérieur à un certain seuil paramétrable (par exemple réglé sur 1 pour éviter les orphelins), lui attribuer le poids INFINITY
- S'il est présent dans le second corpus, lui associer un poids égal au rapport entre sa fréquence relative dans le premier corpus et sa fréquence relative dans le second corpus. Ne garder ce n-gram que si le poids ainsi calculé et supérieur à un certain seuil paramétrable.

Prenons un exemple : le trigramme « une vraie catastrophe » apparaît 12 fois dans le premier corpus, donnant lieu à une fréquence relative de 12/13247 (ce corpus comportant 13247 trigrammes), et seulement 2 fois dans le second corpus, donnant lieu à une fréquence relative de 2/17523 (ce second corpus, plus volumineux, comporte 17523 trigrammes). Ce trigramme se verra ainsi attribué un poids égal au rapport de ces deux fréquences relatives, soit (12/13247) / (2/17523), c'est-à-dire 7.93. Cela signifie que l'on a pratiquement 8 fois plus de chances de trouver ce trigramme dans un texte du premier corpus que du second. Si cette valeur est supérieure au seuil que nous avons fixé, ce trigramme sera donc conservé comme trigramme discriminant, et son poids de 7.93 lui sera associé.

Prenons un second exemple : le trigramme « très belle œuvre » apparaît 4 fois dans le premier corpus, et jamais dans le second. Si 4 est supérieur au seuil paramétrable, nous conservons ce trigramme et lui associons le poids INFINITY (on a une infinité de chances supplémentaires de trouver ce trigramme dans le premier corpus que dans le second), valeur fixée dans la pratique non pas à l'infini, ce qui interdirait la prise en compte d'autres n-grammes, mais à 15, après une série de tests.

2.1.3 Discriminer une catégorie

Dans cette approche, pour discriminer une catégorie, nous souhaitons collecter des n-grammes révélateurs de cette catégorie par rapport à **toutes les autres catégories**. Pour ce faire, dans le cas où il y a plusieurs catégories autres (cas des corpus 1 à 3 de DEFT), nous utilisons la méthode statique « merge » de toutes ces dernières.

Nous constituons donc d'une part le NGramCorpus de la catégorie à discriminer, et d'autre part le merge des NGramCorpus de chacune des autres catégories. A partir de ces deux NGramCorpus, nous obtenons donc le CorpusDiscriminator de la catégorie à discriminer. Par exemple, si nous souhaitons obtenir le CorpusDiscriminator de la catégorie 1 d'un corpus comportant 3 catégories (0, 1 et 2), nous écrivons l'instanciation suivante :

```
discriminatorCat[1] = new CorpusDiscriminator(ngramCat[1],
NGramCorpus.merge(ngramCat[0],ngramCat[2])) ;
```

Notre programme établit automatiquement le CorpusDiscriminator de chacune des catégories. Le travail d'apprentissage est à présent terminé.

2.2 Classification : choisir une catégorie

L'apprentissage étant réalisé, nous pouvons maintenant aborder la question de l'assignation d'une catégorie à un texte d'un corpus. Il s'agit simplement de parcourir le texte en question au moyen d'une fenêtre de longueur n pour un choix via des n-grammes, et pour chacun des n-grammes ainsi constitué, interroger chacun des CorpusDiscriminator de chacune des catégories. On somme alors, le cas échéant (c'est-à-dire lorsque les valeurs sont non nulles), les poids correspondants. Nous obtenons, une fois tout le texte parcouru, autant de sommes qu'il y a de catégories (3 pour les trois premiers corpus, 2 pour le dernier), qui correspondent chacune à l'indice de confiance que l'on peut accorder à la catégorie en question pour ce texte.

Nous pouvons alors assigner comme catégorie celle obtenant la somme de poids la plus élevée, ou, comme cela était aussi possible dans DEFT, proposer un indice de confiance en pourcentage à chacune des catégories, sans statuer de façon catégorique. Dans ce cas, l'indice de confiance pour une catégorie donnée est tout simplement le rapport entre le poids total de cette catégorie sur la somme des poids totaux des autres catégories.

Par ailleurs, si certains corpus se prêtent plus à un calcul via des bigrammes, d'autres des trigrammes, etc., il est fréquent qu'une combinaison de plusieurs traitements en n-grammes, par le cumul des poids de ces derniers, soit plus performante que l'application d'un seul d'entre eux. Ainsi, notre application propose le choix de la plage de n-grammes à appliquer au corpus traité. Par exemple, la plage [2, 3] signifie que l'on cumule les traitements bigrammes et trigrammes.

2.3 Les apports de la linguistique

Les premiers tests réalisés à ce stade montrent des résultats positifs, i.e. supérieurs à un tirage aléatoire, mais leur observation précise révèle parfois un manque de n-grammes discriminatoires lors du processus d'apprentissage. En d'autres termes, lorsque parmi les n-grammes du texte à catégoriser, plusieurs sont aussi présents dans le corpus d'apprentissage, nous obtenons des poids non nuls, et la catégorisation est souvent satisfaisante. Mais lorsque ceux-ci sont trop peu nombreux dans le corpus d'apprentissage, il arrive que certaines catégories obtiennent un poids nul, ou très faible, le choix se faisant alors sur une autre catégorie, souvent mauvaise. Ce phénomène est bien sûr d'autant plus manifeste que le corpus d'apprentissage est réduit.

Or les éléments présentés jusqu'ici pourraient se prêter indifféremment à différents types de flux, pour des classifications de différentes natures, pour autant que les n-grammes soient révélateurs du phénomène étudié. Pourtant, le matériau sur lequel nous nous penchons ici est de nature linguistique, ce qui lui confère un certain nombre de spécificités et de régularités dont nous pouvons tirer parti. En effet, il est fréquent que même si les formes linguistiques de surface diffèrent, les valeurs sémantiques soient pourtant très proches. Nous allons en effet améliorer notre approche en lui appliquant quelques traitements d'ordre linguistique, à la fois lors de l'apprentissage et de l'exécution.

2.3.1 Lemmatisation des corpus

L'idée de la lemmatisation est fondée sur une observation simple mais parfois très puissante : des éléments linguistiques ayant une valeur sémantique proche, mais différant quant à leur genre, leur nombre ou leur temps morphologique, verront leurs formes lemmatisées identiques.

« est un apport », « sera un apport » → « être un apport »

« la bonne idée », « les bonnes idées » → « le bon idée »

De la sorte, en lemmatisant à la fois les corpus d'apprentissage et d'exécution, nous donnons aux n-grammes une généricité permettant de les multiplier virtuellement : un n-gramme donné d'un corpus d'apprentissage aura valeur de tous les n-grammes donnant lieu à la même forme lemmatisée.

La contre partie de ce procédé est qu'en gagnant en généricité, nous créons corollairement un appauvrissement linguistique équivalent, à savoir que « la bonne idée » et « les bonnes idées » auront la même valeur (en fait, virtuellement, seront un seul et même tri-gramme), alors que le second aurait sans doute une valeur discriminante positive plus forte. Nous aurons l'occasion de discuter de ce point lors de l'analyse des résultats.

2.3.2 La synonymie

Dans le même esprit de gagner en généricité, i.e. de multiplier virtuellement la taille des corpus d'apprentissage, nous avons étudié la possibilité de tirer parti de la synonymie. En effet, de façon un peu naïve, le fait que deux mots soient synonymes fait que l'un peut se substituer à l'autre, si bien qu'à partir d'un n-gramme donné, on peut virtuellement générer nombre de n-grammes sémantiquement équivalents.

Dans les faits, plutôt que de générer profusion de n-grammes, nous allons simplement remplacer chaque mot, le cas échéant, par son représentant sémantique (choisi arbitrairement pour une classe de synonymes donnée). Nous appliquons ceci à la fois lors de l'apprentissage et de l'exécution.

Voici une illustration de l'attribution d'un représentant sémantique à une classe de synonymes :

« bon » → bon
« excellent » → bon
« formidable » → bon
« extraordiaire » → bon

On remarque que le représentant sémantique d'une classe donnée se représente lui même (cf. première ligne de l'illustration précédente).

Dans l'optique d'avoir une généricité maximale à moindres frais, nous avons tenté le recours à une ressource linguistique informatique préexistante, le dictionnaire des synonymes du Crisco (cf. Manguin, 2005). Malheureusement, les résultats se sont généralement effondrés, ou tout du moins amenuisés. La principale raison est semble-t-il le problème de la **polysémie**. Un terme donné étant (le plus souvent) polysémique, il en résulte que parmi l'ensemble de ses synonymes vont se trouver des mots dont le sens n'aura rien à voir avec celui qui nous intéresse. Par exemple pour le trigramme « apprécié ce livre », on souhaiterait élargir sa portée à des trigrammes tels que « aimé ce livre » ou « adoré ce livre ». Mais, par le recours au dictionnaire des synonymes, nous l'élargirons aussi à un trigramme tel que « évalué ce livre » (évaluer étant l'une des acceptions possibles d'apprécier), qui pour sa part ne code aucunement une appréciation positive… Le bruit ainsi généré prend le dessus sur le gain offert par la généricité.

Nous nous sommes finalement orientés vers un lexique des synonymes établi manuellement pour la tâche particulière de DEFT, et l'avons décliné en quatre versions relatives à chacun des corpus. Ce mini dictionnaire dédié des synonymes rend compte principalement des adjectifs et des verbes d'évaluation (aimer, détester, être d'accord, etc.) ainsi que des principaux objets dont il est question (œuvre, livre, film, etc.). Il a été établi de sorte à générer le moins de bruit possible, notamment en limitant son étendue aux seuls termes non (ou faiblement) polysémiques.

Nous avons profité de ce dictionnaire pour créer une classe « ponctuation » qui est le représentant de tous les signes de ponctuation d'un texte. Les signes « . », « , », « ; », etc. ont tous pour pseudo synonyme le représentant « ponctuation », si bien que des trigrammes initiaux tels que « belle œuvre , » et « belle œuvre . » seront identiques une fois passés dans le module des synonymes.

La difficulté de la polysémie mise à part, il reste comme écueil à cette piste le fait, une fois encore, que ce que l'on gagne en généricité, on le perd en spécificité. Le bon ratio est donc à trouver, qui dépend notamment de la taille du corpus d'apprentissage. Plus ce dernier est maigre, plus le recours à la généricité sera bénéfique, et vice-versa.

2.3.3 Le traitement de la négation

Enfin, nous nous sommes attaqués à la question de la négation dans nos textes dès l'origine de notre travail tant notre analyse initiale a révélé de façon flagrante combien souvent la négation pouvait inverser le sens d'une valeur sémantique locale.

Une première idée, que nous pouvons qualifier de traitement sémantique, consistait à inverser la valeur sémantique de ce qui est porté par la négation. Nous avions alors affaire à une double difficulté, la première étant de circonscrire ce sur quoi porte la négation (ce qui nécessite une analyse lexicale suffisamment robuste), la seconde d'être capable d'inverser la valeur sémantique correspondante (soit par un dictionnaire des antonymes, mais on retombe alors sur le problème de la polysémie, soit en alimentant les autres catégories, mais lesquelles et comment ?). Nous n'avons pas eu la possibilité de mettre en œuvre cette idée faute de moyens.

Une seconde idée, moins ambitieuse, consiste à éliminer des corpus toutes les parties de proposition qui sont sous le joug d'une négation. Le traitement assez basique que nous avons réalisé consiste à ne pas considérer la partie du flux comprise entre une marque de négation « ne » ou « n' » ou encore « pas », et la prochaine ponctuation, ceci à la fois dans le corpus de test que dans les corpus d'apprentissage. En supprimant de tels segments, on évacue le fait que la valeur sémantique à prendre en compte est difficile à appréhender. Cette fausse bonne idée a été un échec sur tous les corpus, pour une double raison. La première est qu'en évacuant une partie du corpus, on limite d'autant l'apprentissage (et l'exécution) : cela diminue virtuellement la taille des corpus. La seconde est qu'en fait, sans que l'on ait finalement trop à s'en préoccuper, le n-grammes disposant d'une fenêtre suffisamment large (trigrammes, quadrigrammes…) prennent d'eux même en compte la valeur sémantique de la négation : « ai pas aimé », « pas un bon roman », etc.

2.4 Application aux différents corpus et résultats

2.4.1 Présentation

Le traitement réalisé, entièrement automatique, peut être appliqué tel quel à tous les corpus. Néanmoins, nous avons procédé à des ajustements de paramètres selon les spécificités de ces derniers. Les principaux paramètres ajustables sont les suivants :

- Plage de valeurs n des n-grammes : permet de définir sur quels n-grammes appliquer le traitement (monogrammes, bigrammes, trigrammes, etc.), les poids étant cumulés lorsque la longueur de la plage de valeurs de n est supérieure à 1.
- MIN_COUNT_FOR_INFINITY : le nombre minimum nécessaire d'occurrences pour prendre en compte les n-grammes de poids « infini », c'est-à-dire n'apparaissant que dans le corpus à différencier. Fixé par exemple à 2, on ne les gardera que s'ils apparaissent au moins 2 fois.
- ELIMINATE_VALUE : idem, mais pour des n-grammes de poids non « infini », c'est-à-dire apparaissant à la fois dans le corpus à différencier et dans le corpus de comparaison.
- MIN_QUOTIENT : valeur minimale du rapport entre le nombre d'occurrences d'un n-gramme dans le corpus à évaluer et dans le corpus de comparaison. C'est donc en fait, par construction, le poids minimum des n-grammes finalement retenus.
- LEMMATISATION : ON/OFF. Choix d'appliquer ou non la lemmatisation.
- SYNONYMIE : ON/OFF. Choix d'appliquer ou non le traitement de la synonymie.
- Coefficient correcteur à appliquer à chaque catégorie : permet d'ajuster le poids d'une catégorie par rapport à ce que donne le calcul des n-grammes. Par exemple, si l'on constate lors des tests que le rappel de la catégorie 0 est déficitaire par rapport au rappel des autres catégories, on pourra gonfler ce dernier en lui assignant un coefficient correcteur tel que 1.1 (pour 10% d'augmentation des poids).

Nous avons effectué l'essentiel de nos paramétrages en nous basant sur un découpage du corpus d'apprentissage en deux, les premiers 90% pour apprendre, et 10% restants pour tester. Faute de temps, nous n'avons pu procéder qu'à un second test en toute fin d'échéance, avec resp. les derniers 80% et les premiers 20%. Les résultats ont alors été sensiblement différents, malheureusement. Nous avons alors pris le parti d'ajuster les différents paramètres en fonction de ces deux jeux de tests (l'idéal aurait été de faire 10 séries de tests en faisant tourner les découpages 90% -10%).

Mise en garde : les commentaires des sections suivantes sont basés sur les tests 90%-10%, et non sur les corpus de test et d'apprentissage finaux (ne disposant pas de la version notée du corpus de test final, il ne nous est pas possible de faire autrement). Les indications fournies ici seraient donc sans doute un peu différentes avec les corpus réels.

2.4.2 Paramétrage des 4 corpus

Corpus	Plage de n-grammes	Min-count for infinity	Eliminate value	Min quot.	Lemmatisation	Synonymie	Coefficients correcteurs
1	[1, 3]	3	1	1.5	ON	ON	Cat0 : 0.85 Cat1 : 1.35 Cat2 : 1
2	[2, 3]	3	2	1.5	ON	ON	Cat 0 : 1.15 Cat 1 : 1 Cat 2 : 1
3	[2, 3]	3	2	1.5	ON	OFF	Cat 0 : 1.08 Cat 1 : 1.07 Cat 2 : 0.78
4	[1, 3]	3	2	1.5	OFF	ON	Cat 0 : 1 Cat 1 : 1

2.4.3 Corpus 1

L'application de lemmatisation produit un bond du F-Score d'environ 10% (+0.1), ce qui est particulièrement remarquable. L'application des synonymes donne lieu quant à elle à un gain supplémentaire d'environ 0.6% (+0.006).

2.4.4 Corpus 2

L'application de lemmatisation produit un gain du F-Score d'environ 4.5% (+0.045), et celle des synonymes donne lieu quant à elle à un gain supplémentaire d'environ 1.1% (+0.011), ce qui n'est pas négligeable compte tenu du score déjà établi. C'est sur ce corpus que nos principes ont été les plus efficaces, les scores obtenus étant nettement supérieurs aux scores obtenus sur les trois autres.

2.4.5 Corpus 3

L'application de lemmatisation produit un gain du F-Score d'environ 3% (+0.03). Par contre, sur ce corpus, les synonymes donnent lieu à un baisse d'environ 1% du score, ce qui en fait un cas particulier. Notons que nos traitements se prêtent manifestement assez mal à ce corpus, sans doute du fait de sa petite taille. Par ailleurs, il est probable que notre choix de répartition 90% initiaux – 10% finaux ne se soit pas révélé opportun ici, car les résultats sur le corpus réel chutent de plusieurs points. Il aurait donc été prudent de ne pas procéder à des ajustements basés sur un nombre de textes non significatif (10% d'un petit corpus, qui plus est divisé ensuite en 3 catégories…).

2.4.6 Corpus 4

Contre toute attente, et fait unique, ce corpus donne de moins bons résultats avec le processus de lemmatisation. Cela peut provenir selon nous de deux choses :

- la première, statistique, est due à la grande taille du corpus d'apprentissage. Le nombre de n-grammes étant « naturellement » important, le fait de lemmatiser apporte relativement moins de nouveaux n-grammes virtuels. Le bruit généré par cette lemmatisation prend alors le dessus sur le maigre gain en rappel.
- La seconde est de nature linguistique, liée à la nature même du corpus manipulé : il est probable qu'ici, à la fois le temps des verbes (notamment la différence entre passé, présent et futur), ainsi et surtout que la personne (« je » versus « nous ») aient une importance prépondérante, alors même que le processus de lemmatisation les gomme.

La synonymie permet quant à elle d'obtenir un léger gain d'environ 1%.

2.4.7 Conclusion

Nous avons donc observé des différences significatives de résultats entre les corpus, mais aussi la nécessité d'adapter les valeurs des paramètres d'ajustement d'un corpus à l'autre. Il s'avère que la taille du corpus d'apprentissage et surtout que sa nature linguistique entrent en ligne de compte pour ces différents ajustements (cf. 4. Résultats et comparaison des approches).

3 Un classifieur basé sur une chaîne de traitements linguistiques

3.1 Analyse linguistique

La deuxième méthode consiste à repérer et à exploiter un certain nombre d'indices linguistiques qui marquent la présence d'une évaluation positive ou négative dans un énoncé. Elle se base notamment sur des travaux[3] qui avaient été menés sur un thème proche et qui avaient montré la faisabilité d'une telle approche[4]. L'expertise linguistique et certaines ressources ont pu être réinvesties partiellement dans le cadre de DEFT. Toutefois, ce genre d'approche linguistique aurait nécessité une expertise propre aux corpus proposés pour laquelle nous manquions de temps et de moyens compte-tenu de l'ampleur de la tâche.

Nous détaillons par la suite quels sont les types d'indices retenus et quelle est la méthode pour pondérer la valeur évaluative de ces indices. En fin d'analyse, l'objectif est d'obtenir pour chaque corpus, un ensemble de scores qui viennent alimenter un processus de classification basé sur l'extraction automatique de motifs dans une matrice.

3.1.1 Différentes catégories d'indices

Termes évaluatifs.

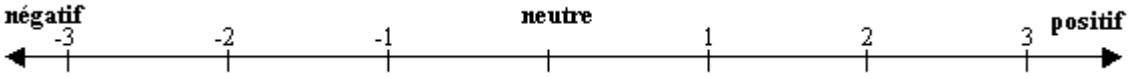
Au niveau lexical, un grand nombre de termes, quelle que soit leur catégorie grammaticale, portent une valeur évaluative intrinsèque.

	Positif	**Négatif**
Noms	*chef d'œuvre, réussite, beauté, perfection, merveille, …*	*nullité, absence, faiblesse, ersatz, navet, déception, ...*
Adjectifs	*beau, superbe, extraordinaire, intéressant, parfait, …*	*nul, incorrect, bâclé, laid, …*
Verbes	*réussir, plaire, …*	*décevoir, fruster, perdre son temps*
Adverbes	*heureusement, magnifiquement, clairement, judicieusement, …*	*malheureusement, hélas, …*

L'accumulation de ces lexies dans l'énoncé fournit autant d'indices susceptibles de mettre en lumière l'opinion de l'auteur. Toutefois, il est possible de rencontrer ces termes dans un contexte différent de l'évaluation. Pour éviter un phénomène de « bruit », la valeur donnée à ces indices est très peu élevée (1 pour les positifs, -1 pour les négatifs). Il aurait sans doute été envisageable, dès cette étape, d'opérer une gradation entre ces divers termes.

[3] Cf. Legallois D & Ferrari S. (2006).

[4] Un nombre appréciable de travaux sur le discours évaluatif a été mené récemment en linguistique anglo-saxonne. Parmi ceux-ci : Martin J. & White P. (2005), Hunston S. & Thompson G. (2000), Bednarek (2006).



mauvais,	*insuffisant,*	*léger,*	*bon,*	*intéressant,*	*remarquable,*
infâme	*indigeste …*	*fade…*	*joli…*	*superbe …*	*divin …*

Faute d'une analyse plus poussée de la concordance de ces termes avec une évaluation réelle et donc, de courir le risque d'attribuer trop d'importance à un terme, nous nous contentons de les considérer au même niveau.

Cadres expérientiels.

Les corpus 1 et 2 sont consacrés à des textes d'opinions sur des objets culturels. De ce fait, nous avons pu réexploiter certains cadres expérientiels décrits par Legallois et Ferrari (2006). Une analyse de l'évaluation d'un objet culturel est vite confrontée à un problème inhérent à la constitution de l'objet même : on peut évaluer différents aspects ou *qualia* ; par exemple, le contenu, le style, la satisfaction ou la déception par rapport à des attentes, etc. L'évaluation peut porter également sur l'auteur du livre, sur l'histoire. Autrement dit, la forme de l'expression d'un jugement est naturellement configurée par rapport à des *cadres expérientiels*. Par exemple, le cadre de l'affect :

Ex : « *Le guépard* […] nous chavire le cœur à jamais »,
« Jane pleure. Et nous aussi nous pleurons. »

Dans notre analyse, nous retenons deux cadres dont les termes relatifs dénotent l'évaluation :

- l'**emprise** des objets évalués sur le lecteur :
« l'auteur plonge le lecteur dans la mythologie »,
« Alain Fleischer se met en marche pour envoûter le lecteur pendant plus de quatre cents pages » ,
« *Les associés* […] n'emporte jamais réellement l'adhésion du spectateur », …
« Il vampirise notre intérêt par sa bonhomie bienveillante et son côté bougon sympathique. »
- les **attentes** satisfaites ou non du lecteur :
« On reste sur notre faim »
« On peut regretter le classicisme du choix des auteurs »
« On déplorera par contre le saucissonnage artificiel et purement commercial de la série »
« […] font de ce film une agréable surprise estivale

Ces indices sont également annotés par un score propre qui, selon les cas, est égal à 1 ou –1.

3.1.2 Différents « poids » d'indices

Objets du domaine.

L'observation des différents corpus nous permet de constater une certaine régularité quant aux objets sur lesquels sont portés une évaluation. Il est possible de prévoir les termes désignant ces objets. L'hypothèse est alors de se dire : « Lorsqu'un terme évaluatif ou expérientiel concerne directement un terme du domaine, il est beaucoup plus probable que l'on soit en présence d'une évaluation réelle ».

Ex : « un article *intéressant* »
« un *beau* film »
« le roman nous *entraîne* dans l'intimité d'une famille bourgeoise »

Ce type d'indices nous semble plus convaincant que la simple présence des mots « intéressant », « beau », « entraîne », etc. qui peuvent intervenir hors-contexte évaluatif.

De plus, nous catégorisons deux types d'objets du domaine : les termes qui désignent **un objet général** particulièrement important, et ceux uniquement relatifs à **une partie de l'objet**. Nous considérons, dans une critique de film par exemple, que critiquer « un acteur » a moins d'impact sur l'opinion générale du texte que s'il s'agissait d'une critique portant sur l'objet « film ».

	Termes généraux (coef. 4)	**Termes partiels (coef. 2)**
Corpus 1	*film, roman, album, livre, spectacle, divertissement, comédie, …*	*personnage, histoire, scénario, acteur, dialogue, musique, décor, …*
Corpus 2	*jeu, titre, version, opus, …*	*niveau, mode, son, gameplay, univers, graphisme, prise en main, …*
Corpus 3	*article, papier, rapport, contribution, travail, étude, recherche, …*	*résultats, approche, méthode, outil, application, expérience, …*
Corpus 4	hypothèse non considérée sur ce corpus.	

Modificateurs d'intensité.

Nous avons vu précédemment que certains adverbes pouvaient être intrinsèquement évaluatifs (*judicieusement, clairement,* …) ; une autre caractéristique d'un certain nombre d'adverbes est de permettre à un auteur de moduler l'opinion qu'il souhaite faire partager.

Ex : « Un film particulièrement réussi »
« Un papier véritablement intéressant »

Ce rôle est également tenu par certains adjectifs, comme dans :
« un pur bonheur »
« un véritable échec »

La présence de ces modificateurs d'intensité associée à un indice évaluatif lexical augmente (ou diminue si l'indice initial est négatif) le poids de l'indice selon un coefficient de 2. Ainsi :

« un pur bonheur » : pur → coefficient (intensité) : 2
bonheur → évaluation intrinsèque : 1
score de l'indice → 2

Combiné avec la règle précédente, il est possible d'obtenir :

« un papier véritablement mauvais » : papier → coefficient (terme général du domaine) : 4
véritablement → coefficient (intensité) : 2
mauvais → évaluation intrinsèque : -1
score de l'indice → -8

Marques de négation.

Selon le même principe, nous tentons de tenir compte des tournures négatives pour inverser la valeur de l'indice repéré (coefficient –1).

« des personnages sans réelle saveur » : personnage → coefficient (terme partiel du domaine) : 2
sans → coefficient (négation) : -1
réel → coefficient (intensité) : 2
saveur → évaluation intrinsèque : 1
score de l'indice → -4

« approche ne me semble guère probante » : approche → coefficient (terme partiel du domaine) : 2
[ne … guère] → coefficient (négation) : -1
probant → évaluation intrinsèque : 1
score de l'indice → -2

Toutefois un certain nombre de tournures négatives ne sont pas considérées. En particulier, lorsque la marque de négation ne se situe pas à proximité de l'indice.

Marques de concession.

Après analyse du corpus 3, il nous a semblé intéressant d'envisager la pondération des indices inclus dans les tournures de phrases concessives.

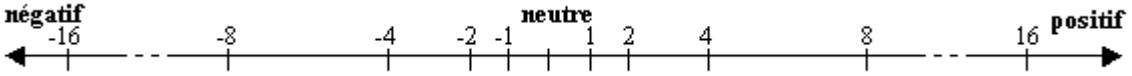
Ex : « la section 4 […] est intéressante mais ne propose aucune solution »

Dans cet exemple, l'adjectif « intéressant » laisse suggérer une évaluation positive, or l'aspect évaluatif de cette proposition est ambigu voire légèrement négatif. Dès lors, pour éviter de donner de

l'importance à des indices qui ne sont que les prémices d'une tournure concessive, nous annulons le score produit par ceux-ci. Cette hypothèse n'est testée que sur le corpus 3 pour lequel les tournures concessives sont relativement fréquentes et témoignent, a priori, de la nature des textes : des soumissions peuvent difficilement être complètement en dehors des attentes du relecteur, et une soumission peut toujours être améliorée.

Séquences lexico-grammaticales.

La combinaison de ces séquences lexico-grammaticales permet d'envisager une échelle d'indices résumée par la figure suivante :



score	indice	score	indice
-16	« un film véritablement très mauvais », …	+1	« poétique », « pas mal », …
-8	« un roman très ennuyeux », …	+2	« jolis effets spéciaux », « pas complètement inintéressant», …
-4	« l'introduction est particulièrement imprécise », …	+4	« beau film », « casting particulièrement réussi », …
-2	« approche guère probante », …	+8	« une BD très drôle », …
-1	« nul », «aucun intérêt »	+16	« cet album est une véritable grande réussite » …
0	« l'article est intéressant mais […] »		

3.1.3 Calcul du score à l'échelle du texte et à l'échelle de parties de texte

L'objectif de cette analyse est de produire, pour tous les textes d'un corpus, un score positif et un score négatif sur l'ensemble de l'énoncé en sommant les scores des indices trouvés. Du point de vue de l'analyse du discours, il nous a semblé cohérent de préciser également des scores propres à certaines parties du discours qui peuvent marquer plus fortement l'évaluation ou ayant des chances de refléter au mieux l'opinion associée à l'énoncé. En général, le premier et le dernier paragraphe (« introduction » et « conclusion ») ont ainsi un score qu'il peut être intéressant de préciser indépendamment du score général. L'hypothèse émise est que l'auteur aura tendance à annoncer la couleur de son opinion dès les premiers instants de l'énoncé, et qu'il pourra éventuellement synthétiser ses arguments en fin de texte. Cependant, les parties considérées sont variables selon les corpus. Après l'analyse préalable des différents corpus, nous avons constaté une récurrence de sections particulières sur un bon nombre de textes d'un corpus.

Ainsi, le **corpus 2** contient une partie sous-titrée par « Note Générale : ». Une telle section est susceptible de bien résumer la teneur des propos de l'auteur et est assimilable à une conclusion.

Certains textes du **corpus 3**, précisent explicitement par un sous-titre, l'objet d'une critique : « Commentaire », « Originalité », « Référence », « Importance », « Exactitude », « Rédaction ». D'emblée, l'analyse par classe de ce corpus nous montre qu'une critique portée sur la rédaction d'un article n'est absolument pas révélatrice de l'acceptation ou non de celui-ci. C'est pourquoi nous considérons comme une partie, l'ensemble des rubriques à l'exception de celle concernant la rédaction. Par ailleurs, la partie « Commentaire » semble faire l'objet d'une critique d'aspect général. Nous examinons donc indépendamment les indices contenus dans cette dernière.

Dans le **corpus 1**, le premier paragraphe représente le titre de l'œuvre. Nous considérons donc que l'introduction est constituée par l'union des deux premiers paragraphes. Les critiques étant assez longues, la conclusion est figurée arbitrairement par les deux derniers paragraphes.

Le **corpus 4** étant très souvent constitué de textes courts (un seul paragraphe), nous déterminons là aussi de façon arbitraire, que les deux premières phrases figurent l'introduction, et les deux dernières la conclusion.

Ainsi, nous obtenons différents scores d'indices positifs et négatifs pour chaque texte. Avec l'apport de la méthode d'extractions de motifs fréquents dans une matrice, nous espérons à ce stade que ces différents scores permettront une amélioration des résultats.

3.2 Mise en œuvre des traitements linguistiques

Nous utilisons *LinguaStream*, une plate-forme dédiée au TAL qui permet, dans une même chaîne de traitements, d'utiliser différents formalismes déclaratifs afin de marquer des objets linguistiques. Nous nous appuyons ici en particulier sur une grammaire Prolog (composant DCG Marker) et sur des expressions régulières. Afin de minimiser le besoin en ressources et le temps de calcul inhérent à ce genre d'analyse automatique, un traitement a préalablement scindé les fichiers XML des différents corpus à notre disposition. En début de chaîne, nous disposons d'un fichier XML pour chaque texte.

La chaîne de traitements de la figure 1 montre les différents composants utilisés pour notre analyse. Cette chaîne est exécutée automatiquement pour chaque texte. Après une segmentation en mots et une catégorisation grammaticale, quatre types d'expressions régulières enrichissent le fichier XML de base afin de marquer certaines parties du texte (cf. 3.1.3).

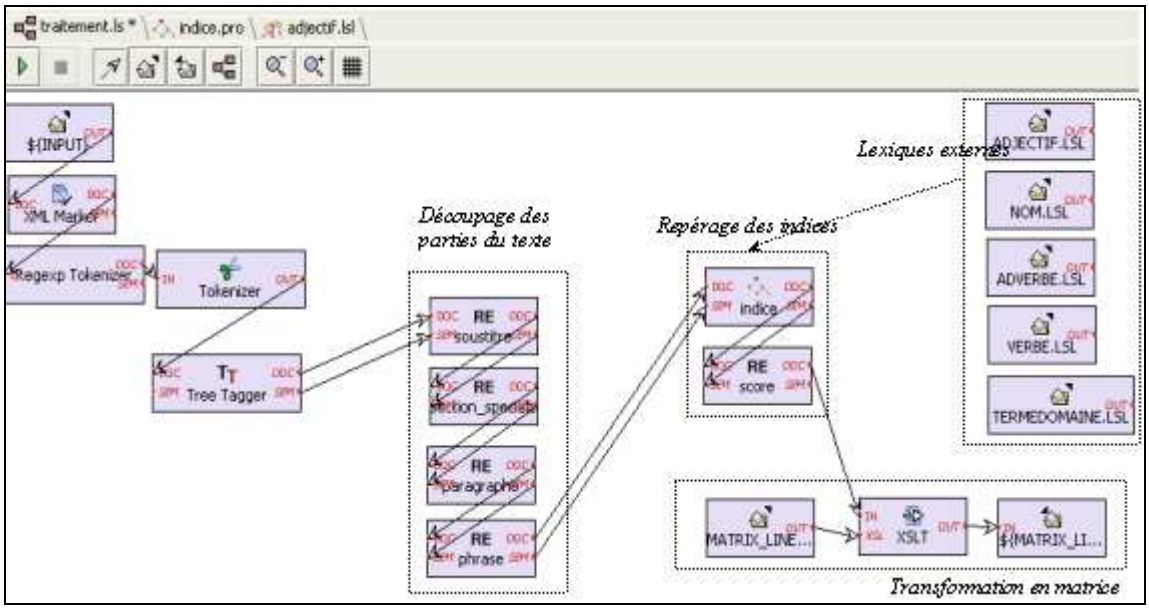


Fig. 1 – chaîne de traitements dans LinguaStream

Le composant « central » de notre approche est élaboré à partir du composant DCG Marker de LinguaStream. Il s'agit d'une grammaire Prolog qui opère un balisage des indices selon les règles décrites précédemment. Cette grammaire utilise des ressources lexicales externes constituées des formes lemmatisées des termes qui nous sont utiles. Pour chacun de ces termes, il est possible d'associer certains attributs qui correspondent aux caractéristiques propres que nous souhaitons leur donner.

En fin de chaîne, un dernier composant applique une transformation XSL au fichier XML enrichi par les marquages successifs. Les règles de transformation permettent de comptabiliser et regrouper les différents scores de l'énoncé. Au final, la chaîne produit un fichier texte constitué d'une ligne tabulée résumant l'évaluation des différentes parties du discours qu'il a nous semblé bon de considérer. Un post-traitement permet de reconstituer une matrice complète à partir de toutes les lignes produites.

3.3 Un classifieur supervisé

Le savoir-faire du GREYC en fouille de données concerne l'utilisation de méthodes à base de motifs ensemblistes. Les bases de données recensant des objets décrits par des attributs booléens, un motif est une conjonction d'attributs. Nous disposons depuis une dizaine d'années d'algorithmes performants pour

extraire les motifs fréquents (les motifs présents dans un nombre minimum d'objets) et construire les règles d'association. De la forme $X \rightarrow Y$, ces règles sont mesurées par une fréquence et une confiance, qui indiquent le nombre d'objets contenant $X \cup Y$ et la probabilité conditionnelle d'apparition de *Y* connaissant celle de *X*. Lorsque ces règles concluent sur un attribut de classe, elles peuvent être utilisées pour construire un classifieur automatique.

Plusieurs méthodes existent pour effectuer de la classification supervisée à partir de règles associations. Historiquement, la première et la plus simple est CBA (cf. Liu et *al.*, 1998)(Classification Based on Association). Cette méthode extrait les règles d'association de fréquence et confiance minimales désirées par l'utilisateur, et ordonne ces règles suivant leur confiance. Lorsqu'un nouvel exemple se présente, la première règle qui peut s'appliquer propose une valeur de classe.

Ce procédé a été raffiné par la méthode CMAR (cf. Li et *al.*, 2001) (Classification based on Multiple class-Association Rules) qui ne se contente plus d'une seule règle pour prendre la décision de classification. Les règles sont cette fois-ci mesurées par un indice de corrélation fourni par un χ^2 normalisé. On évite également la redondance entre les règles en ne conservant que celles qui sont à prémisse minimale. Un nouvel exemple sera classé à l'issue d'un vote réalisé par toutes les règles qui s'appliquent, selon leur pondération.

3.3.1 Notre méthode

Pour nos expériences, nous avons implémenté une méthode proche de CMAR, mais qui utilise des règles *disjonctives* ou *généralisées*. Contrairement aux règles d'association classiques, les règles généralisées sont de la forme $X \rightarrow \vee Y$ et concluent sur une disjonction d'attributs plutôt que sur une conjonction. Il est ainsi possible d'obtenir des règles *positives* (qui, en concluant sur un attribut de classe, entérinent la possibilité que l'exemple à classer appartienne à cette classe, si elle coïncide avec la prémisse) et des règles négatives (qui, en concluant sur la négation d'un attribut de classe, excluent la possibilité de classe correspondante) (cf. Antonie et Zaïane, 2004). Les règles positives sont de la forme $X \rightarrow c \vee Y$ (*c* est un attribut de classe) et se reformulent $X \overline{Y} \rightarrow c$. Les règles négatives de la forme $X c \rightarrow \vee Y$ et se reformulent $X \overline{Y} \rightarrow \overline{c}$. Dans les deux cas, elles s'appliquent pour classer tout exemple qui contient le motif *X*, mais aucun des attributs de *Y*.

Selon le modèle de CMAR, les règles sont pondérées par une mesure de χ^2. Pour un nouvel exemple, les règles positives voient leur pondération s'ajouter au score, les règles négatives soustraient leur pondération. Au final, la classe avec le meilleur score est désignée.

3.3.2 Aménagements pour les données du défi

La répartition des classes sur les différents corpus est très hétérogène. Dans cette configuration, les méthodes de classification à base d'association sont peu efficaces, car les classes dominantes fournissent plus de règles que les autres. Nous avons donc réalisé l'apprentissage sur un échantillon équilibré de chaque corpus.

D'autre part, nous avons limité la conclusion des règles à un singleton pour les corpus 1 et 2, car cela fournissait le meilleur résultat. En revanche, dans le corpus 3 qui contient peu d'objets, nous n'avons trouvé que peu de règles : nous avons alors dû extraire des règles généralisées dont la conclusion comportait jusqu'à trois attributs. Pour le très fourni corpus 4, le problème inverse s'est posé : le temps de calcul nécessaire à la constitution d'un classifieur fiable était insurmontable et nous avons renoncé à proposer une solution avec cette méthode.

4 Résultats et comparaison des approches

4.1 Tableau des résultats

Une analyse rapide des résultats obtenus (cf. Fig . 2) montre que la méthode « n-grammes » est la plus efficace sur les corpus 1, 2 et 4. De plus, les résultats sur le corpus 2 sont sensiblement meilleurs que les autres.

L'origine des différences dans les résultats nous semble variée : elle peut être liée à la nature des corpus fournis ou aux méthodes choisies. Certaines caractéristiques propres à chacun des corpus peuvent influer différemment selon les cas.

	« n-grammes »	*« analyse linguistique »*	*«approche combinée »*
Corpus 1	F-Score : 0,577 Préc. : 0,583 | Rappel : 0,571	F-Score : 0,457 Préc. : 0,444 | Rappel : 0,472	F-Score : 0,532 Préc. : 0,533 | Rappel : 0,532
Corpus 2	F-Score : 0,761 Préc. : 0,782 | Rappel : 0,741	F-Score : 0,506 Préc. : 0,493 | Rappel : 0,520	F-Score : 0,715 Préc. : 0,705 | Rappel : 0,726
Corpus 3	F-Score : 0,414 Préc. : 0,414 | Rappel : 0,414	F-Score : 0,474 Préc. : 0,476 | Rappel : 0,472	
Corpus 4	F-Score : 0,673 Préc. : 0,676 | Rappel : 0,669		

Fig. 2 – Tableau des résultats obtenus pour les différentes approches

4.2 Nature des corpus.

Corpus 1. La variété des objets critiqués aurait demandé une expertise humaine plus poussée pour bien considérer les usages linguistiques propres à ces différents genres d'énoncés[5]. On peut donc penser que les résultats de l'approche « linguistique » sont améliorables. Cette diversité conjuguée à la relative grande taille du corpus, permettant un entraînement, rend la méthode par « n-grammes » plus intéressante.

Corpus 2. Les tests de jeux vidéos ciblent un public précis, et le niveau de langue moins soutenu implique une variété lexicale et syntaxique plus réduite que pour les autres corpus. De plus, la présence de paragraphes récurrents présentant les différentes parties du jeu (graphismes, jouabilité, …) contribue à donner à ces textes un aspect « formulaire ». Par ces contraintes, le rédacteur de la critique est guidé à exprimer précisément les différentes facettes de son opinion, voire à réitérer son jugement. Ces considérations sont particulièrement rentabilisées dans le cas de l'utilisation des « n-grammes ».

Corpus 3. Un début de classification manuelle réalisée en amont a montré la difficulté, pour un humain, de déterminer la classe d'un texte. Une soumission peut avoir reçu une bonne critique mais ne pas être acceptée, ou inversement un article moyen peut tout de même être accepté. La variabilité du taux de sélection d'articles à une conférence nous semble être un paramètre important à prendre en compte ici. De ce fait, l'automatisation de cette tâche nous a paru dès le départ comporter une difficulté relativisant les faibles résultats pour ce corpus. La taille restreinte de ce corpus est un facteur qui peut expliquer le score plus faible de l'approche par « n-grammes » pour laquelle un entraînement est nécessaire.

Corpus 4. Il est à noter que certains textes exprimant une opinion sur un amendement à une loi et non directement sur cette dernière, le résultat enregistré peut être contraire à celui attendu, même par une expertise humaine. Il en résulte des résultats dans l'absolu un peu inférieurs à la réelle efficacité du traitement. Néanmoins, la méthode n-grammes réalise un score tout à fait intéressant, tirant bénéfice semble-t-il de la très grande taille du corpus. La méthode Linguastream n'a quant à elle pas pu être appliquée pour les raisons évoquées précédemment.

4.3 Nature des méthodes.

Une des propriétés importantes qui différencient les deux approches tient dans leur façon de discriminer les classes de textes et en particulier la classe intermédiaire (classe « 1 ») : *l'approche « n-grammes »* est capable de discriminer les trois classes de texte à partir des n-grammes révélateurs d'une classe par rapport aux deux autres classes. *l'approche « linguistique »*, quant à elle, cherche à dégager une forte présomption d'opinion négative ou positive à partir d'indices. La classe intermédiaire est, par conséquent, plus difficile à discriminer. La question est de savoir comment établir des indices concrets de

[5] Les travaux de Legallois D. & Ferrari F. (2006) s'intéressent essentiellement à l'évaluation des critiques de livres. Certaines hypothèses retenues ont été testées sur les autres objets de façon « aveugle ».

« neutralité » ou de savoir à quel moment il n'y a pas suffisamment d'indices majoritairement présent, et donc par défaut de considérer le texte comme étant de classe 1.

Par ailleurs, la différence dans la nature de ces méthodes nous permet d'envisager leur enrichissement mutuel. La constitution des lexiques de l'approche « linguistique » peut ainsi être améliorée par les n-grammes révélateurs d'une classe. Réciproquement, une expertise linguistique permettrait une évaluation de certains n-grammes et d'évacuer ceux qui paraissent non pertinents d'un point de vue sémantique.

Enfin, deux améliorations peuvent être envisagées. D'une part, la prise en compte de la catégorie syntaxique du n-gramme (syntagme nominal ou verbal) pourrait améliorer les résultats s'il s'avérait qu'une catégorie était plus pertinente qu'une autre. D'autre part, il serait intéressant d'étudier les effets d'une lemmatisation ciblée (par exemple ne lemmatiser que les verbes, que les noms, etc.) afin de voir l'impact qu'elle a sur chacune des catégories grammaticales sur les différents corpus.

5 Conclusion

L'équipe du GREYC a pu mettre à l'œuvre des compétences multiples pour relever ce défi, menant notamment à la mise en place d'un traitement à base de n-grammes, d'une chaîne de traitements linguistiques « LinguaStream », ainsi que d'un classifieur supervisé.

A l'issue des tests finaux, des différences notables apparaissent entre les approches. En l'état (nos travaux respectifs n'ont pu être menés que sur 2 mois), on constate que l'approche par n-grammes est celle qui produit les meilleurs résultats lorsque les corpus d'apprentissage sont suffisamment fournis, soit sur trois des quatre corpus du défi. Corollairement, l'approche plus directement sémantique est d'autant plus intéressante que l'apprentissage est effectué sur un corpus réduit, ce que l'on constate sur le corpus 3.

La combinaison des deux approches qui a été menée jusqu'à présent s'est contentée de s'appuyer sur les résultats pris indépendamment de ces deux dernières. Nous constatons un résultat dégradé par rapport à la meilleure méthode, l'autre méthode apportant plus de bruits que d'indices pertinents. Une combinaison des approches en amont de la classification supervisée nous paraît judicieuse ; elle pourra suivre les quelques idées présentées dans la partie 4.

Références

ANTONIE M.-L., ZAÏANE O. (2004), *An Associative Classifier based on Positive and Negative Rules*, ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'04), Paris, France.

BEDNAREK M. (2006), *Evaluation in Media Discourse*, Continuum.

HUNSTON S., THOMPSON G. (eds) (2000), *Evaluation in Text. Authorial Stance and the Construction of Discourse*, Oxford University Press.

LEGALLOIS D., FERRARI S. (2006), *Vers une grammaire de l'évaluation des objets culturels*, Schedae, prépublication n°8, fascicule n°1, pages 57-68.

LI W., HAN J., PEI J. (2001), *CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules*, IEEE International Conference on Data Mining (ICDM'01), San Jose, USA.

LIU B., HSU W., MA Y. (1998), *Integrating classification and association rules mining*, International Conference on Knowledge Discovery and Data Mining (KDD'98), New York, USA, pages 80-86.

MANGUIN J.L. (2005) *La dictionnairique Internet : l'exemple du dictionnaire des synonymes du CRISCO*, CORELA – Cognition, Représentation, Langage, Numéro spécial.

MARTIN J., WHITE P. (2005), *The Language of Evaluation: Appraisal in English*, Palgrave Macmillan Hardcover.

STUBBS M., BARTH I. (2003), *using recurrent phrases as text-type discriminators : a quantative method and some findings* in Functions of language 10 :1, 61-104.

WIDLÖCHER A., BILHAUT F. (2005), *La plate-forme LinguaStream : un outil d'exploration linguistique sur corpus*, In Actes de TALN 2005, Dourdan, France, pp. 517-522.

Classification d'opinions par méthodes symbolique, statistique et hybride

Sigrid Maurel, Paolo Curtoni et Luca Dini

CELI-France, SAS
38000 Grenoble
{maurel, curtoni, dini}@celi-france.com
http://www.celi-france.com

Résumé : La classification automatique de textes d'opinion est le DÉfi Fouille de Texte 2007. CELI-France présente trois méthodes différentes pour effectuer cette classification de textes de quatre corpus différents du DEFT'07. La première méthode est symbolique, la seconde statistique et la dernière, hybride, est une combinaison des deux premières. La combinaison permet de tirer parti des avantages des deux méthodes, à savoir la robustesse de l'apprentissage automatique statistique et la configuration manuelle symbolique orientée par rapport à l'utilisation d'applications réelles. L'*opinion mining* se fait au niveau de phrase, puis le texte entier est classé selon sa polarité en positif, moyen ou négatif.

Mots-clés : méthodes symbolique, statistique et hybride, classification d'opinions, analyse au niveau de phrase.

1 Introduction

DEFT'07 est la troisième édition de la conférence d'évaluation du DÉfi Fouille de Texte. Le thème de cette année est la classification de textes d'opinion, présents dans différents domaines de textes. Le corpus comprend des critiques de films, de livres et de disques, et de jeux vidéos. Dans un tout autre genre il y a des textes de débats politiques, et finalement des relectures d'articles scientifiques.

Un texte d'une critique de film, d'un article, etc. contient un jugement argumenté de l'auteur du texte, positif, moyen ou négatif, sur un sujet donné. Mais il contient aussi des parties sans sentiments, par exemple dans le résumé du livre sur lequel porte la critique. Le défi est de trouver avec précision les parties pertinentes pour la classification automatique du texte entier.

CELI-France a mis au point trois méthodes pour classer les textes des différents corpus mis à disposition par le comité d'organisation du DEFT'07. Ces trois méthodes ont servi pour traiter les corpus *aVoiraLire* (critiques de films, livres, disques, etc.), *jeuxvidéo* (critiques de jeux vidéo) et *relectures* (relectures d'articles scientifiques) ; seule la méthode statistique a été utilisée pour le corpus des *débats politiques* (notes de débats parlementaires).

La première est une méthode symbolique qui inclut un système d'extraction d'information (adapté aux corpus). Elle est basée sur les règles d'un analyseur syntaxico-sémantique. La deuxième est une méthode statistique basée sur des techniques d'apprentissage automatique. Enfin, une dernière méthode combine les techniques des deux précédentes.

Après une brève présentation des corpus, nous décrivons les trois méthodes développées.

2 Les corpus

Les corpus mis à disposition par DEFT'07 sont assez différents les uns des autres, que ce soit par la taille des corpus que par la taille de chaque texte. Certains sont structurés, d'autres non, et le nombre de fautes d'orthographe présent dans les textes varie aussi beaucoup (pour plus de détails c.f. section 3.2.1).

Les textes sur les jeux vidéo sont en général beaucoup plus longs que les textes de critiques de films et livres. Ils sont aussi plus structurés avec des parties différentes concernant le graphisme, le scénario, la jouabilité, etc. La longueur des critiques de films et livres varie beaucoup d'un texte à l'autre, on ne peut pas en dégager de structure interne.

Environ un tiers des relectures d'articles scientifiques est structuré en différentes parties : la rédaction, l'originalité, l'importance, etc., mais les deux-tiers restants ne le sont pas, et en général ces textes sont beaucoup plus denses que ceux du premier tiers.

La longueur des textes rapportant des débats politiques varie énormément. Ce corpus est le plus important en taille, il contient environ dix fois plus de textes que celui des jeux vidéo et des critiques de films et livres, et presque vingt fois plus que celui des relectures d'articles.

Aux trois premiers corpus sont associées trois classes de jugement (*positif*, *moyen* et *négatif*), alors que deux classes de jugement seulement (*pour* (c'est-à-dire *positif*) et *contre* (*négatif*)) sont associées au dernier corpus.

3 Méthode symbolique

La méthode symbolique se base sur une analyse syntaxique du texte faite par un analyseur fonctionnel et relationnel (c.f. les travaux sur l'analyse syntaxique et sémantique de (Aït-Mokhtar *et al.*, 2001; Basili *et al.*, 1999; Dini, 2002; Dini & Mazzini, 2002)). Cet analyseur traite un texte donné en entrée phrase par phrase et en extrait, pour chaque phrase, les relations syntaxiques présentes. Il s'agit de relations syntaxiques de base telles que le modifieur d'un nom, d'un verbe, le sujet et l'objet de la phrase, etc., et de relations plus complexes telles que la coréférence entre deux syntagmes de la phrase.

La possibilité est donnée à l'utilisateur d'élaborer une grammaire à sa guise et d'ajouter de nouvelles règles pour extraire les relations auxquelles il s'intéresse. Pour ce faire il peut modifier les règles d'extraction de relations (par exemple ajouter des règles pour de nouvelles relations), augmenter/diminuer les traits sur les mots dans le lexique qui agissent sur les règles, enlever certaines parties du traitement, etc.

À la fin de l'analyse un indice de confiance est calculé par la méthode symbolique. Il servira à la méthode hybride (c.f. section 5) pour déterminer le résultat final.

La polarité POS/NEG/MOY attribuée au texte entier dépend du rapport entre la quantité de relations d'opinions positives et négatives. Celle-ci est évaluée en relation avec la quantité de relations d'opinions moyennes. Une majorité de relations d'opinions positives détermine une polarité positive du texte, tandis qu'une majorité de relations d'opinions négatives provoque une polarité négative du texte entier. L'équilibre entre relations d'opinions positives et négatives entraine une classification moyenne du texte. La quantité de relations d'opinion moyenne influence la quantité de relations d'opinions positives et négatives nécessaire afin que soit attribué au texte une polarité positive ou négative.

L'algorithme pour la classification utilise les formules suivantes :

```
POS, NEG, MOY = nombre d'opinions positives, négatives et moyennes

BalanceInterval=((0.133)*(MOY/(POS+NEG+MOY)))+(0.066)

PosNegRatio=(POS/(POS+NEG))+((0.5-(POS/(POS+NEG)))*(MOY/(POS+NEG+MOY)))

IF((0.5-BalanceInterval) <= PosNegRatio <= (0.5+BalanceInterval))
     ASSIGNMENT=MOY
     SCORE=(BalanceInterval-ABS(PosNegRatio-0.5))/BalanceInterval
ELSE IF(POS>NEG)
     ASSIGNMENT=POS
     SCORE=POS/(POS+NEG+MOY)
ELSE
     ASSIGNMENT=NEG
     SCORE=NEG/(POS+NEG+MOY)
```

La méthode symbolique a été utilisée pour les corpus *aVoiraLire*, *jeuxvidéo* et *relectures d'articles*. Les textes du corpus des *débats politiques* contiennent des textes trop pauvres en sentiments exprimés directement, c'est-à-dire qu'ils contiennent trop peu, voire pas du tout, de mots ou de syntagmes qui expriment des sentiments et qui obéissent aux règles définies par la grammaire. La raison de ce comportement de l'analyseur est que le style des discours parlementaires est trop éloigné du style des textes du tourisme pour lequel la grammaire a initialement été développée (c.f. la section 3.1).

Les textes du corpus du tourisme utilisés pour configurer la grammaire de base proviennent d'un site de forums sur internet. Le style des forums et des newsgroups est assez particulier. Le style des discours

parlementaires est beaucoup plus formel, les phrases sont plus longues et le langage est plus soutenu. C'est pour cette raison que seule la méthode statistique a été utilisée pour ce dernier corpus.

3.1 Grammaire

La grammaire utilisée a été initialement développée afin d'extraire les relations de sentiments exprimés dans une phrase dans le cadre d'un projet sur le tourisme en France. Une relation de sentiment a en général deux arguments : le premier est l'expression linguistique qui véhicule le sentiment en question, le deuxième la cause du sentiment (si la cause est exprimée dans la phrase).

Ceci donne pour la phrase « J'aime beaucoup Grenoble. » la relation `SENTIMENT_POSITIF(aimer, Grenoble)`. L'attribut `POSITIF` de la relation, c'est-à-dire la valeur de sa classe, indique qu'il s'agit d'un sentiment positif.

L'analyse se base sur les mots du lexique qui ont reçu des traits spécifiques qui marquent le sentiment positif ou négatif. Les relations de sentiments moyens sont extraites sur la base de constructions de phrases. Il s'agit pour la plupart de verbes (*aimer*, *apprécier*, *détester*, ...) et d'adjectifs (*magnifique*, *superbe*, *insupportable*, ...), mais aussi de quelques noms communs (*plaisir*) et d'adverbes (*malheureusement*). L'attribut de la relation (`positif` ou `négatif`[1]) d'un sentiment sera inversé en cas d'une négation dans la phrase. Si possible, les pronoms *qui* et *que* qui se rapportent à une entité présente ailleurs dans la même phrase, seront remplacés par cette entité (e.g. « Grenoble est une ville qui vaut vraiment le détour hiver comme été. »).

Cette configuration de la grammaire générique a été faite sur la base d'un travail d'annotation manuelle (à l'aide du logiciel Protégé 3.2[2] avec le plugin Knowtator[3]) de textes venant du domaine du tourisme. Ce corpus du tourisme contient une centaine de textes dont environ 75 ont été annotés. Chaque texte est composé de messages des utilisateurs du forum ; la longueur varie entre dix et 55 messages par document. Un message peut ne contenir qu'une phrase ou plusieurs paragraphes. Le corpus entier occupe un peu plus d'un mégaoctet sur le disque dur. L'annotation de ce corpus avec Protégé et Knowtator a été faite dans la lignée des travaux de (Riloff *et al.*, 2006; Wiebe & Mihalcea, 2006; Riloff *et al.*, 2005).

L'annotation inclut les informations de cause, d'intensité et de l'émetteur du sentiment. Dans « J'aime énormément Grenoble. » *aimer* véhicule le sentiment, *Grenoble* est la cause du sentiment et *je* est l'émetteur du sentiment. L'adverbe *énormément* exprime l'intensité, le sentiment ici est plus intense que dans la phrase « J'aime bien Grenoble. ».

L'annotation pour le tourisme ne contient pas seulement les valeurs *positif* et *négatif* pour classer les sentiments, mais est détaillée beaucoup plus finement (voir par exemple les travaux de (Mathieu, 2000, 2006)). Le schéma d'annotation choisi est plus fin que la simple opposition positif-négatif. En effet, nous avons repris la taxonomie d'(Ogorek, 2005) qui propose 33 sentiments (17 positifs et 16 négatifs) différents auxquels nous avons ajouté les pseudo-sentiments *bon-marché*, *conseil*, *cher* et *avertissement*. Ces sentiments sont classés en sous-groupes comme *amour*, *joie*, *tristesse*, *mépris*, etc.

3.2 Grammaires pour DEFT'07

Nous avons dû adapter la grammaire de base pour participer à DEFT'07. Elle a été paramétrée pour répondre aux besoins des différents corpus DEFT'07 ; du point de vue lexical mais aussi pour résister aux fautes d'orthographes répétitives. Le point le plus important à modifier a été la classification des sentiments et en particulier l'introduction de la notion de sentiment moyen. En effet pour le tourisme il n'y a que des sentiments positifs et négatifs. Notre approche pour le projet du tourisme est qu'au niveau des phrases les sentiments sont positifs ou négatifs. Il n'est pas nécessaire d'utiliser des sentiments moyens dans le domaine du tourisme, dans la mesure où la taxonomie utilisée (c.f. ci-dessus) permet de nuancer suffisamment.

Nous avons donc construit deux nouvelles grammaires différentes de celle utilisée initialement. La première, commune aux corpus *aVoiraLire* et *jeuxvidéo*, la deuxième, pour le corpus des *relectures*.[4]

[1] L'attribut `moyen` évoqué dans l'introduction n'existe pas dans la grammaire de base, il a été ajouté seulement dans les grammaires DEFT'07 pour répondre aux besoins des corpus.

[2] `http://protege.stanford.edu/`

[3] `http://bionlp.sourceforge.net/Knowtator/index.shtml`

[4] Rappel : Il n'existe pas de grammaire spéciale pour le corpus des *débats politiques*, ce corpus n'a été traité uniquement que par la méthode statistique (c.f. section 4).

L'objectif de ces trois grammaires (celle de base (c.f. la section précédente 3.1) et celles pour DEFT'07) est d'extraire un maximum d'informations du texte, les sentiments positifs, moyens et négatifs. Au final, CELI-France souhaite utiliser les méthodes de l'*opinion mining* et pas seulement de la classification textuelle. L'analyse des textes au niveau des phrases convient bien à cette approche.

Les textes sont analysés phrase par phrase. Chaque phrase peut contenir zéro, une ou plusieurs relations de sentiment. Il est tout à fait possible d'avoir des relations de sentiments positifs et négatifs dans une même phrase. À la fin de l'analyse du texte un sentiment global lui est attribué selon le nombre de relations positives, moyennes et négatives qu'il contient.

Les deux grammaires DEFT'07 se distinguent l'une de l'autre essentiellement par le lexique de mots qui reçoivent les traits `positif` et `négatif` correspondants aux valeurs des classes des textes (et par leur liste de termes, c.f. section 3.3). Par exemple, le lexique de la grammaire des *relectures* contient les mots *bon* et *bien* (« Cet article est *bien* écrit. », « Le papier est d'une *bonne* qualité. »). Ces mots ont été supprimés du lexique de la grammaire *aVoiraLire/jeuxvidéo* parce qu'ils produisent trop de relations éronnées (« Il a fait *bien* des choses avant de se mettre à faire des films. », « L'actrice est venue au *bon* moment. »). Souvent il ne s'agit pas de vrais sentiments.

Des règles spéciales pour créer des relations pour les textes avec un jugement de sentiment moyen ont été ajoutées. Par exemple quand une phrase contient un sentiment positif et un sentiment négatif coordonnés par *mais*, l'attribut des sentiments est remplacé par l'attribut `moyen` (« Ce jeu est *amusant* au début **mais** *ennuyant* la deuxième semaine. »). Ceci aide à classifier un texte qui contient des phrases avec des sentiments positifs et négatifs. Le texte entier sera alors classé comme moyen.

3.2.1 Les fautes d'orthographe dans les textes

Les fautes d'orthographe dans les textes des corpus posent parfois des problèmes d'analyse. Un mot où l'accent est mal mis peut changer complètement l'analyse. Prenons la phrase « Cet article me parait inacceptable car extrêmement confus et peu compréhensible, en tout cas par quelqu'un ne disposant que des informations contenues dans l'article. » Ici, il manque l'accent circonflexe sur le *i* de *parait*, le verbe. L'analyse retourne l'infinitif *parer* au lieu de *paraître*. La relation d'objet prédicatif entre *me* et *paraître* ne peut pas être extraite. C'est une condition indispensable pour l'extraction de la relation de sentiment entre *article* et *inacceptable*.

Heureusement, la grammaire permet d'intercepter certains des cas les plus fréquents, par exemple par l'ajout de mots mal orthographiés au lexique (*interressant*, sans accent et avec deux *r*, est apparu plusieurs fois dans les corpus.).

Les règles syntaxiques sont souvent assez souples pour faire l'accord entre un nom et un adjectif, ou un nom et un verbe même si le *e* ou le *s* manque. Mais malheureusement, il y a aussi des textes tellement mal écrits dans les corpus que l'analyse peut échouer. D'après ce que nous avons pu observer ces textes sont heureusement en minorité dans les corpus et ne modifient pas trop les résultats.

3.3 Listes de termes

Une liste de termes a été élaborée pour chaque corpus. Chaque liste contient les noms qui sont propres au domaine du corpus, voici quelques exemples. Pour le corpus *aVoiraLire* : *film*, *livre*, *album*, ... ; pour le corpus *jeuxvidéo* : *jeu*, *graphisme*, *soft*, ... ; et pour le corpus *relectures* : *article*, *papier*, *résultat*, ... Grâce à ces listes, des relations erronées, c'est-à-dire dont le deuxième argument n'est pas dans la liste parce qu'il n'appartient pas au domaine, peuvent être refusées.

Par exemple dans le corpus *aVoiraLire* cette mesure s'applique à la plupart des relations extraites de la partie résumé du film, du livre, etc. Prenons la phrase suivante : « Le héros a passé une *magnifique* journée. » Le mot *journée* n'étant pas dans la liste, la relation `SENTIMENT_POSITIF(magnifique,journée)` est ainsi refusée. Ce qui répond bien au défi car cette phrase ne contient pas un sentiment exprimé de l'auteur du texte (ce qui serait le cas par exemple la phrase « Ce livre est vraiment *magnifique*. »), mais fait partie du résumé de l'histoire.

Chaque liste ne contient que des noms communs, toutes les relations qui contiennent des noms propres sont gardées telles quelles (« Bref, inutile de dépenser le moindre euro pour ce *Yetisports* qui n'en vaut vraiment pas la chandelle. »). Les relations extraites à partir de mots qui ne sont pas des noms (« Je n'aime pas aller au cinéma. » ⇒ `SENTIMENT_NEGATIF(aimer,aller)`) sont gardées elles aussi.

Ces listes ont été élaborées automatiquement à partir des textes des corpus d'entrainement. Elles contiennent tous les noms en deuxième argument d'une relation, si cette relation a été extraite d'un texte de la même valeur de la classe et que l'indice de confiance calculé dépasse un seuil prédéfini. C'est-à-dire : un nom dans une relation de sentiment positif doit se trouver dans un texte avec la valeur *positif* de la classe dans le corpus d'entrainement.

À l'intérieur de chaque liste, les termes sont ordonnés par la fréquence avec laquelle ils ont satisfait les conditions d'extraction en relations. L'utilisation de ces listes permet d'augmenter le F-score des résultats d'environ 5-10%, selon le corpus.

4 Méthode statistique

La méthode statistique est une technique d'apprentissage automatique qui se base sur (Pang & Lee, 2004; Pang *et al.*, 2002; Pang & Lee, 2005)[5]. Nous l'avons adaptée aux corpus de langue française. Nous l'avons testé d'une part sur les textes du projet sur le tourisme, et d'autre part sur les textes des corpus du projet DEFT'07. (Pang & Lee, 2004) proposent deux axes de classification possibles, soit dans l'opposition subjectif-objectif, soit dans la distinction des opinions subjectives dans l'opposition positif-négatif. La technique de base de la méthode de (Pang & Lee, 2004) ne considère donc pas les sentiments moyens. C'est la raison pour laquelle nous avons dû ajouter une méthode pour ceux-ci.

(Pang & Lee, 2004) améliorent la classification de l'axe positif-négatif en supprimant d'abord du texte toutes les phrases objectives et en faisant la classification seulement sur la partie subjective. Cet *extract* correspond dans leurs expérimentations à 60% du texte original. Nous n'avons pas retenu cette façon de faire à cause du manque d'un corpus d'entrainement ayant des parties subjectives et objectives bien distinctes. Nous avons choisi de faire la séparation de texte subjectif-objectif à l'aide de la méthode symbolique (c.f. section 3) qui permet d'obtenir finalement des résultats plus nuancés. Les extraits peuvent être vus comme de bons résumés du texte.

La méthode statistique se base sur des n-gram. Pour les projets sur la langue française (DEFT'07 et le tourisme) nous avons choisi $n = 12$. Comme pour la méthode symbolique, un indice de confiance est attribué aux textes. Il permet de comparer le résultat avec celui de la méthode symbolique pour en conclure le résultat final avec la méthode hybride. Pour l'entrainement des textes pour ce projet, les techniques de *support vector machines* (SVM) et de *naive bayes* (NB) ont été utilisées. Les résultats sont légèrement meilleurs avec NB, mais ceci reste négligeable.

Nous avons travaillé dans un premier temps avec la partie qui distingue les phrases subjectives des phrases objectives. Malheureusement, les résultats ne sont pas très encourageants, ceci est dû au fait que dans les corpus d'entrainement, les parties subjectives ne sont pas clairement distinctes des parties objectives. Souvent, les deux parties sont mélées l'une à l'autre.

Des expérimentations ont été faites avec le corpus *aVoiraLire*, en prenant seulement la/les première(s) et/ou la/les dernière(s) phrase(s) du texte. Nous sommes partis de l'hypothèse que le jugement de l'auteur dans une critique de livre ou de film se trouve la plupart du temps au début ou à la fin du texte, la place du milieu étant vraisemblablement occupée par le résumé du livre ou du film. Les résultats de classification positif ou négatif avec cette technique sont meilleurs qu'en prenant le texte en entier, mais cette technique n'a finalement pas été retenue, car elle ne sera pas facilement reproductible sur des textes provenants d'autres domaines que la critique de film et de livre, où il n'y a pas forcément un résumé au milieu du texte.

Pour le projet DEFT'07 l'entrainement du module statistique a donc été réalisé uniquement sur les phrases de chaque texte qui ont été sélectionnées par la méthode symbolique, qui contiennent donc des sentiments, et selon les valeurs de leur classe (positif, moyen ou négatif) attribuées à chaque corpus par le comité d'organisation. Les résultats soumis correspondent à cet entrainement. Ils sont ensuite confrontés aux résultats de la méthode symbolique pour donner un résultat final pour chaque texte.

5 Méthode hybride

La méthode hybride est une combinaison des deux méthodes précédentes (c.f. sections 3 et 4). Elle prend en entrée les sorties des deux autres méthodes et calcule d'après les indices de confiance de chaque résultat, une moyenne qui sera traduite en positif, moyen ou négatif.

[5] `http://www.cs.cornell.edu/home/llee/papers.html`

Cette méthode a été utilisée pour trois des quatre corpus donnés (*aVoiraLire*, *jeuxvidéo* et *relectures*). Elle a donné les meilleurs résultats pour les corpus *jeuxvidéo* avec un F-score de 0,71, contre 0,54 (méthode symbolique) et 0,70 (méthode statistique) et *relectures* avec un F-score de 0,54, contre 0,48 (méthode symbolique) et 0,51 (méthode statistique).[6] Pour le corpus *débats politiques* seule la méthode statistique a été utilisée.

La classification définitive est calculée avec les deux méthodes symbolique et statistique. Les résultats respectifs sont confrontés pour obtenir une classification finale. La pondération exacte est encore sujet d'expérimentations et en cours de travail.

Grâce au refus de certaines relations de sentiment erronées par la méthode symbolique, l'entrainement de la méthode statistique se fait sur un corpus plus homogène, et le résultat sera plus précis.

C'est une approche qui permet de garder la robustesse de l'apprentissage automatique de la méthode statistique et d'orienter en même temps la base de l'entrainement sur une configuration manuelle de la méthode symbolique. Ceci permet de corriger de façon significative les erreurs de l'apprentissage automatique et d'intégrer les spécificités du cahier des charges, c'est-à-dire les particularités de chaque corpus (à l'aide de lexiques et de listes de termes différents selon le domaine d'application).

6 Conclusion

La combinaison des méthodes symbolique et statistique a donné des résultats plus précis que chacune des méthodes employée séparément.

La figure 1 résume les résultats obtenus. Pour les *débats politiques* il n'y a qu'une valeur du F-score puisque ce corpus n'a été traité que par la méthode statistique.

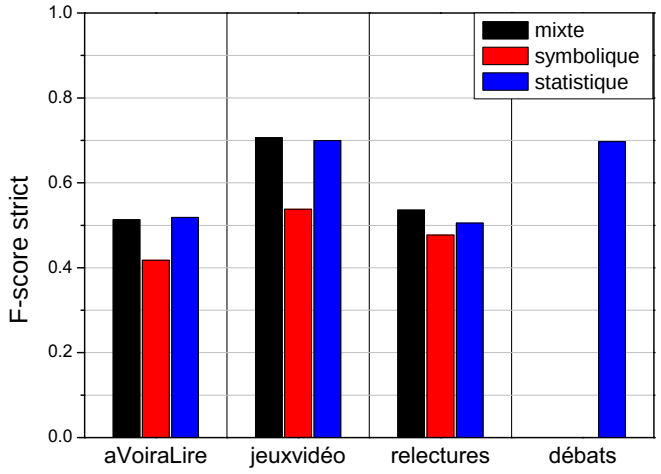


FIG. 1 – Les résultats, le F-score strict, de chaque corpus selon la méthode de classification utilisée.

L'intérêt de la méthode hybride repose dans la considération des contextes d'application de ses produits. Il est bien connu que la méthode purement symbolique a souvent pour le client un coût d'entrée plutôt élevé. Cette considération est liée au temps de configuration, de repérage ou de la création de lexiques spécifiques, de taxonomies etc. On voit donc que la méthode hybride obtient de meilleurs résultats, cela permet de profiter des avantages d'une approche symbolique sans pour autant en avoir tous les inconvénients.

L'utilisation d'une méthode hybride permet, au contraire, de minimiser les coûts de configuration, en réduisant une partie du travail à l'annotation de textes, une tâche qui dans la plupart des cas peut être réalisée

[6] Pour le corpus *aVoiraLire* le meilleur résultat a été obtenu par la méthode statistique avec un F-score de 0,52, contre 0,51 (méthode hybride) et 0,42 (méthode symbolique). Ce corpus n'est probablement pas assez uniforme (il parle de livres, films actuels au cinéma, disques, films plus anciens enregistrés, ...) pour pouvoir faire une liste de termes plus performante.

par le client lui-même. Les algorithmes d'apprentissage automatique sont alors en mesure de donner des premiers jugements au niveau du texte entier.

Or, le jugement donné par rapport à un texte est souvent d'une utilité limitée. Par exemple savoir que, dans un forum sur la téléphonie, il y a 30.000 messages à polarité positive et 15.000 à polarité négative, n'est pas le type d'information qui peut être activement utilisée par une compagnie de téléphonie. Déjà en intégrant la polarité avec un système d'extraction d'entités nommées (marque, modèle, etc.), on peut avoir des résultats plus spécifiques et donc plus informatifs.

Ce qui est le plus important, c'est qu'avec ce type de système statistique on peut ajouter, selon la méthode exposée dans cet article, une couche *symbolique* au fur et à mesure, de plus en plus importante dès que les exigences d'une application deviennent plus précises. On peut par exemple superposer une couche d'identification de jugement, qui permet d'avoir une visibilité sur les jugements sans devoir lire le texte dans son entier. On peut identifier certains patrons sémantiques qui sont d'importance capitale pour une application donnée et qui doivent avoir la priorité sur les résultats statistiques (par exemple le soucis de securité exprimé par les internautes sur un certain modèle de voiture).

Les exemples pourraient être multipliés. Ce qui est important est que, pour des raisons commerciales, la démarche hybride que nous avons tenue en DEFT'07 est importante non seulement pour des raisons scientifiques de performance (le meilleur résultat entre les technologies que nous avons adoptées) mais, aussi et surtout pour des raisons de développement et acceptation par le marché.

Références

AÏT-MOKHTAR S., CHANOD J.-P. & ROUX C. (2001). A multi-input dependency parser. In *Actes d' IWPT*.

BASILI R., PAZIENZA M. T. & ZANZOTTO F. M. (1999). Lexicalizing a shallow parser. In *Actes de TALN'99*.

DINI L. (2002). Compréhension multilingue et extraction de l'information. In F. SEGOND, Ed., *Multilinguisme et traitement de l'information (Traité des sciences et techniques de l'information)*. Editions Hermes Science.

DINI L. & MAZZINI G. (2002). Opinion classification through information extraction. In ZANASI, BREBBIA, EBECKEN & MELLI, Eds., *Data Mining III*, p. 299–310. WIT Press.

MATHIEU Y. Y. (2000). *Les verbes de sentiment. De l'analyse linguistique au traitement automatique*. CNRS Editions.

MATHIEU Y. Y. (2006). A computational semantic lexicon of french verbs of emotion. In J. G. SHANAHAN, Y. QU & J. WIEBE, Eds., *Computing attitude and affect in text*: *Theorie and applications*, p. 109–124. Springer.

OGOREK J. R. (2005). Normative picture categorization: Defining affective space in response to pictorial stimuli. In *Actes de REU'05*.

PANG B. & LEE L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Actes d' ACL'04*, p. 271–278.

PANG B. & LEE L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Actes d' ACL'05*, p. 115–124.

PANG B., LEE L. & VAITHYANATHAN S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Actes d' EMNLP'02*, p. 79–86.

RILOFF E., PATWARDHAN S. & WIEBE J. (2006). Feature subsumption for opinion analysis. In *Actes d' EMNLP'06*, p. 440–448.

RILOFF E., WIEBE J. & PHILLIPS W. (2005). Exploiting subjectivity classification to improve information extraction. In *Actes d' AAAI'05*.

WIEBE J. & MIHALCEA R. (2006). Word sense and subjectivity. In *Actes d' ACL'06*, p. 1065–1072.

Comment faire pour que l'opinion forgée à la sortie des urnes soit la bonne ? Application au défi DEFT 2007

Juan Manuel Torres-Moreno[1,2], Marc El-Bèze[1], Fréderic Béchet[1], Nathalie Camelin[1]

[1] Laboratoire Informatique d'Avignon, BP 1228,
84911 AVIGNON Cedex 9 FRANCE
{juan-manuel.torres, marc.elbeze, frederic.bechet}@univ-avignon.fr
http://www.lia.univ-avignon.fr/equipes/TALNE

[2] Ecole Polytechnique de Montréal, Département de génie informatique,
H3C3P8 Montréal (Québec) Canada

Résumé : Nous présentons des modèles d'apprentissage probabilistes appliqués à la tâche de classification telle que définie dans le cadre du défi DEFT07 : la classification d'un texte suivant l'opinion qu'il exprime. Pour classer les textes, nous avons utilisé plusieurs classifieurs et une fusion. Une comparaison entre les résultats en validation et en tests montrent une coïncidence remarquable, et mettent en évidence la robustesse et performances de la fusion que nous proposons. Les résultats que nous obtenons, en termes de précision, rappel et F-score sur les sous corpus de test sont très encourageants.

Mots-clés : Méthodes probabilistes, Apprentissage automatique, Classification de textes par leur contenu, défi DEFT.

1 Introduction

Dans le cadre de la plate-forme AFIA 2007[1], sera organisé en juillet 2007 à Grenoble (France) un atelier centré sur un défi qui avait pour objet la fouille de textes. Ce défi est la troisième édition de DEFT (DÉfi Fouille de Textes) (Azé & Roche, 2005; Azé *et al.*, 2006). De notre coté, ceci est la deuxième participation dans DEFT de l'équipe TALNE du Laboratoire Informatique d'Avignon (LIA)[2]. Lors de la première competition en 2005 (El-Bèze *et al.*, 2005), notre équipe avait remporté le défi. A l'époque, le problème était de classer les segments des allocutions de Jacques Chirac et François Mitterrand préalablement mélangées[3].

Le défi actuel a été motivé par le besoin de mettre en place des techniques de fouille des textes permettant de classer de textes suivant l'opinion qu'ils expriment. Concrètement, il s'agit cette fois de classer les textes de quatre corpus en langue française selon les opinions qui y sont formulées.

La classification d'un corpus en classes pré-déterminées, et son corollaire le profilage de textes, est une problématique importante du domaine de la fouille de textes. Le but d'une classification est d'attribuer une classe à un objet textuel donné, en fonction d'un profil qui sera explicité ou non suivant la méthode de classification utilisée. Les applications sont variées. Elles vont du filtrage de grands corpus (afin de faciliter la recherche d'information ou la veille scientifique et économique) à la classification par le genre de texte pour adapter les traitements linguistiques aux particularités d'un corpus.

La tâche proposée par DEFT'07 vise le domaine applicatif de la prise de décision. Attribuer une classe à un texte, c'est aussi lui attribuer une valeur qui peut servir de critère dans un processus de décision. Et en effet, la classification d'un texte suivant l'opinion qu'il exprime a des implications notamment en étude de marchés. Certaines entreprises veulent désormais pouvoir analyser automatiquement si l'image que leur renvoie la presse est plutôt positive ou plutôt négative. Des centaines de produits sont évalués sur Internet par des professionnels ou des internautes sur des sites dédiés : quel jugement conclusif peut tirer de cette masse d'informations un consommateur, ou bien encore l'entreprise qui fabrique ce produit ? En dehors du marketing, une autre application possible concerne les articles d'une encyclopédie collaborative sur

[1] Association Française pour l'Intelligence Artificielle, http ://afia.lri.fr

[2] http ://www.lia.univ-avignon.fr

[3] Pour plus de détails concernant DEFT'05, voir le site http ://www.lri.fr/ia/fdt/DEFT05

Internet comme Wikipédia : un article propose-t-il un jugement favorable ou défavorable, ou est-il plutôt neutre suivant en cela un principe fondateur de cette encyclopédie libre ?

A priori, un travail de classification des avis d'opinion paraît simple. Or, de nombreuses raisons font que le problème est complexe. Facteur aggravant : on ne dispose que de corpus de taille moyenne, déséquilibrés par rapport à leurs classes.

Dans cet article, nous décrivons quelques méthodes employées dans le cadre de ce défi. Nous décrivons en section 2 les corpus et la méthode d'évaluation proposée. En Section 4, nous présentons les outils de classification de texte utilisées. La représentation de textes ainsi qu'une agglutination et normalisation graphique sont detaillés en section 3. Nos outils de classification sont décrits en section 4. Des expériences et résultats sont rapportés et discutés en section 5, avant de conclure et d'envisager quelques perspectives.

2 Description des corpus

Les organisateurs du défi DEFT'07 ont mis à la disposition des participants quatre corpus hétérogènes :
- Critiques de films, livres, spectacles et bandes dessinées (**aVoiraLire**) ;
- Tests de jeux vidéo (**jeuxvideo**) ;
- Relectures d'articles scientifiques (**relectures**) ;
- Débats parlementaires (**debats**).

On trouvera ci-après une brève description de chacun d'entre eux :

aVoiraLire. Ce corpus comporte 3 460 critiques et les notes qui leur sont associées. Etant donné que beaucoup d'organes de diffusion de critiques de films ou de livres[4] attribuent, en plus du commentaire, une note sous la forme d'une icône. Les organisateurs du défi ont retenu une échelle de 3 niveaux de notes. Ceci donne lieu à 3 classes bien discriminées : 0 (mauvais), 1 (moyen), et 2 (bien).

jeuxvideo. Le corpus de tests de jeux vidéo comprend 4 231 critiques. Chaque critique comporte une analyse des différents aspects du jeu – graphisme, jouabilité, durée, son, scénario, etc. – et une synthèse globale du jugement. Comme pour le corpus précédent, a été retenue une échelle de 3 niveaux de notes, qui donne les 3 classes 0 (mauvais), 1 (moyen), et 2 (bien).

relectures. Ce corpus comporte 1 484 relectures d'articles scientifiques qui alimentent les décisions de comités de programme de conférences et renvoient des conseils et critiques aux auteurs. L'échelle retenue comporte 3 niveaux de jugement. La classe 0 est attribuée aux relectures qui proposent un rejet de l'article, la classe 1 est attribuée aux relectures qui retrouvent l'acceptation sous condition de modifications majeures ou en séance de posters, et la classe 2 regroupe les acceptations d'articles avec ou sous des modifications mineures. Ce corpus (comme le suivant) a subi un processus préalable d'anonymisation de noms des personnes.

debats. Le corpus des débats parlementaires est composé de 28 832 interventions de députés portant sur des projets de lois examinés par l'Assemblée Nationale. À chaque intervention, est associé le vote de l'intervenant sur la loi discutée. 0 (en faveur) ou 1 (contre).

Les corpus ont été scindés par les organisateurs en deux parties : une partie (environ 60%) des données a été fournie aux participants comme données d'apprentissage afin de mettre au point leurs méthodes, et une autre partie (environ 40%) a été réservée pour les tests proprement dits. Sous peine de disqualification, aucune donnée, en dehors de celles fournies par le comité d'organisation ne pouvait être utilisée. Ceci exclut notamment l'accès aux sites web ou à n'importe quelle autre source d'information. Nous présentons sur le tableau 1, des statistiques brutes (nombre de textes et nombre de mots) des différents corpus. Des exemples portant sur la structure et les détails des corpus, peuvent être consultés dans le site du défi[5].

2.1 Évaluation stricte

Le but du défi consiste à classer chaque texte issu des quatre corpus selon l'avis qui y est exprimé. Positif, négatif ou neutre dans le cas où il y a trois classes. Pour ou contre dans, le cas binaire (le corpus de débats parlementaires contient uniquement deux types d'avis). Intuitivement, la tâche de classer les avis d'opinion des articles scientifiques est la plus difficile des quatre car le corpus afférent contient beaucoup

[4] Par exemple voir le site http ://www.avoir-alire.com

[5] http ://deft07.limsi.fr/corpus-desc.php

Corpus	Textes (A)	Mots (A)	Textes (T)	Mots (T)
aVoiraLire	2 074	490 805	1 386	319 788
jeuxvideo	2 537	1 866 828	1 694	1 223 220
relectures	881	132 083	603	90 979
debats	17 299	2 181 549	11 533	1 383 786

TAB. 1 – Statistiques brutes sur les quatre corpus d'apprentissage (A) et de test (T).

moins d'informations que les trois autres, mais d'autres caractéristiques particulières à chaque corpus ont aussi leur importance. Les algorithmes seront évalués sur des corpus de test (T) avec des caractéristiques semblables à celui d'apprentissage (A) (cf. tableau 1), en calculant le *Fscore* des documents bien classés, moyenné sur tous les corpus :

$$Fscore(\beta) = \frac{(\beta^2 + 1) \times \langle Précision \rangle \times \langle Rappel \rangle}{\beta^2 \times \langle Précision \rangle + \langle Rappel \rangle} \tag{1}$$

où la précision moyenne et le rappel moyen sont calculés comme :

$$\langle Précision \rangle = \frac{\sum_{i=1}^{n} Précision_i}{n} \; ; \; \langle Rappel \rangle = \frac{\sum_{i=1}^{n} Rappel_i}{n} \tag{2}$$

Etant donné pour chaque classe i :

$$Précision_i = \frac{\{\text{Nb de documents correctement attribués à la classe } i\}}{\{\text{Nb de documents attribués à la classe } i\}} \tag{3}$$

$$Rappel_i = \frac{\{\text{Nb de documents correctement attribués à la classe } i\}}{\{\text{Nb de documents appartenant à la classe } i\}} \tag{4}$$

D'après les règles du défi, un document est attribué à la classe d'opinion i si :
- seule la classe i a été attribuée à ce document, sans indice de confiance spécifié ;
- la classe i a été attribuée à ce document avec un meilleur indice de confiance que les autres classes (s'il existe un indice de confiance)

2.2 Indice de confiance pondéré

Un système de classification automatique peut attribuer à un document une distribution de probabilité sur les différentes classes au lieu de lui attribuer une seule classe. L'indice de confiance est cette probabilité pour un document d'appartenir à une classe d'opinion donnée.

Le F-score pondéré par l'indice de confiance a été utilisé, à titre indicatif, pour des comparaisons complémentaires entre les méthodes mises en place par les équipes.

Dans le F-score pondéré, la précision et le rappel pour chaque classe ont été pondérés par l'indice de confiance. Ce qui donne :

$$Précision_i = \frac{\sum_{\text{AttribuéCorrect}_{i=1}}^{\text{NbAttribuéCorrect}_i} \text{Indice\_confiance}_{\text{AttribuéCorrect}_i}}{\sum_{\text{Attribué}_{i=1}}^{\text{NbAttribué}_i} \text{Indice\_confiance}_{\text{Attribué}_i}} \tag{5}$$

$$Rappel_i = \frac{\sum_{\text{AttribuéCorrect}_{i=1}}^{\text{NbAttribuéCorrect}_i} \text{Indice\_confiance}_{\text{AttribuéCorrect}_i}}{\{\text{Nb de documents correctement attribués à la classe } i\}} \tag{6}$$

avec :
- $\text{NbreAttribuéCorrect}_i$: nombre de documents appartenant effectivement à la classe i et auxquels le système a attribué un indice de confiance non nul pour cette classe.
- NbreAttribué_i : nombre de documents attribués auxquels le système a attribué un indice de confiance non nul pour la classe i.

Dans le cadre de DEFT'07, le calcul du F-score retenu par les organisateurs est ensuite calculé à l'aide des formules (1) et (2), du F-score classique modifié (cette réécriture suppose évidemment que β soit égal à 1 de façon à ne privilégier ni précision ni rappel).

3 Représentations de textes

Un même texte peut être représenté par les différents paramètres qu'il est possible d'en extraire. Les représentations les plus courantes sont les mots, POS ou lemmes.

La tâche qui nous occupe consiste à retrouver l'*opinion* exprimée dans les textes. En nous inspirant de l'approche typique de l'analyse des opinions (Hatzivassiloglou & McKeown, 1997), nous utilisons un paramètre de représentation supplémentaire, une étiquette nommée *seed*. Un *seed* est un mot susceptible d'exprimer une polarité positive ou négative (Wilson *et al.*, 2005).

Notre protocole de construction du lexique de *seeds* consiste en deux étapes. Premièrement, une liste de mots polarisés est créée manuellement. Exemple : *aberrant, compliments, discourtois, embêtement, . . .*. Afin de généraliser la liste de mots polarisés obtenue, chaque mot est remplacé par son lemme. Nous obtenons un premier lexique de 565 *seeds*. Deuxièmement, un modèle *BoosTexter* est appris sur les textes représentés en mots. Les mots sélectionnés par ce modèle sont filtrés manuellement, lemmatisés et ajoutés au lexique. Au final, nous obtenons un lexique d'environ 2 000 *seeds*. Une phrase représentée en *seeds* ne contient alors que les lemmes faisant partie de ce lexique.

3.1 Normalisation graphique

Le recours à une étape de prétraitement comme la lemmatisation est motivé par le taux de flexion élevé de la langue française. Néanmoins, dans le problème qui nous occupe, il s'avère utile de ne pas voir disparaître nombre d'informations comme par exemple certains conditionnels ou subjonctifs. Dans une relecture d'article, la présence de propositions comme « *Il aurait été préférable* » ou « *il eût été préférable* » laisse supposer que l'arbitre n'est pas totalement en faveur de l'acceptation du texte qu'il a relu. Pour ne en être privés, nous avons bridé la lemmatisation pour un petit nombre de cas susceptibles de servir de points d'appui lors de la prise de décision. Pour au moins deux systèmes, les textes lemmatisés ont été soumis à une étape que l'on pourrait qualifier de normalisation graphique. Quelque 30 000 règles écrites pour l'occasion ont permis de réunifier les variantes graphiques (essentiellement des noms propres) et de corriger un grand nombre de coquilles. Il est à noter que certaines de ces fautes d'orthographe ont pu être introduites par l'étape de réaccentuation automatique que nous avons appliquée au préalable sur les quatre corpus.

En cas d'ambiguïté, ces récritures sont faites en s'appuyant sur les contextes gauches ou droits (parfois les deux). Exemple : *Thé-Old-Republic* $\Rightarrow$ *the-Old-Republic*.

Ces règles de réécriture avaient aussi pour but de combler certaines lacunes de notre lemmatiseur. Il n'est pas inutile de ramener à leur racine des flexions même peu fréquentes de verbes qui ne se trouvaient pas dans notre dictionnaire (comme *frustrer*, *gâcher*, ou *gonfler*).

Enfin, quelques règles (peu nombreuses) avaient pour mission d'unifier sous une même graphie des variantes sémantiques (par exemple : *tirer-balle-tempe* et *tirer-balle-tête*).

3.2 Agglutination

Les différents exemples donnés ci-dessus font apparaître des regroupements sous la forme d'expressions plus ou moins figées[6]. Celles-ci ont été constituées par application de règles régulières portant sur des couples de mots. Pour leur plus grande partie, les 30 000 règles que nous avons utilisées proviennent d'un simple calcul de collocation effectué selon la méthode du rapport de vraisemblance (Mani & Maybury, 1999). Une autre part non négligeable est issue de listes d'expressions disponibles sur la toile comme celle qui se trouve à l'adresse http ://www.linternaute.com/expression/recherche

Nous y avons ajouté également des proverbes (comme *tirer-son-épingle-jeu*, *mettre-feu-poudre*) extraits de listes se trouvant sur des sites comme http ://www.proverbes.free.fr/rechprov.php

Mais nous sommes conscients que même si nous avons tenté de contrôler au maximum ces ajouts, des expressions comme « *les pieds sur terre* » ou « *un pied à terre* » ont pu être fondues à tort dans une même graphie pied-terre.

Enfin d'autres expressions proches des slogans martelés lors de campagnes électorales, (comme *travailler-plus-pour-gagner-plus* ou *ordre-juste*) nous ont été fournies par une actualité plus brûlante.

[6]Pour l'identification de plusieurs noms propres (noms de jeux vidéo et vedettes du show-bizz) les étudiants et les enfants de l'un des co-auteurs de cet article ont été mis à contribution. Qu'ils en soient ici remerciés.

4 Outils de classification

Les outils de classification de texte peuvent se différencier par la méthode de classification utilisée et par les éléments choisis afin de représenter l'information textuelle (mot, étiquette morpho-syntaxique –*Part Of Speech*, POS–, lemmes, stemmes, sac de mots, sac de n-grammes, longueur de phrase, etc.). Parce qu'il n'y a pas de méthode générique ayant donné la preuve de sa supériorité (dans tous les cas de classification d'information textuelle), nous avons décidé d'utiliser une combinaison de différents classifieurs et de différents éléments de texte. Cette approche nous permet, en outre, d'en déduire facilement les mesures de confiance sur les hypothèses produites lors de l'étiquetage.

Neuf systèmes de décisions ont été implantés utilisant les différents classifieurs présentés ci-bas et les différentes représentations présentées dans la section 3. Ainsi, il s'agit d'obtenir des *avis différents* sur l'étiquetage d'un texte. En outre, le but n'est pas d'optimiser le résultat de chaque classifieur indépendamment mais de les utiliser comme des outils dans leur paramétrage par défaut et d'approcher l'optimum pour la fusion de leurs résultats.

Parce que ces outils sont basés sur des algorithmes de classification différents avec des formats d'entrée différents, ils n'utilisent pas les mêmes éléments d'information afin de caractériser un concept. Une combinaison de plusieurs classifieurs utilisant différentes sources d'information en entrée peut permettre d'obtenir des résultats plus fiables, évaluée par des mesures de confiance basées sur les scores donnés par les classifieurs. Nous ferons ensuite une présentation brève des classifieurs utilisés :

4.1 LIA_SCT

LIA-SCT Béchet *et al.* (2000) est un classifieur basé sur les arbres de décisions sémantiques (SCT-Semantic Classification Tree (Kuhn & De Mori, 1995)).

Il suit le principe d'un arbre de décision : à chaque nœud de l'arbre une question est posée qui subdivise l'ensemble de classification dans les nœuds fils jusqu'à la répartition finale de tous les éléments dans les feuilles de l'arbre.

La nouveauté des SCT réside dans la construction des questions qui se fait à partir d'un ensemble d'expressions régulières basées sur une séquence de composants. Leur ordre dans le vecteur d'entrée a donc une importance. De plus, chaque composant peut se définir suivant différents niveaux d'abstraction (mots et POS par exemple) et d'autres paramètres plus globaux peuvent également intégrer le vecteur (nombre de mots du document par exemple).

Lorsque le SCT est construit, il prend des décisions sur la base de règles de classification statistique apprises sur ces expressions régulières. Lorsqu'un texte est classé dans une feuille, il est alors associé aux hypothèses conceptuelles de cette feuille selon leur probabilité.

LIA-SCT est utilisé dans le Système 6, où les textes sont représentés en lemmes.

4.2 BoosTexter

BoosTexter Schapire & Singer (2000) est un classifieur à large marge basé sur l'algorithme de boosting : *Adaboost* Freund & Schapire (1996). Le but de cet algorithme est d'améliorer la précision des règles de classification en combinant plusieurs hypothèses dites *faibles* ou peu précises.

Une hypothèse faible est obtenue à chaque itération de l'algorithme de boosting qui travaille en re-pondérant de façon répétitive les exemples dans le jeu d'entraînement et en ré-exécutant l'algorithme d'apprentissage précisément sur ces données re-pondérées. Cela permet au système d'apprentissage faible de se concentrer sur les exemples les plus compliqués (ou problématiques).

L'algorithme de boosting obtient ainsi un ensemble d'hypothèses faibles qui sont ensuite combinées en une seule règle de classification qui est un vote pondéré des hypothèses faibles et qui permet d'obtenir un score final pour chaque constituant de la liste des concepts.

Les composants du vecteur d'entrée sont passés selon la technique du sac de mots et les éléments choisis par les classifieurs simples sont alors des n-grammes sur ces composants ;

Quatre de nos systèmes utilisent le classifieur *BoosTexter* :

- Système 1 (LIA_BOOST_BASELINE) : la représentation d'un document se fait en mots. BoosTexter est appliqué en mode 3-grammes ;
- Système 2 (LIA_BOOST_BASESEED) : chaque document est représenté en seeds, chaque seed est pondéré par son nombre d'occurrences, en mode uni-gramme ;

– Système 3 (LIA_BOOST_SEED) : chaque document est représenté par les mots et également par les seeds toujours pondérés par leur nombre d'occurrences, en mode uni-grammes ;
– Système 4 (LIA_BOOST_CHUNK) : L'outil *LIA-TAGG*[7] est utilisé pour découper le document en un ensemble de syntagmes lemmatisés. Chaque syntagme contenant un *seed* ainsi que le syntagme précédent et suivant sont retenus comme représentation. Les autres syntagmes sont rejetés de la représentation du document. *BoosTexter* est appliqué en mode 3-grammes sur cette représentation.

4.3 SVMTorch

SVMTorch Collobert et al. *(2002)* est un classifieur basé sur les machines à support vectoriel (Support Vector Machines –*SVM*–) proposées par Vapnik (Vapnik, 1982, 1995).

Les SVM permettent de construire un classifieur à valeurs réelles qui découpe le problème de classification en deux sous-problèmes : transformation non-linéaire des entrées et choix d'une séparation linéaire *optimale*. Les données sont d'abord projetées dans un espace de grande dimension où elles sont linéairement séparables selon une transformation basée sur un noyau linéaire, polynomial ou gaussien. Puis dans cet espace transformé, les classes sont séparées par des classifieurs linéaires qui déterminent un hyperplan séparant correctement toutes les données et maximisant la *marge*, la distance du point le plus proche à l'hyperplan.

Elles offrent, en particulier, une bonne approximation du principe de minimisation du risque structurel (*i.e* : trouver une hypothèse h pour laquelle la probabilité que h soit fausse sur un exemple non-vu et extrait aléatoirement du corpus de test soit minimale).

Dans nos expériences, la technique la plus simple du sac de mots est utilisée : un document est représenté comme un vecteur dont chaque composante correspond à une entrée du lexique de l'application et chaque composante a pour valeur le nombre d'occurrences de l'entrée lexicale correspondant dans le texte.

Le système (LIA_NATH_TORCH) est obtenu avec *SVMTorch*. Le vecteur d'entrée est représenté par le lexique des *seeds*.

4.4 Timble

Timble Daelemans et al. *(2004)* est un classifieur implémentant plusieurs techniques de *Memory-Based Learning –MBL–*. Ces techniques, descendantes directes de l'approche classique des k-plus-proches-voisins (*K Nearest Neighbor k-NN*) appliquée à la classification, ont prouvé leur efficacité dans un large nombre de tâches de traitement du langage naturel.

Le paramétrage par défaut de *TiMBL* est un algorithme *MLB* qui construit une base de données d'*instances de base* lors de la phase d'entraînement. Comme pour SVM-Torch, une instance est un vecteur de taille fixe dont les composantes sont les entrées du lexique ayant pour valeur le nombre d'occurrences dans le document. À cela s'ajoute une composante indiquant quelle est la classe à associer à ce vecteur de paires { caractéristique-valeur }. Lorsque la base de données est construite, une nouvelle instance est classée par comparaison avec toutes les instances existantes dans la base, en calculant la distance de celle-ci par rapport à chaque instance en mémoire. Par défaut, *TiMBL* résout l'algorithme *1-NN* avec la métrique *Overlap Metric* qui compte simplement le nombre de composantes ayant une valeur différente dans chacun des 2 vecteurs comparés. Cette métrique est améliorée par l'*Information Gain –IG–* introduit par Quinlan (1986, 1993) qui permet de mesurer la pertinence de chaque composante du vecteur.

Le système 7 (LIA_TIMBLE) est formé de l'outil *TiMBL* appliqué sur les *seeds*.

4.5 Modélisation probabiliste uni-lemme (LIA_JUAN)

Nous avons voulu simplifier au maximum un classifieur et savoir si les modèles n-grammes avec $n > 1$ apportent vraimment des éléments discriminants. Nous avons décidé d'implanter un classifieur incorporant des techniques élementaires sur les n-lemmes. Ces techniques, descendantes directes de l'approche probabiliste (Mani & Maybury, 1999) appliquées à la classification de texte, ont prouvé leur efficacité dans le défi précédent (El-Bèze *et al.*, 2005).

Les textes ont été filtrés légèrement (afin de garder notamment des petites tournures comme la voix passive, les formes interrogatives ou exclamatives), un processus d'agregation de mots composés (via un

[7] http ://www.lia.univ-avignon.fr/chercheurs/bechet/download_fred.html

dictionnaire simple), puis regroupés dans des mots de la même famille. Ce processus comporte une lemmatisation particulière. Ainsi, des mots tels que : *chantaient*, *chant*, *chantons*, et même *chanteurs* et *chanteuses* seront ramenés au lemme « chanter ». Nous avons limité notre modèle à $n = 1$, soit des uni-lemmes, ce qui nous évite de calculer beaucoup de coefficients de lissage. Nous avons transformé donc chaque document en un sac d'uni-lemmes. Puis nous avons calculé la classe d'appartenance d'un document comme :

$$P_t(w) \approx \prod_i \lambda_1 P_t(w_i) + \lambda_0 U_0 \tag{7}$$

Nous avons appliqué ce modèle d'uni-lemmes à tous les corpus, sans faire d'autres traitements particulières.

4.6 LIA_MARC

Nous avons envisagé ici de recourir à une modélisation somme toute classique en théorie de l'information, tout en cherchant à y intégrer quelques unes des spécificités du problème. La formulation que nous avons retenue initialement se rapproche de celle que nous avions employée lors d'un précédent DEFT (El-Bèze *et al.*, 2005).

$$\widetilde{t} = Arg_t \max P(t) \times P(w|t) = Arg_t \max P(t) \times P_t(w) \tag{8}$$

L'étiquette t pouvant prendre ses valeurs dans un ensemble de cardinal réduit à 2 ou 3 éléments [0-1] ou [0-2], a priori le problème pourrait paraître simple, et la quantité des données fournies suffisante pour bien apprendre les modèles. Même si le vocabulaire propre aux différents corpus n'est pas si grand (entre 9 000 mots différents pour le plus petit corpus et 50 000 pour le plus grand), il reste que certaines entrées sont assez peu représentées. Aussi dans la lignée de ce qui se fait habituellement pour calculer la valeur du second terme de l'équation 8 nous avons opté pour un lissage de modèles n-lemmes (n allant de 0 à 3).

$$P_t(w) \approx \prod_i \lambda_3 P_t(w_i|w_{i-2}w_{i-1}) + \lambda_2 P_t(w_i|w_{i-1}) + \lambda_1 P_t(w_i) + \lambda_0 U_0 \tag{9}$$

L'originalité de la modélisation que nous nous sommes proposés d'employer dans le cadre de DEFT'07 réside essentiellement dans les aspects discriminants du modèle. Par manque de place, il ne nous est pas possible de détailler ici les différentes caractéristiques de cette nouvelle approche. Cela sera fait lors d'une publication ultérieure. Mais nous pouvons en dire au moins quelques mots. Lors de l'apprentissage, les comptes des n-lemmes sont rééchelonnés en proportion de leur pouvoir discriminant. Ce dernier est estimé selon un point de vue complémentaire au critère d'impureté de Gini selon la formule suivante.

$$G(w,h) \approx \sum_i P_t^2(t|w,h) \tag{10}$$

Les entrées w et leurs contextes gauches h qui ne sont apparus qu'avec une étiquette donnée t et pas une autre, ont un pouvoir discriminant égal à 1. Ce critère a été lissé avec un sous-critère G' permettant de favoriser (certes dans une moindre mesure que G) les couples (w,h) qui n'apparaissent que dans 2 étiquettes sur 3. Notons tout d'abord que l'emploi de tels critères discriminants est une façon de pallier le fait que l'apprentissage par recherche d'un maximum de vraisemblance ne correspond pas vraiment aux données du problème. Deuxièmement, il est aisé de comprendre combien un regroupement massif des entrées lexicales par le biais des collocations (cf. section 3) peut avoir un effet déterminant sur le nombre des événements à coefficient discriminant élevé. Ces deux remarques visent à souligner que sur ce point particulier le fameux croissement entre methode symbolique et numérique a son mot à dire.

En dernier lieu, nous avons aussi adapté le calcul du premier terme $P(t)$ de l'équation 8 en combinant la fréquence relative de l'étiquette t avec la probabilité de cette même étiquette sachant la longueur du texte traité. Pour cela, nous avons eu recours à la loi Normale.

5 Résultats et discussion

Afin de tester nos méthodes et de règler leurs paramètres, nous avons decidé de scinder l'ensemble d'apprentissage (A) de chaque corpus en cinq sous-ensembles approximativement de la même taille (en nombre de textes à traiter). La procédure d'apprentissage a été la suivante : nous avons concaténé quatre des cinq

sous-ensembles et gardé le cinquième pour le test. Les ensembles ainsi concaténés seront appelés dorénavant, ensembles de devéloppement (D) et le restant, ensemble de validation (V). Cinq expériences par corpus ont été ainsi effectuées tour à tour.

Nous allons présenter nos résultats en deux items : d'abord ceux obtenus sur les ensembles de devéloppement (D) et validation (V) où nous avons paramétré nos systèmes, et ensuite les résultats sur les données de test (T) en appliquant les algorithmes. On notera que les résultats different quelque peu des résultats officiels, car depuis la cloture du défi notre système à évolué.

5.1 Évaluation sur les corpus de devéloppement (D) et de validation (V)

Le decoupage de chaque corpus en cinq sous-ensembles de devéloppement (D) est le fruit d'un tirage aléatoire. Ce découpage permet, selon nous, d'éviter de régler les algorithmes sur un seul ensemble d'apprentissage (et un autre seul de test), ce qui pourrait conduire à deux travers, le biais expérimental et/ou le phenomène de surapprentissage.

Nous présentons dans les tableaux 2 (**aVoiraLire**), 3 (**jeuxvideo**), 4 (**relectures**) et 5 (**debats**) des statistiques des sous-ensembles de devéloppement (T) et validation (V) en fonction de leurs classes pour chacun des corpus.

Corpus **aVoiraLire**							
	Total	Classe 0		Classe 1		Classe 2	
Ensembles (D)	Textes	Textes	%	Textes	%	Textes	%
1	1 660	231	13,92	486	29,27	943	56,81
2	1 659	249	15,01	494	29,78	916	55,21
3	1 659	244	14,71	498	30,02	917	55,27
4	1 659	248	14,95	490	29,54	921	55,51
5	1 659	264	15,91	492	29,66	903	54,43
Ensembles (V)	textes	Textes	%	Textes	%	Textes	%
1	414	45	10,84	123	29,64	247	59,52
2	415	61	14,70	123	30,12	229	55,18
3	415	65	15,66	117	28,19	233	56,14
4	415	60	14,46	121	29,16	234	56,39
5	415	78	18,84	129	31,16	207	50,00

TAB. 2 – Statistiques par classe sur les ensembles de devéloppement (D) et de validation (V), **aVoiraLire**.

Corpus **jeuxvideo**							
	Total	Classe 0		Classe 1		Classe 2	
Ensembles (D)	textes	Textes	%	Textes	%	Textes	%
1	2 032	412	20,28	917	45,13	703	34,59
2	2 029	395	19,47	905	44,60	729	35,93
3	2 029	350	17,25	951	46,87	728	35,88
4	2 029	467	23,02	946	46,62	616	30,36
5	2 029	364	17,94	945	46,57	720	35,48
Ensembles (V)	textes	Textes	%	Textes	%	Textes	%
1	505	133	26,18	221	43,50	154	30,31
2	508	30	5,91	220	43,31	258	50,79
3	508	147	28,94	215	42,32	146	28,74
4	508	102	20,08	261	51,38	145	28,54
5	508	85	16,83	249	49,31	171	33,86

TAB. 3 – Statistiques par classe sur les ensembles de devéloppement (D) et de validation (V), **jeuxvideo**.

Sur la figure 1, nous montrons le F-score du système de fusion sur les quatre corpus (V). L'apprentissage a été fait sur les ensembles de devéloppement, et le F-score est calculé sur les cinq ensembles de validation

Corpus **relectures**							
	Total	Classe 0		Classe 1		Classe 2	
Ensembles (D)	textes	Textes	%	Textes	%	Textes	%
1	708	151	21,33	262	37,01	295	41,67
2	704	179	25,43	208	29,55	317	45,03
3	704	184	26,14	200	28,41	320	45,46
4	704	206	29,26	214	30,40	284	40,34
5	704	188	26,70	228	32,39	288	40,91
Ensembles (V)	textes	Textes	%	Textes	%	Textes	%
1	173	39	22,03	50	28,25	88	49,72
2	177	21	11,86	64	36,16	92	51,98
3	177	43	24,29	78	44,07	56	31,64
4	177	48	27,12	70	39,55	59	33,33
5	177	76	43,93	16	9,25	81	46,82

TAB. 4 – Statistiques par classe sur les ensembles de devéloppement (D) et de validation (V), **relectures**.

Corpus **debats**					
	Total	Classe 0		Classe 1	
Ensembles (D)	textes	Textes	%	Textes	%
1	13 840	7 893	57,03	5 947	42,97
2	13 839	8 525	61,60	5 314	38,40
3	13 839	8 710	62,94	5 129	37,06
4	13 839	7 587	54,82	4 890	35,33
5	13 839	7 913	57,18	5 926	42,82
Ensembles (V)	textes	Textes	%	Textes	%
1	3 459	2 487	71,88	973	28,12
2	3 460	1 841	53,21	1 619	46,79
3	3 460	1 690	48,84	1 770	51,16
4	3 460	1 875	58,39	1 585	56,54
5	3 460	2 507	72,48	952	27,52

TAB. 5 – Statistiques par classe sur les ensembles de devéloppement (D) et de validation (V), **debats**.

(V). On peut constater que le corpus de relectures d'articles scientifiques est le plus difficile à traiter. En effet, ce corpus comporte le plus petit nombre de textes ($\approx$ 704 en devéloppement, $\approx$ 177 en validation). Il est aussi dur à classer étant donnée des particularités propres à ce corpus que nos avons detecté : les arbitres corrigent souvent le texte des articles à la volée (directement dans leurs commentaires), ce qui est une introduction de bruit. Nous y reviendrons lors de la discussion de nos résultats.

Corpus	Précision	Rappel	F-score	Correctes	Total
aVoiraLire (V)	0,6419	0,5678	0,6026	1 385	2 074
jeuxvideo (V)	0,8005	0,7730	0,7865	2 005	2 537
relectures (V)	0,5586	0,5452	0,5518	510	881
debats (V)	0,7265	0,7079	0,7171	12 761	17 299

TAB. 6 – Performances en Précision, Rappel et F-score obtenus par notre méthode de fusion, sur les quatre corpus de validation (V).

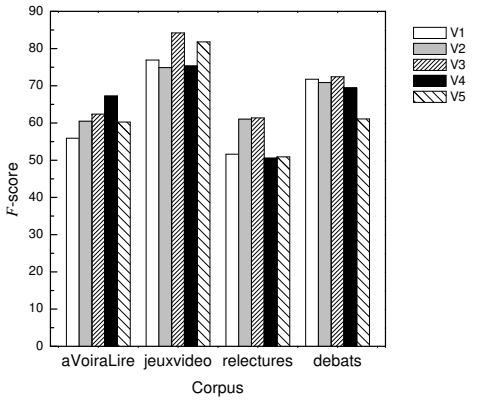


FIG. 1 – F-score obtenu par l'algorithme de fusion sur les cinq ensembles de validation (V). Nous affichons des résultats regroupés par corpus.

5.2 Évaluation sur les corpus de test

Nous avons défini l'ensemble d'apprentissage $\{A_j\} = \{D_j\} \cup \{V_j\}$; $j =$ {**aVoiraLire**, **jeuxvideo**, **relectures**, **debats**}. Dans le tableau 7, nous montrons les statistiques par classe pour les quatre corpus de test (T) et d'apprentissage (A). On peut constater que la distribution des données en apprentissage et en test est très homogène, ce qui en principe, facilite la tâche de n'importe quel classifieur.

Corpus de	Total	Classe 0		Classe 1		Classe 2	
test	textes	Textes	%	Textes	%	Textes	%
aVoiraLire (T)	1 386	207	14,94	411	29,65	768	55,41
jeuxvideo (T)	1 694	332	19,60	779	45,99	583	34,42
relectures (T)	603	157	26,04	190	31,51	256	42,45
debats (T)	11 533	6 572	56,98	4 961	43,02	⊘	⊘
Corpus	Total	Classe 0		Classe 1		Classe 2	
d'apprentissage	textes	Textes	%	Textes	%	Textes	%
aVoiraLire (A)	2 074	309	14,90	615	29,65	1150	55,45
jeuxvideo (A)	2 537	497	19,59	1166	45,96	874	34,45
relectures (A)	881	227	25,77	278	31,55	376	42,68
debats (A)	17 299	10 400	60,12	6 899	39,88	⊘	⊘

TAB. 7 – Statistiques par classe sur les quatre corpus d'apprentissage (A) et de test (T).

Corpus	Précision	Rappel	F-score	Correctes	Total
aVoiraLire (T)	0,6540	0,5590	0,6028	931	1 386
jeuxvideo (T)	0,8114	0,7555	0,7824	1 333	1 694
relectures (T)	0,5689	0,5565	0,5626	353	603
debats (T)	0,7307	0,7096	0,7200	8 403	11 533

TAB. 8 – Performances en Précision, Rappel et F-score obtenus par notre méthode de fusion, sur les quatre corpus de test (T).

En figure 2, nous montrons les performances en F-score de chacun de nos classifieurs, ainsi que leurs moyennes sur les quatre ensembles de test. On constate que les classifieurs LIA_TIMBLE et LIA_SCT ont les perfomances les plus basses.

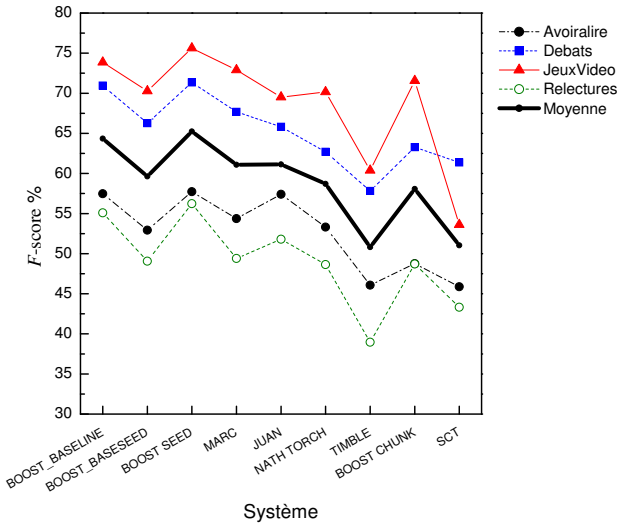


FIG. 2 – F-score de chacune des méthodes sur les quatre corpus de test.

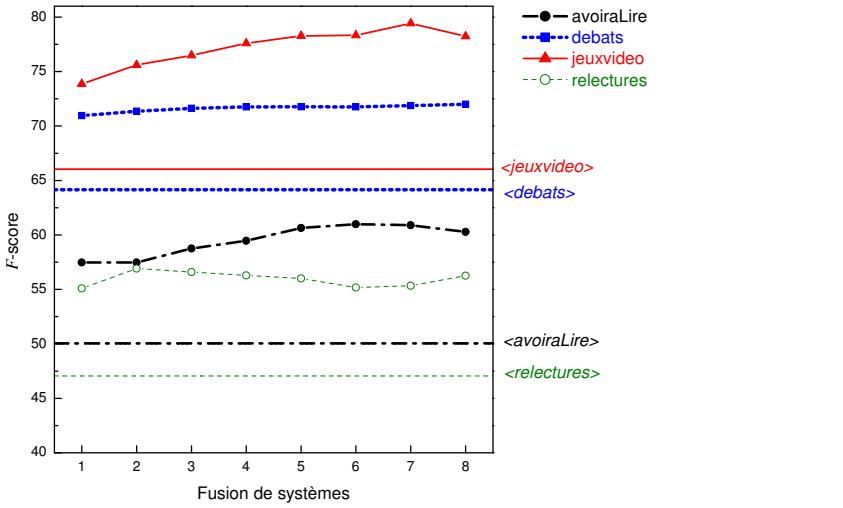


FIG. 3 – F-score de la fusion suivant nos 9 méthodes ajoutées. Nous affichons aussi les moyennes de l'ensemble des soumissions des participants (voir table 9) : nous nous situons au-dessus des moyennes, tous corpus confondus.

En figure 3, nous montrons les performances en F-score d'une fusion « incrémentale » des méthodes ajoutées. Cependant, l'ordre affiché n'a strictement aucun impact dans la fusion finale : il a été choisi uniquement pour mieux illustrer les résultats. On peut voir que nos résultats se placent bien au dessus de la moyenne des équipes participantes dans le défi DEFT'07 (voir la table 9 de l'Annexe), tous corpus confondus.

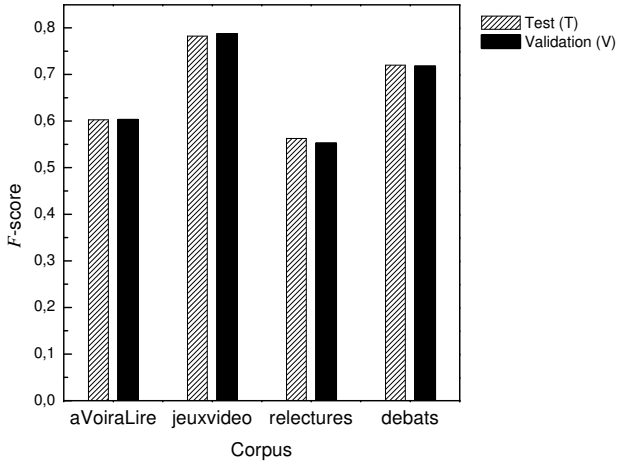


FIG. 4 – Comparaison du F-score de l'ensemble de validation (V) vs. celui de test (T) pour chacun des corpus, obtenu par notre système de fusion.

Sur la figure 4 nous montrons une comparaison du F-score de l'ensemble de validation (V) vs. celui de test (T), sur les quatre corpus. On peut constater la remarquable coïncidence entre les deux, ce qui signifie que notre stratégie d'apprentissage et de validation sur cinq sous-ensembles et de fusion de plusieurs classifieurs a bien fonctionné.

5.3 Discussion

Nous avons constaté que l'utilisation de collocations et réécriture (cf. section 3.1) permet d'augmenter les perfomances des méthodes. Par exemple, avec la méthode probabiliste à base d'uni-lemmes sur le corpus de validation nous sommes passés de 1 285 à 1 310 bien classés (F=57,41 → 58,89) dans le corpus **aVoiraLire**, de 1801 à 1916 (F=70,77 → 75,15) en **jeuxvideo**, de 445 à 455 (F=48,36 → 49,53) en **relectures** et de 10 364 à 11 893 (F=62,21 → 67,12) en **debats**. Dans le corpus de test les gains sont aussi non négligeables. Nous sommes passés en **aVoirAlire** de 863 à 860 (F=57,40 → 56.32) ; de 7530 à 7635 (F=65,82 → 66,88) en **debats** ; de 1169 à 1205 (F=69,51 → 71,48) en **jeuxvideo** et de 317 à 313 (F=51,81 → 52,04) en **relectures**. Ceci confirme l'hypothèse que la réécriture aide à mieux capturer la polarité des avis.

Nous avons realisé une analyse *post-mortem* de nos résultats. Nous présentons ci-bas, quelques exemples de notices qui ont été mal classés par nos systèmes. Nous avons delibérement gardé les notices dans leur état, même avec les fautes d'ortographe. En particulier, nous avons décidé de montrer majoritairement, des avis d'opinion venant du corpus de relectures d'articles scientifiques, corpus qui avait posé plus de difficultés aux algorithmes (F-score plus faible) que les autres. Par exemple, considérez la notice 3 :36 (**relectures**) :

> ***3 :2 relectures***
> *L'idée d'appliquer les méthodes de classification pour définir des classes homogènes de pages web est assez originale par contre, la méthodologie appliquée est classique. Je recommande donc un « weak accept » pour cet article.*

Nos systèmes l'ont classé 1 (accepté avec des modifications majeures), et après une lecture directe, on pourrait en déduire que la classe est 1 alors que la référence est 2 (accepté).

Notice 3 :2 (**relectures**). L'article a été accepté mais notre système le rejette. Il comporte beaucoup d'expressions négatives comme : « *parties de l'article me paraissent déséquilibrées* », « *Le travail me paraît inachevé* », »*la nouvelle méthode proposée pose des problèmes complexes ... qui ne sont pas traités dans ce papier* », cependant il a été accepté.

> ***3 :2 relectures***
> ***Les différentes parties de l'article me paraissent déséquilibrées****. Les auteurs présentent d'abord un état de l'art dans le domaine de la visualisation des connaissances dans les systèmes de gestion de connaissances. Ils décrivent ensuite le serveur <anonyme /> et sa représentation des connaissances sous forme d'arbre en section 3 et une partie de la section 4. L'approche proposée par les auteurs (représentation par graphes n'est présentée qu'en 4.2 sur moins d'une page).* ***Les problèmes posés par cette méthode sont survolés par les auteurs****, ils font référence aux différents papiers traitant de ces problèmes et* ***n'exposent pas du tout*** *les heuristiques choisies dans leurs approches.* ***Le travail me paraît inachevé****, et* ***la nouvelle méthode proposée pose des problèmes complexes*** *au niveau de la construction de ce graphe* ***qui ne sont pas traités dans ce papier****.*

Notice 3 :6 (**relectures**). L'article a été accepté, alors que notre système le rejette. L'article arbitré est peut-être trop court, mais la relecture qui le concerne, elle l'est aussi :

> ***3.6 relectures***
> *Article trop court pour pouvoir être jugé. Je suggère de le mettre en POster si cela est prévu.*

Pour la notice 3 :9 (**relectures**) on décèle le même problème : l'article est accepté alors que le système le rejette. Constatons que l'arbitre focalise uniquement sur des remarques de forme :

> ***3 :9 relectures***
> *Question : comment est construit le réseau bayesien ? Un peu bref ici... Remarques de forme : page 2, 4ème ligne, « comprend » 5ème ligne : "annotées" ou "annoté" page 3 : revoir la phrase confuse précédant le tableau dernière ligne, répétition de "permet" page 5 : 7ème ligne accorder "diagnostiqué" et "visé" avec "états" ou avec "connaissances"*

Pour le texte de la notice 3 :567 (**relectures**), l'article en question est rejeté alors que le système l'accepte. Phrases encourageantes au début finisent par être mitigées. Beaucoup d'expressions positives (« *bien organisé* », « *facile à suivre* », « *bibliographie est plutôt complète* », « *solution proposée et interessante et originale* », « *La soumission d'une nouvelle version ... sera intéressante* ») n'arrivent pas a renverser le rejet.

> ***3 :567 relectures***
> *Commentaire : L'article est* ***plutôt bien organisé*** *(malgré de trop nombreux chapitres), le cheminement de* ***la logique est facile à suivre****. Cependant il y a de trop nombreuses fautes de français ainsi que d'anglais dans le résumé.* ***La bibliographie est plutôt complète****.* ***La solution proposée et interessante et originale****, cependant des notions semblent mal maîtrisées. Ainsi dans la section 8, la phrase «Cette convergence ne vient pas des algorithmes génétiques de manière intrinsèque, mais de l'astuce algorithmique visant à conserver systématiquement le meilleur individu dans la population » démontre une incompréhension du fonctionnement même d'un algorithme génétique. La soumission d'une nouvelle version modifiée de cet article, présentant également les premiers résultats obtenus avec le prototype à venir* ***sera intéressante pour la communauté****.*
> *Références : Originalité : Importance : Exactitude : Rédaction :*

Pour finir, une notice du corpus de films, livres et spectacles : le texte 1 :10 du corpus **aVoiraLire**. Malgré des expressions telles que : « *...est un événement* », « *Agréable surprise* » ou encore « *l'image d'une cohérence artistique retrouvée* », la notice reste difficile à classer. Évidemment notre système se trompe. Mettons à notre tour le lecteur au défi de trouver la véritable classe[8].

[8] Si vous êtiez tenté de le mettre en classe 2 (bien), sachez que la classe véritable est la 1 (moyen).

aVoiraLire 1 :10
Depuis trente-six ans, chaque nouvelle production de David Bowie est un événement. Heathen , ne fait pas exception à cette règle. On reconnaît instantanément la patte de son vieux compère Tony Visconti. La voix de Bowie est mise en avant. Agréable surprise, surtout qu'elle n'a rien perdu depuis ses débuts. Là, commence le voyage. Ambiance, mélange dosé des instruments. Dès l'ouverture de l'album avec Sunday , un sentiment étrange nous envahit. Comme si Bowie venait de rentrer d'un voyage expérimental au coeur même de la musique. Retour aux sources. L'ensemble du disque est rythmé par cette pulsation dont le duo a le secret. Le tout saupoudré de quelques pincées d'électronique. Le groupe est réduit au minimum. Outre Bowie en chef d'orchestre et Visconti, David Torn ponctue les compositions de ses guitares aventureuses et Matt Chamberlain apporte de l'âme à la rythmique. Un quatuor à cordes fait une apparition, comme Pete Townshend (The Who) ou Dave Grohl (ex-batteur de Nirvana). Avec trois reprises réarrangées et neuf compositions originales, le 25e album de Bowie est à l'image d'une cohérence artistique retrouvée.

6 Conclusion et perspectives

La classification de textes en fonction des tendances d'opinion qu'ils expriment reste une tâche très difficile, même pour une personne. La lecture directe ne suffit pas toujours pour se forger un avis et privilégier une classification pa rapport à une autre. Nous avons décidé d'utiliser des approches de représentation numériques et probabilistes, afin de rester aussi indépendant que possible des sujets traités. Nos méthodes ont fait leur preuve. Nous avons confirmé l'hypothèse que la réécriture (normalisation graphique) et les collocations aident à capturer le sens des avis. Ceci se traduit par un gain de performances. Nous avons présenté une stratégie de fusion assez simple de méthodes. Celle ci s'est avérée robuste et performante. Nos F-scores sont au-dessus des moyennes sur les quatre corpus de test, notamment sur celui de **jeuxvideo**. La stratégie de fusion a montré des résultats supérieurs à n'importe laquelle des méthodes individuelles. La dégradation en précision et rappel reste faible, même si nous n'avons pas écarté de la fusion des méthodes peut-être moins adaptées à cette problématique. La fusion est donc une façon robuste de combiner plusieurs classifieurs. Il faut souligner la remarquable équivalence entre les résultats obtenus lors de l'apprentissage et la prédiction sur les ensembles de test : à un point près de différence.

Le module de fusion n'a pas été optimisé, un même poids étant attribué au vote de chaque système. Ceci nous ouvre facilement la voie à une possible amélioration.

Le corpus de relectures des articles scientifiques reste de loin le plus difficile à traiter. Nous avions déjà avancé l'hypothèse qu'en raison du faible nombre de notices, il serait difficile à classer. Il y a d'autres facteurs qui interviennent également. Les relectures souvent comportent, dans le corps du texte, des corrections adressées aux auteurs. Ceci vient bruiter nos classifieurs. Les relectures sont parfois trop courtes, ou bien elles ont été redigées par des arbitres non francophones (encore une source de bruit) ou contienent beaucoup d'anglicismes (*weak acceptation, boosting, support vector,...*). Un autre facteur, peut-être plus subtil : un article peut être lu par plusieurs arbitres (deux, trois voire plus) qui émettent des avis opposés. Dans une situation où les arbitres A et B acceptent l'article et un troisième C le refuse, (comme cela risque d'être le cas du présent article) normalement l'article doit être accepté. Donc, dans le corpus **relectures**, l'avis de C sera asimilé à la classe acceptée, et cela malgré son avis négatif.

Remerciements

Nous remercions Thomas Heitz (LRI-I&A) et Martine Hurault-Plantet (LIMSI-LIR) ainsi que le comité d'organisation de la campagne de DEFT'07. Tous nos remerciements également à tous les relecteurs de cet article que nous avons choisi d'anonymiser.

Annexe : résultats des participants sur DEFT'07

Sur le tableau 9 nous montrons les moyennes et les écarts-types de l'ensemble des soumissions à DEFT'07 pour les quatre corpus, d'après les résultats fournis par les organisateurs du défi. Ces résultats ont été obtenus avec le F-score strict (voir équation 3).

Corpus	Précision	Rappel	F-score
aVoiraLire (T)	0,5276 ± 0,0982	0,4829 ± 0,0683	0,5004 ± 0,0668
jeuxvideo (T)	0,6925 ± 0,0996	0,6367 ± 0,0921	0,6604 ± 0,0864
relectures (T)	0,4804 ± 0,0490	0,4617 ± 0,0477	0,4706 ± 0,0468
debats (T)	0,6545 ± 0,0564	0,6298 ± 0,0645	0,6416 ± 0,0594

TAB. 9 – Moyennes et écarts-types de l'ensemble des soumissions des participants pour les quatre corpus.

Références

AZÉ J., HEITZ T., MELA A., MEZAOUR A.-D., PEINL P. & ROCHE M. (2006). Préparation de DEFT'06 (DÉfi Fouille de Textes). In *Proc. of Atelier DEFT'06*, volume 2.

AZÉ J. & ROCHE M. (2005). Présentation de l'atelier DEFT'05. In *Proc. of TALN 2005 - Atelier DEFT'05*, volume 2, p. 99–111.

BÉCHET F., NASR A. & GENET F. (2000). Tagging unknown proper names using decision trees. In *38th Annual Meeting of the Association for Computational Linguistics, Hong-Kong, China*, p. 77–84.

COLLOBERT R., BENGIO S. & MARIÉTHOZ J. (2002). Torch : a modular machine learning software library. In *Technical Report IDIAP-RR02-46, IDIAP*.

DAELEMANS W., ZAVREL J., VAN DER SLOOT K. & VAN DEN BOSCH A. (2004). Timbl : Tilburg memory based learner, version 5.1, reference guide. *ILK Research Group Technical Report Series*, p. 04–02.

EL-BÈZE M., TORRES-MORENO J.-M. & BÉCHET F. (2005). Peut-on rendre automatiquement à César ce qui lui appartient ? Application au jeu du Chirand-Mitterrac. In *Actes de la conférence "Traitement Automatique des Langues Naturelles" (TALN 2005) - Atelier DEFT'05*, volume 2, p. 125–134.

FREUND Y. & SCHAPIRE R. E. (1996). Experiments with a new boosting algorithm. In *Thirteenth International Conference on Machine Learning*, p. 148–156.

HATZIVASSILOGLOU V. & MCKEOWN K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, p. 174–181, Morristown, NJ, USA : Association for Computational Linguistics.

KUHN R. & DE MORI R. (1995). The application of semantic classification trees to natural language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(5), 449–460.

MANI I. & MAYBURY M. T. (1999). *Advances in Automatic Text Summarization*. MIT Press.

QUINLAN J. (1986). Induction of decision trees. *Machine Learning*, **1**(1), 81–106.

QUINLAN J. (1993). *C4. 5 : Programs for Machine Learning*. Morgan Kaufmann.

SCHAPIRE R. E. & SINGER Y. (2000). BoosTexter : A boosting-based system for text categorization. *Machine Learning*, **39**, 135–168.

VAPNIK V. N. (1982). *Estimation of Dependences Based on Empirical Data*. New York, USA : Springer-Verlag Inc.

VAPNIK V. N. (1995). *The nature of statistical learning theory*. New York, USA : Springer-Verlag Inc.

WILSON T., WIEBE J. & HOFFMANN P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP*, p. 347–354, Vancouver, Canada.

Approches statistiques et LSA

An LSA Approach in DEFT'07 Contest

Murat AHAT[1], Wolfgang LENHARD[2], Herbert BAIER[2],

Vigile HOAREAU[1], Sandra JHEAN-LAROSE[1] et Guy DENHIÈRE[1]

1Équipe CHArt « Cognition Humaine et Artificielle » - E.A. 4004

EPHE - CNRS
41 Rue Gay Lussac
75005 Paris
France

murat.ahat@google.com

2 Department of Psychology
Universität Würzburg
Röntgenring 10
97070 Würzburg
Germany

wolfgang.lenhard@uni-wuerzburg.de

1 Introduction

In this DEFT'07 contest, we are presented with texts grouped in four categories which include judgments as well. The task is to determine to which judgment the texts belong within a category, while the category of the text is already given. Simply speaking, the task is Document Classification. To achieve this goal, we have built a simple automatic/semi automatic process based on Latent Semantic Analysis (LSA), which produces encouraging results. This process takes into account what the psychology of comprehension has called the mental representation of a document to propose an algorithm for categorization. This approach is based on the achievements accomplished by cognitive psychology in text comprehension and its simulation (Denhière, Lemaire, Bellissens et Jhean-Larose, 2005). Following sections will describe how we have built the automatic process, and the results we have obtained. Finally, the results are discussed.

2 Automated Document Classification

Automated processing is vital in classification of a large number of texts. Without an automated process, classification of texts would be a tedious and boring job. We have based our automated process on LSA - Latent Semantic Analysis. LSA is a statistical technique from the field of natural language processing (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990). It permits to extract the relations between words based on their common occurrences in texts. Its procedure is completely statistical in nature. That means the word meanings reflected by the cooccurrences are extracted completely without any specification of rules or dictionaries.

2.1 Extracting the Latent Semantic Content of Written Language

LSA is based on text corpora with each single text usually being split into fragments, as for example paragraphs (Wiemer-Hastings, 1999). The information stored in the corpora can formally be represented by building a frequency matrix. The columns of this matrix contain the fragment

Actes du troisième défi fouille de texte, DEFT2007, Grenoble, France, 3 juillet 2007.
Proceedings of the Third DEFT Workshop, DEFT2007, Grenoble, France, 3rd July 2007.
Pages 147-157

indices and the rows contain the different words. Each cell holds the frequency of a specific word in a specific document. Depending on the size of the corpora, the included text material and the language, most cells are empty. Compared to English language, this is especially true for languages with a high number of inflected words and rich compounding as for example French and German, at least when no lemmatization (reduction of inflected words to their dictionary form) or filtering of infrequent words is applied to the raw text material. A medium sized corpus in German language (e. g. 5 million words in 50 000 documents) usually results in a sparse matrix with a density below .05%.

This huge frequency matrix already comprises all the information that is subsequently utilised for text processing. Nevertheless, it is too huge to reasonably apply it to similarity judgements because it contains of irrelevant information ("noise"). To reduce the noise, several steps are necessary. First of all, words with either very low or very high frequency are usually filtered out. In the next step a weighting algorithm is applied to the matrix (Nakov, Popova, & Mateev, 2001) in order to emphasize words with a specific meaning. Finally, the matrix is decomposed via Singular Value Decomposition (SVD) similar to the procession in a Principal Component Analysis (PCA). Contrary to the eigenvalue decomposition in PCA, where a decomposition of the square matrix of covariances is done, the SVD used in LSA is the decomposition of a rectangular matrix of weighted term frequencies (mathematical description see Berry, Dumais, & O'Brien, 1995; Martin, & Berry, 2007). The decomposition results in three orthogonal partial matrices: A term matrix (comparable to the factor values in PCA), a document matrix (comparable to the factor loads in PCA) and a diagonal matrix with the singular values (comparable to eigenvalues in PCA).

By reducing the number of extracted dimensions to a minimum, noise is reduced and the amount of data and memory consumption is downsized. Contrary to PCA, there is no criterion how many dimensions should be extracted. Numbers of 300 to 1500 dimensions turned out to work best (Dumais, 1991; Graesser et al., 1999; Nakov, 2000; Wild, Stahl, Stermsek, & Neumann, 2005). As the SVD is extremely resource costly in case the frequency matrices are decomposed completely, it is highly recommended to use the Lanczos (1951) algorithms, to do an abridged computation with a predefined number of dimensions. The SVD is an extremely resource costly process. However, the Lanczos (1951) iterative algorithm can be used to decompose very large and sparse rectangular matrices, when a predefined number of dimensions is demanded.

Eventually, the results of the SVD establish an n-dimensional orthogonal space ("semantic space"), where the terms and documents are distributed according to their common usage. Thus, the vector of a term partly represents its semantic content. The position within the semantic space reflects the relation of the meaning to other words or documents. As a result, words occur near to each other in the semantic space if they are often used in the same contexts, no matter whether they are actually used in the same documents or not (higher order cooccurrences; Lemaire, & Denhière, 2004; Kontostathis, & Pottenger, 2002).

2.2 Using LSA for Similarity Judgements and Information Retrieval

The SVD is a relatively resource intense computation process. Once it is finished, it enables high-performance similarity judgements between words or texts. There are several possible similarity measures like the Euclidian distance or the cosine of the angle between two vectors. The cosine turned out to be a robust similarity measure (Landauer, Laham, Rehder, & Schreiner, 1997; Rehder, Schreiner, Wolfe, Laham, Landauer, & Kintsch, 1998). Moreover, it yields a simple interpretation because it can be used just like a linear correlation.
New text material can be compared by projecting it into the semantic space, a process called "folding in". Simply speaking, the words of the new text are filtered and weighted in the same way

as the original text corpus and multiplied by the respective word vectors in the semantic space. Subsequently the vectors of the words are summed up to a new vector whose length and direction represent the meaning of the new text. To find similar documents within the semantic space, new material is folded into the semantic space and texts are retrieved, that meet predefined proximity thresholds.

In contrast to other methods of automated text analysis, LSA is able to categorize semantically related texts as similar, even when they do not share a single word. For example, the two sentences "A penguin is a bird that lives on fish and krill" and "Penguins are birds, which eat crabs and fishes" have a cosine of .763, despite the fact, that they do only share the word "and" (computation done with word material in German language). On the other hand, the first sentence has only a cosine of .563 with the sentence "A whale is a marine mammal that lives on fish and krill", although there is a large word overlap between the two sentences (demonstration available under Lenhard, Baier, Schneider, & Hoffmann, 2006).

Altough its elegance, there are limitations to LSA. One of the main points is the lack of syntax, word order and negation. For an LSA-system "heaven" and "hell" mean more or less the same, because the two concepts are highly interconnected to each other. As a result, LSA-systems in general do not work well on topics and tasks that highly rely on argumentation structure and logic. Moreover LSA usually performs better on texts containing multiple sentences compared to short answers (e.g., only single sentences; Landauer, Foltz, & Laham, 1998).

Despite the fact, that LSA is only a statistical technique and does not yield real verbal intelligence, LSA nonetheless exhibits an astonishing degree of expertise on tasks that afford verbal intelligence and semantic knowledge, as for example multiple choice knowledge tests (Landauer, & Dumais, 1997; Lenhard, Baier, Hoffmann, & Schneider, in press), automatic essay grading (Landauer, Laham, Rehder, & Schreiner, 1997; Lenhard, Baier, Hoffmann, & Schneider, in press), the measurement of textual coherence and prediction of reader's comprehension (Foltz, Kintsch, & Landauer, 1998), the prediction of knowledge gains of readers on the basis of their background knowledge (Wolfe et al., 1998) and last but not least intelligent tutoring systems (e. g. Caccamise, Franzke, Eckhoff, Kintsch, & Kintsch, 2007; Wade-Stein, & E. Kintsch, 2004; Lenhard, in submission).

2.3 Technical Infrastructure

The LSA-platform is based on a server / client architecture. The server is the core of the system and was designed to support multitasking and multi-user capabilities as well as portability (platform independent). Moreover it supports most of language encodings and thus can be used on almost every language. The main tasks on the server are user authentication and authorization, corpora administration, generation and weighting of frequency matrices, singular values decomposition (SVD), generation of semantic spaces and calculation of text similarities. The Lanczos interactive algorithm (Lanczos, 1951) is used for decomposing the singular values. The server supplies an API (Application Programming Interface) that allows third party software (clients) to communicate locally. The remote communication is supported via remote method invocation using Java RMI that extends the server API. Thus, a client can invoke a remote object in the server in an easy and standard way.

There are several client applications such as the server administration web interface (Lenhard, Baier, Schneider, & Hoffmann, 2006), a system for automatic essay scoring of student writings in university lectures and laboratory prototypes of conText, an intelligent tutoring system aimed at fostering text comprehension in students (Lenhard, Baier, Hoffmann, Schneider, Lenhard, 2007).

The computation time for a semantic space varies with the density of the frequency matrix and the distribution of the words. The decomposition of the biggest DEFT'07 corpus (Débat) and extraction of 300 dimensions took 18 min 12 sec for instances, whereas the smallest (Articlè) afforded only 39 sec time (Pentium IV, 3.2 GHz, 3 GB RAM). Post the SVD, the spaces are loaded into RAM and calculation of text similarities take only fractions of millisecs. The results reported in this paper have been done on clients running in Paris, while the LSA-server had been placed in Würzburg/Germany.

2.4 Hypotheses

We have two sets of hypothesis : the first set corresponds to the characteristic that an ideal memory space should have to get the best performance in order to solve a categorization task. The second set of hypthesis corresponds to the materials characteristics in order to apply the similarity comparison.

2.4.1 Caracteristics of semantic spaces

Our first set of hypothesis concerns the « experience » effect that the memory space build would have. We contrast two cases. In the first case, we build a semantic space corresponding to a big semantic memory where all types of knowledge are mixed . In the second case, we construct different semantic spaces which correspond to specific types of knowledge. The first will be called « the Big box method »and the second case will be called the « small box » method. We test which case has the best perfomance in a categarisation task.

2.4.2 Caracteristics of the material used to applied comparisons

Our second set of hypotheses concerns the characteristics of the mental representation corresponding to each category between which we have to choose. The idea is that, in a situation of a categorization task, a subject should have a proper mental representation of the different categories he/she have to choose between. Let us consider the case, that someone gives me a critic of a movie. To tell something about the jugment given by the critic, I must know something about how people make their judgment about a good movie or a bad movie. In other words, I must have a proper mental representation of « what is a good critic of a movie » and « what is a bad critic of a movie ». To follow this idea, if a subject has a proper mental representation of the categories that he/she has to choose between, the second step should be to compare the similarity between the incoming object and the mental representation for each category then he has to choose between them the answers of the category is the one that is the most similar to the object. For each category, we create a vector corresponding to all the documents of this category. For example, for the category « good movie », we compute the corresponding vector to the whole document « good movie ». We assume that this vector corresponds to the intention of the category « good movie ». We assume also that this vector corresponds to the mental representation of the categorys intention « good movie ». Following this idea, in order to know if a critic corresponds to the category « good movie », we will compare it with the vector corresponding the category « good movie ». These vectors should be like targets : the more specific they are, the better they are, then more accurate the results will be. We call : target files or target vectors.

3 Building and Training of Spaces

As soon as we had obtained the training data from Deft, the top priority was to build the spaces necessary for classification. Table 1, shows the distribution of documents among types and

judgments in the training data. In the table, the rows represent the types, while columns stand for judgments in the types. In particular, there is no judgment 2 in type Débats.

	0	1	2	Total
aVoiraLire	309	615	1150	2074
Débats	10400	6899	0	17299
Jeuxvidéo	497	1166	874	2537
Relecture	227	278	376	881
				22791 (Total)

Table 1. Distribution of Documents through Types and Judgments.

From Table 1, it is clear that the documents are not evenly distributed among types and judgments. This gives rise to two questions: i) If all the documents are used for building spaces, would this result in unbalanced spaces? ii) If we balance the documents to build a balanced space, would the left out documents cause information loss? Apart from those, there is another consideration about the space: Should we use a big space (big box) for all the types, or one small space (small box) per type? After trying different configurations, we decided to use 3 different methods to build spaces for the final tests:

1 Big box space balanced by number.
2 Small box spaces balance by number.
3 Small box spaces with all the documents.

We can see, there is one big box method, while there are two small box methods. During the unofficial tests, though, the big box method does not perform as well as small box methods, the overloaded LSA server prevented us from finishing the official tests with small box spaces which is balanced by number before the deadline. Hence we included the big box results in the official test results. In the following subsections, the details about those spaces are described.

At this point we have successfully extracted every document from XML files, and organized them in hierarchic directories: everything is ready for space construction.

3.1 Big box space balanced by number

In the purpose of producing a balanced LSA matrix, we decided to balance the number of documents for building the space. In Table 2 the balanced numbers for Documents are shown. We choose 300 dimensions for every judgment, because this is the closest number to the least numbered judgment, i.e. type Relecture, judgment 0 and 1. There are at least 10 documents that are not selected from each judgment in every type, even if the documents in the judgment are not enough to extract 300 dimensions, and they are for unofficial testing purpose. This means we have 120 documents in total from all the types and judgments. In the case of type Jeuxvidéo, there is no document in judgment 2, thus 450 documents are chosen for each of judgment 0 and 1 to make a total number of 900 for the type. The same for the unofficial test documents: 15 documents are chosen from each type of Juexvidéo to make a total number of 30 documents for the type.

	0	1	2	total
aVoiraLire	299	300	300	899
Débats	450	450	0	900
Jeuxvidéo	300	300	300	900

Relecture	217	268	300	785
Bigbox				3484

Table 2. Document numbers for balanced by number space.

3.2 Small box spaces with all the documents

The one big box for all the types may works well for distinguishing types, which is already known, whilst it may not work well for judgments inside types because of the noise from other types inside the space. A solution is to split the big box space into four smaller spaces, each corresponding to a type. In this small box method, no balancing is used, which means all the training documents are used to build the spaces. This guarantees the spaces are trained by all the information available. In Table 3, we can see the contents of these spaces. Thanks to their unbalanced feature, the table can be directly derived from Table 1.

	0	1	2	Total Documents
aVoiraLire space	309	615	1150	2074
Débats space	10400	6899	0	17299
Jeuxvidéo space	497	1166	874	2537
Relecture space	227	278	376	881

Table 3, small box spaces with all the documents.

3.3 Small box spaces balanced by number

We have already made an argument for small box above in subsection 3.2. However, with all the documents included for a space, there may occur unbalance inside the space. For example, in Table 3, the aVoiraLire space includes 2074 documents, 15% of which are from judgment 0, 30% are from judgment 1, and the rest i.e. 55% are from judgment 2. Being afraid that the unbalanced spaces may influence test results, the small box spaces are balanced by number in this method. Table 4, shows the contents of every space.

	0	1	2	Total Documents
aVoiraLire space	309	309	309	927
Débats space	6899	6899	0	13798
Jeuxvidéo space	497	497	497	1491
Relecture space	227	227	227	681

Table 4, Small box spaces balanced by number

3.4 Other spaces

There are some other methods we have tested but have dumped for different reasons.

1 Small box spaces for judgments. There are three spaces which contain documents from judgment 0, 1 and 2 respectively. The spaces do not differentiate between types. The unofficial tests gave very poor results for this method, and we dumped it. The argument is that even if the spaces seem to be organized by judgment similarity, the words for judgment vary in different types, which in turn diminishes the similarity. Further more, documents from different types bring noise to each other, thus make judgment more inaccurate.

2 Small box spaces balanced by percentage. This method is the same as the method in 3.2 apart from we balance the documents by percentage. For example we take 90 percent of documents from each judgment to make the spaces. This seems we have "balanced" the documents, but the unbalanced feature of the space will not be solved by this improper balancing method, besides there some information loss occurs, due to the not included documents. In the other hand, we can always produce the method 3.2 by increasing the percentage to 100 percent. Thus when the percentage increases, the result will be similar to the 3.2 method.

4 Final Tests and arguments

With the above spaces, we are almost ready to do the tests. We performed three different tests, each of which corresponded to each of the above spaces. But before that, for every test we had to prepare target files and test files. Test files were the same for every test, while the target files differed from one another. In Table 5, the test files' contents are shown.

Type	Test Documents
aVoiraLire	1388
Débats	11534
JeuxVidéo	1695
Relecture	603

Table 5, Official test documents.

If a space, its target files and test files are given, we are able to perform the test. Below code explains the algorithm for the test:

```
Declarations:
        target-file0; //judgment 0
        target-file1; //judgment 1
        target-file2; //judgment 2
        test-file; //test file
        space;   //lsa space
Program:
        result0 = cosine(space, target-file0, test-file);
        result1 = cosine(space, target-file1, test-file);
        result2 = cosine(space, target-file2, test-file);
        result = max(result0, result1, result2);
        if ( result == result0)
                then judgment = 0;
        else if (result == result1)
                then judgment = 1;
        else judgment = 2;
        return judgment;
```

Let us explain the code. The declaration part declares target-filetarget-files, test-file and space. These information could be passed to this part of code as parameters. Then the program compares the target-file and the test-file. The comparison is done by computing the cosine between two vectors of these two files in the given space (see more details in section 2). And the last part of the code is to find out which result is the maximum one, and to which target-file it corresponds to. This means when the cosine between a target-file and the test-file is maximum, the test-file is close to this target-file, rather than two others. From this, we say the test-file belongs to the judgment to which the target-file belongs. There some points we would like to make clear. i) This piece of code is only for one test-file, for multiple test files it should be called by another program in a loop structure. Of course, in a real world implementation, this code could be embedded into a program with more efficiency in mind. ii) In the code there are three target files, but in the case of Débats , there should be only two target files.

We have explained test files and spaces but not yet target files. Target files are the one against which we compare our test files. In general it contains all the documents from one judgment which are used to build spaces. So with the change of space, the target files change. In the following subsections, we will talk about test target files, and test results.

4.1 Test for big box space balanced by number

In this test, there are 11 target files every one of which corresponds to a judgment. We can use Table 2 to explain the contents of a target file. For example, a target file for type relecture, judgment 0, contains 217 documents. We don't have to change spaces, since there is only one space.

The results are shown below:

Corpus avoir-alire (1), exécution 1, F-score pondéré :

Précision: 0.459110186683814 Rappel: 0.49455465710474

F-score pondéré: 0.476173746841085

Corpus des jeux vidéo (2), exécution 1, F-score pondéré :

Précision: 0.647385184622607 Rappel: 0.633138889933088

F-score pondéré: 0.64018278968055

Corpus des relectures (3), exécution 1, F-score pondéré :

Précision: 0.397253652191251 Rappel: 0.399309105766007

F-score pondéré: 0.398278727028512

Corpus des débats (4), exécution 1, F-score pondéré :

Précision: 0.577640356467526 Rappel: 0.575419817485701

F-score pondéré: 0.576527948843027

4.2 Test for small box space with all the documents

Here we use 4 small spaces, and 11 target files. Target files can be explained using Table 3.
the results are:

Corpus avoir-alire (1), exécution 2, F-score pondéré :
Précision: 0.610107317646253 Rappel: 0.58790185104376
F-score pondéré: 0.598798791785183

Corpus des jeux vidéo (2), exécution 2, F-score pondéré :
Précision: 0.740923815003921 Rappel: 0.663222700360692
F-score pondéré: 0.699923388295196

Corpus des relectures (3), exécution 2, F-score pondéré :
Précision: 0.503115560359155 Rappel: 0.510760891580065
F-score pondéré: 0.506909400421208

Corpus des débats (4), exécution 2, F-score pondéré :
Précision: 0.679694375399586 Rappel: 0.683170682019693
F-score pondéré: 0.681428095142408

4.3 Test for small box spaces balanced by number

We use four small spaces and 11 target files. Target files can be explained by Table 4. Even if the result of this test is not officially submitted, we believe they are worth mentioning here.
there results are:

Corpus avoir-alire (1), exécution 3, F-score pondéré :
Précision: 0.386077238255826 Rappel: 0.408823220435605
F-score pondéré: 0.397124792556954

Corpus des jeux vidéo (2), exécution 3, F-score pondéré :
Précision: 0.644021148754158 Rappel: 0.591851261951918
F-score pondéré: 0.616835081537432

Corpus des relectures (3), exécution 3, F-score pondéré :
Précision: 0.453810019726474 Rappel: 0.462140584841882
F-score pondéré: 0.457937419064932

Corpus des débats (4), exécution 3, F-score pondéré :
Précision: 0.670371869281329 Rappel: 0.67373015608171
F-score pondéré: 0.672046817281907

4.4 Discussion

From the results we can see, the 4.2 test has the best result. 4.1 has the worst result. and 4.3 is a little bit better than 4.1 but not so good as 4.2. These results show the better performance in the categorization task when similarity is computed from a specific memory space than when it is computed from a global memory space. Some similar results have been found concerning relative judgement of importance of sentences in comprehension of narratives (Denhière, Hoareau, Jhean-Larose, Lehnard, Baïer, Bellissens,2007).

5. Conclusion

We have implemented two hypothesis about the categorization activity. We have compared two methods called “Big box” versus “Small Box” that correspond to two hypothesis about the type of

knowledge organization of semantic memory. The results are better with - the “Small box” method than with the “Big Box” method, even if the resulting semantic spaces are constructed of a small amount of data and that LSA is a statistical method.

6. Literature

BERRY, M. W, DUMAIS, S. T, AND O'BRIEN, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review, 37(4)*, 573-595.

CACCAMISE, D., FRANZKE, M., ECKHOFF, A., KINTSCH, E., & KINTSCH, W. (2007). Guided Practise in Technology-Based Summary Writing. In D. S. MCNAMARA (Ed.), Reading Strategies (375-396). Mahwah, N. J., Erlbaum.

DEERWESTER, S., DUMAIS, S. T., FURNAS, G. W., LANDAUER, T. K., & HARSHMAN, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society For Information Science, 41*, 391-407.

DENHIÈRE, G., LEMAIRE, B., BELLISSENS, C. & JHEAN-LAROSE, S. (2005). Psychologie cognitive et compréhension de texte : une démarche théorique et expérimentales. In S. Porhiel & D. Kintgler (Eds), L’unité texte (pp. 74-95) Pleyben : Perspectives.

DENHIÈRE, G., LEMAIRE, B., BELLISSENS, C. & JHEAN-LAROSE, S. (2007). A semantic space for modeling a child semantic memory. In T. LANDAUER, D.S. MCNAMARA, S. DENNIS, & W. KINTSCH (Eds), *Handbook of Latent Semantic Analysis (pp. 143-167).* Hillsdale, N.J. : Lawrence Erlbaum Associates.

DENHIERE, G., HOAREAU, Y. V., JEAN-LHAROSE, S., LENHARD,W.,BAIER,H., BELLISSENS, C. (2007). Human Hierarchization of Semantic Information in Narratives and Latent Semantic Analysis. *Proceedings of the First European Workshop on Latent Semantic Analysis in Technology Enhanced Learning,* 15-17.

DUMAIS, S. T. (1991). Improving the retrieval of information from external resources. *Behavior Research Methods, Instruments, & Computers, 23*, 229-236.

FOLTZ, P. W., KINTSCH, W.,& LANDAUER, T. K. (1998). The measurement of textual Coherence with latent Semantic Analysis. *Discourse Processes, 25*, 285-307.

GRAESSER, A., WIEMER-HASTINGS, K., WIEMER-HASTINGS, P., KREUZ, R., & THE TUTORING RESEARCH GROUP (1999). AutoTutor: A simulation of a human tutor. *Journal of Cognitive Systems Research, 1*, 35-51.

KONTOSTATHIS, A., & POTTENGER, W.M. (2002). Detecting patterns in the LSI term-term matrix. *Workshop on the Foundation of Data Mining and Discovery, IEEE International Conference on Data Mining*.

LANCZOS, C. (1950). An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators [Online]. Available: http://www.cs.duke.edu/courses/fall06/cps258/references/Krylov-space/Lanczos-original.pdf, date of retrieval: 10.01.2007). *Journal of Research of the National Bureau of Standards, 48*, 255-282.

LANDAUER, T. K., & DUMAIS, S. T. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review, 104*, 211-240.

LANDAUER, T. K., FOLTZ, P. W., & LAHAM, D. (1998). Introduction to Latent Semantic Analysis. *Discourse Processes, 25*, 259-284.

LANDAUER, T. K., LAHAM, D., REHDER, B., & SCHREINER, M. E., (1997). How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. In M. G. SHAFTO, & P. LANGLEY (Eds.), *Proceedings of the 19th annual meeting of the Cognitive Science Society (pp. 412-417)*. Mawhwah, NJ: Erlbaum.

LEMAIRE, B., & DENHIÈRE, G. (2004) Incremental Construction of an Associative Network from a Corpus. In K. FORBUS, D. GENTNER, & T. REGIER (Eds.), *Proceedings 26th Annual Meeting of the Cognitive Science Society (825-830)*, Chicago.

LEMAIRE, B., DENHIÈRE, G., BELLISSENS, C. & JHEAN-LAROSE, S. (2066). A model and a computer program for simulating text comprehension. *Behavior Research Methods,* 38 (4), 628-637.

LENHARD, W. (in submission). Bridging the Gap to Natural Language: Latent Semantic Analysis as a Tool for the Development of Intelligent Tutoring Systems. LENHARD, W., BAIER, H. SCHNEIDER, W., & HOFFMANN, J. (2006). Forschungsprojekt "Förderung des

Textverständnisses": LSA-Modul [Research Project „Enhancing text comprehension“, LSA-Module]. accessible under: http://www.summa.psychologie.uni-wuerzburg.de/summa/coa/login/ (last accessed April 2007).

LENHARD, W., BAIER, H., HOFFMANN, J., & SCHNEIDER, W. (in press). Automatische Bewertung offener Antwortformate mittels Latenter Semantischer Analyse [Automatic Scoring of Open Text Responses via Latent Semantic Analysis]. *Diagnostica*.

LENHARD, W., BAIER, H., HOFFMANN, J., SCHNEIDER, W., & LENHARD, A. (2007). Training of summarisation skills via the use of content based feedback. *Proceedings of the First European Workshop on Latent Semantic Analysis in Technology Enhanced Learning,* 26-27.

Martin, D., & Berry, M. (2007). Mathematical Foundations Behind Latent Semantic Analysis. In T. K. Landauer, D. S. McNamara, S. Dennis, & W. Kintsch (Eds.), *The Handbook of Latent Semantic Analysis*. Mahwah, NJ: Erlbaum.

NAKOV, P. (2000). Getting Better Results With Latent Semantic Indexing. *Proceedings of the Students Presentations at the European Summer School in Logic Language and Information (ESSLLI'00)*, 156-166.

NAKOV, P., POPOVA, A., & MATEEV P. (2001). Weight functions impact on LSA performance. *Proceedings of the EuroConference Recent Advances in Natural Language Processing, RANLP'01*, 187-193.

REHDER, B., SCHREINER, M. E., WOLFE, M. B., LAHAM, D., LANDAUER, T. K., & KINTSCH, W. (1998). Using Latent Semantic Analysis to assess knowledge: Some technical considerations. *Discourse Processes, 25*, 337-354.

TISSERAND, D., JHEAN-LAROSE, S., DENHIÈRE, G. (2007). Eye movement analysis and Latent Semantic Analysis on a comprehension and recall activity. *Proceedings of the First European Workshop on Latent Semantic Analysis in Technology Enhanced Learning,* 36-37.

WADE-STEIN, D., & KINTSCH, E. (2004). Summary Street: Interactive computer support for writing. *Cognition and Instruction, 22*, 333-362.

WIEMER-HASTINGS, P. (1999). How latent is Latent Semantic Analysis? *Proceedings of the Sixteenth International Joint Congress on Artificial Intelligence (pp. 932- 937).* San Francisco: Morgan Kaufmann.

WILD, F., STAHL, CH., STERMSEK, G., & NEUMANN, G. (2005). Parameters Driving Effectiveness of Automated Essay Scoring with LSA. *Proceedings of the 9th International Computer Assisted Assessment Conference,* 485-494.

WOLFE, M. B., SCHREINER, M. E., REHDER, B., LAHAM, D., FOLTZ, P. W., KINTSCH, W., & LANDAUER, T. K. (1998). Learning from text: Matching readers and text by Latent Semantic Analysis. *Discourse Processes, 25*, 309-336.

Index des auteurs

Index des mots-clés