Fouille d'opinions

Sébastien Gillot

Rapport de stage

Fouille d'opinions

Sébastien Gillot (MRI parcours P4)

4 juin 2010

Équipe: Texmex
Encadrants: Vincent Claveau et Pascale Sébillot

Table des matières

Introduction

L'internet social a récemment fait exploser la disponibilité de documents textuels exprimant des opinions ou des sentiments, par exemple dans les groupes de discussions, les blogs, forums et autres sites spécialisés dans les critiques de produits. Les opinions disponibles sur l'internet ont un impact considérable sur les internautes. Des sondages (Pang et Lee (2008)) montrent que la majorité (80%) des internautes ont déjà fait des recherches d'avis sur un produit et que ces derniers sont prêts à payer deux fois plus cher pour un produit dont l'avis est plus favorable qu'un autre. Les entreprises prennent en compte ce paramètre et l'analyse d'opinions est depuis longtemps une composante importante dans leurs prises de décisions. La nécessité de traiter automatiquement les opinions se fait donc fortement ressentir. L'analyse automatique des opinions, aussi appelée fouille d'opinions, concerne l'extraction d'un sentiment dans une source telle qu'un texte sans structure prédéfinie. Les sentiments reconnus peuvent être classés soit positifs soit négatifs, soit en des classes définies plus finement.

Le sujet de notre stage portait sur la construction d'un tel système de fouille. Nous axons notre recherche sur la généricité, c'est-à-dire que nous souhaitons diminuer l'intervention humaine dans le processus de fouille. Pour cela, des exemples de textes d'opinions annotés selon le sentiment (positif ou négatif) qui leur est associé sont exploités. Nous nous plaçons donc dans un cadre d'apprentissage supervisé et explorons le problème de fouille d'opinions à travers l'utilisation de méthodes standards. Beaucoup de ces méthodes nécessitent des ressources *a priori* telles que le vocabulaire et la structure des phrases. Dans ce rapport, nous tentons de pousser la généricité de ces méthodes en diminuant ce besoin en ressources *a priori*.

Ce rapport se divise en trois parties. La première s'intéresse aux problèmes de représentation et de classification des données pour la fouille d'opinions. La seconde décrit notre approche de ce problème. Nous constatons dans cette partie l'intérêt à diminuer les ressources manuellement définies afin de gagner en généricité. La troisième s'axe sur un problème d'annotation. En effet, l'annotation des exemples est généralement coûteuse. Elle nécessite soit l'utilisation d'une technique de classification basique (e.g. prédire le sentiment positif si le texte est accompagné d'une note supérieure à 10), soit l'avis d'experts pour chaque exemple. Nous cherchons dans cette troisième partie à diminuer le nombre d'exemples annotés en profitant des informations que l'on peut tirer de textes non-annotés. Dans une quatrième partie, nous concluons et donnons quelques perspectives à ce travail.

1 Le problème de fouille d'opinions

L'opinion peut-être définie (Liu (2010)) comme l'expression des sentiments d'une personne envers une entité, étant subjective et opposée à l'expression de faits. Dans l'exemple d'opinion 1, certaines phrases (1, 6) sont simplement factuelles (ou objectives), tandis que d'autres (2, 3, 4, 5) relatent le sentiment du critique envers l'entité.

```
"(1) I bought an iPhone a few days ago. (2) It was such a nice phone.
   (3) The touch screen was really cool. (4) The voice quality was
   clear too. (5) Although the battery life was not long, that is ok
   for me. (6) However, my mother was mad with me as I did not tell
   her before I bought it. ... "
```

Exemple 1 – Extrait d'évaluation de l'iPhone (utilisé par Liu (2010))

Le terme « fouille d'opinions » est utilisé pour évoquer le traitement automatiques des opi-

nions, des sentiments et de la subjectivité dans les textes. Ce domaine est connu sous les noms de *opinion mining* (Pang et Lee (2008)), *sentiment analysis* (Liu (2010)), ou encore *subjectivity analysis* et est souvent associé à un problème de classification sur des textes évaluatifs comme ceux disponibles sur *Amazon*[1] ou *Epinions*[2].

Les premiers travaux évoqués dans la littérature s'intéressent à la classification de textes suivant des genres, dont certains tels que « éditorial » sont subjectifs (Karlgren et Cutting (1994)). En 1994, Wiebe s'intéressait plus précisément à l'idée de subjectivité, en cherchant à détecter des *private states*, définit comme des états (dans ce cas des parties de textes) qui ne peuvent pas être associés à des observations objectives et vérifiables. À cette époque, les données disponibles devaient être annotées manuellement, ce qui rendait la tâche assez laborieuse. Avec l'essor de l'internet social, la disponibilité de données annotées par le critique (en associant une note à son texte) a explosée. En 2002, Turney, Pang et coll. encouragent la recherche dans le domaine de *sentiment analysis* en classant des critiques de cinéma. Dans les travaux de Turney, la classification dans ce domaine donnait les moins bon résultats parmi les autres domaines (automobiles, banques et tourisme). Pang parvient tout de même à de bons résultats en utilisant des techniques d'apprentissages simples et adaptables aux autres domaines.

La principale idée derrière la fouille d'opinions est de guider le processus de décision d'un utilisateur en proposant un résumé des évaluations sur un concept donné. Dans leur livre, Pang et Lee (2008) exposent les intérêts socio-économiques d'un hypothétique moteur de recherche qui répondrait à la requête « Que pensent les gens de .. ? ». Ce problème est approché par Vernier et coll. (2009) qui tentent de répondre à la requête « Que pensent les gens de Sarah Palin ? ». De nombreux auteurs (Pang et coll. (2002); Boiy et Moens (2009); Liu (2010)) utilisent la polarité du texte pour faire ce résumé. Nous nous situons parmi eux. Notons que le problème peut être situé différemment (Dave et coll. (2003)) en utilisant la liste des caractéristiques évaluatives pour résumer (e.g. `[(touch screen : cool), (voice quality : clear), (battery life : not long)]`, pour l'exemple 1).

Dans cette partie, nous présentons le domaine de la fouille d'opinions et l'état de l'art relatif à notre problème. Nous rapprochons le problème à deux fondamentaux : la représentation des données et leur classification.

Nous considérons deux sous-problèmes intrinsèques à la fouille d'opinions. D'abord il s'agit de déterminer de bons indices pour l'opinion, ensuite de déterminer comment utiliser ces indices. Nous parlons d'abord de la représentation des données qui a une part très importante. Puis nous évoquons les méthodes de classifications dans le cadre d'un apprentissage supervisé. Ensuite, nous rappelons les mesures d'évaluations de la classification de textes. Enfin, nous nous penchons sur la portabilité d'un classificateur.

1.1 Représentation de données

Dans de nombreux travaux, le texte est considéré comme un ensemble de mots souvent sans structure (représentation « sac de mots »). Néanmoins, l'utilisation de « sac de mots » ne se révèle pas toujours efficace, car les données issues du langage naturel sont séquentielles. Dave et coll. (2003) utilise les bi-grammes pour capturer la négation. Une approche alternative (chaque mots suivant une négation jusqu'à la ponctuation suivante reçoit une étiquette indiquant la négation) est utilisée par Pang et coll. (2002) dans son corpus de cinéma. L'utilisation de n-grammes et les étiquetages de mots peuvent être considérés comme des pré-traitement sur le texte avant la sélection des mots les plus significatifs.

1. `www.amazon.com`
2. `www.epinions.com`

Sélection du lexique Pour analyser un document, il est nécessaire de le représenter sous les bonnes dimensions. Dans notre contexte, les dimensions sont les termes (mots ou groupes de mots) qui composent le texte. Ces termes sont sélectionnés, soit de manière statistique selon leurs apparitions dans des documents positifs ou négatifs (Pang et coll. (2002)), soit de manière plus fines (mais aussi moins génériques). La manière fine utilise le plus souvent un lexique minimal et des connaissances grammaticales (*Part-of-speech tagging*) sur les mots. Selon Hatzivassiloglou et Wiebe (2000), les adjectifs sont de bons indicateurs d'un sentiment, mais leurs orientations dépendent du contexte. Turney (2002) détermine l'orientation des adjectifs dans un large corpus, selon leur proximité avec des mots graines ("excellent" et "poor"). La mesure utilisée par Turney est appelée PMI (*pointwise mutual information*), et est adaptée pour être utilisée dans un moteur de recherche (e.g. Altavista) disposant d'un opérateur « près de », lors de l'acquisition des données. Esuli et Sebastiani (2006) utilisent cette mesure pour construire un lexique pour la fouille d'opinions, chaque mot de ce lexique est associé à ses degrés de positivité, de négativité et de neutralité. Ce lexique est un sous-ensemble de la base de données lexicale *WordNet* et l'orientation des termes est déterminée grâce à un ensemble de classificateurs choisis pour leur bons résultats. Kaji et Kitsuregawa (2007) améliorent l'utilisation de PMI grâce des indices structurels afin de construire un lexique pour le Japonais. Notons que leur méthode profite d'une grande quantité de phrases d'opinions. Hu et Liu (2004) identifient les mots qui décrivent la caractéristique d'un produit (e.g. "The battery life is long") et les utilisent au moment de la classification. Polanyi et Zaenen (2006) adaptent l'orientation des mots selon le contexte (capturable par avec une approche numérique).

1.2 Classification

Une fois considéré un ensemble de mots ou plus généralement d'attributs pour décrire les documents, on peut s'intéresser à une phase de classification. Il est commode de distinguer deux grandes familles dans le domaine de la classification de textes. L'une, dite numérique, utilise des techniques d'apprentissage supervisé, où un grand corpus d'entraînement est utilisé pour construire un classificateur. L'autre, dite symbolique, utilise des règles qui s'appliquent sur des symboles le plus souvent définis manuellement. Les deux approches se distinguent surtout par le nombre de symboles qui peuvent représenter le document. Dans une approche symbolique, ces symboles sont souvent définis à l'aide d'experts, tandis que dans l'approche numérique, chaque mot rencontré dans le corpus d'apprentissage est potentiellement un symbole du texte. Les approches numériques offrent les intérêts d'être facilement adaptables à de nouvelles entrées, et la possibilité de mesurer un degré de certitude du classificateur.

Les techniques supervisé sont le plus souvent utilisées pour classifier des textes, comme le font Pang et coll. (2002). Ces mêmes auteurs (2004) appliquent des techniques supervisées pour écarter les phrases subjectives. Leur classificateur procède donc en deux étapes : filtrage puis classification. Boiy et Moens (2009) associent en cascade trois classificateurs. Leur but est d'écarter d'abord les exemples subjectifs avec le premier, puis les textes dont le sentiment est simple à identifier avec le second. Le dernier classificateur utilise des relations syntaxiques coûteuses et faillibles.

Afin de choisir le bon lexique et la bonne méthode de classification d'opinions, il est nécessaire d'évaluer les résultats de classification, c'est l'objet de la sous-section suivante.

1.3 Évaluation des méthodes

Pour mesurer l'efficacité d'un classificateur dans un problème à n classes (en l'occurrence deux : positif et négatif), trois mesures sont utilisées : la précision, le rappel et le F-score.

$$Précision = \frac{\sum_{i=1}^{n} precision_i}{n}$$

et

$$Rappel = \frac{\sum_{i=1}^{n} rappel_i}{n}$$

Étant donné pour chaque classe i :

$$précision_i = \frac{nombre\ de\ documents\ correctement\ attribués\ à\ la\ classe\ i}{nombre\ de\ documents\ attribués\ à\ la\ classe\ i}$$

$$rappel_i = \frac{nombre\ de\ documents\ correctement\ attribués\ à\ la\ classe\ i}{nombre\ de\ documents\ appartenant\ à\ la\ classe\ i}$$

La précision indique le degré de vérité des résultats obtenus, tandis que la rappel indique leur pertinence. Il est intéressant de faire un compromis entre la précision et le rappel, c'est le rôle du $F_\beta score$ dont le coefficient β est inversement proportionnel à l'importance qui est donnée à la précision par rapport au rappel.

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

Nous utilisons le $F_1 score$, qui est le compromis équilibré entre précision et rappel.

$$F_1 score = 2 \times \frac{Précision \times Rappel}{Précision + Rappel}$$

Habituellement, l'utilisation de ces mesures entre dans un processus de validation. Nous en utilisons deux : la validation par test et la validation croisée. Dans une validation par test, on apprend un modèle sur un corpus d'entrainement et on test la classification sur un autre corpus. Dans une validation croisée, on ne considère pas précisément d'ensemble d'apprentissage et de test mais des sous-ensembles d'un corpus qui sont utilisés pour se valider les uns les autres. La validation croisée est considérée plus fiable qu'une simple validation par test.

L'évaluation des méthodes ne reposent pas seulement sur les mesures d'évaluation, elle repose aussi sur la définition des ensembles d'entraînement et de test. Dans notre problème, si le corpus d'entraînement est trop proche du corpus de test, l'obtention de bons résultats n'assure pas que le modèle appris soit un bon modèle pour la fouille d'opinions. Nous pensons donc que l'évaluation de nos méthodes doit être faite dans un domaine sensiblement différent. Cela nous amène à présenter la portabilité.

1.4 Problème de la portabilité

Selon le domaine et la langue varie la disponibilité des textes annotés. Il est intéressant de pouvoir réutiliser un classificateur appris avec des textes fortement disponibles (e.g. des critiques issues du domaine automobile) pour classer des textes faiblement disponibles (e.g. issues de domaine du théâtre). Aue et Gamon (2005) montrent qu'un classificateur d'opinions est spécifique au domaine, mais aussi que la classification dans certains domaines dont les exemples présentent un language varié (e.g. cinema à cause des descriptions de films) peut profiter des exemples issues d'autres domaines. La portabilité est une problème qui a récemment reçu l'attention de nombreuses auteurs (Vernier et coll. (2009); Boiy et Moens (2009); DEF (2009)).

Dans les sous-sections suivantes nous décrivons notre approche du problème et présentons nos résultats.

2 Approche du problème

Dans les grandes lignes de la fouille d'opinions, on peut faire ressortir trois étapes : acquisition des données, apprentissage et test. Nous commençons par décrire nos corpus.

2.1 Les données

Plusieurs auteurs collectent leur données sur des sites tels que *IMDB* et *Amazon* (Pang et coll. (2002); DZICZKOWSKI (2008); Liu (2010)) dont les critiques sont disponibles grâce au *RSS*. Afin d'avoir une vérité terrain à propos de la polarité des critiques, nous avons choisis quant à nous d'aspirer de sites où le critique doit associer une note à son avis. Les données ont été collectées des sites *ParlonsTV*[3] et *JEUXVIDEO.COM*[4]. Nous donnons aux critiques une structure uniforme (auteur, date, catégorie, sujet, titre, texte, note, identité). Les problèmes d'encodage sont résolus en UTF8 et les images HTML sont normalisés. Le reste du texte est conservé brut.

Critiques issues de ParlonsTV Le site *ParlonsTV*, contient un panel assez large de sous-domaine de la télévision (émissions, séries, présentateurs, chaines, *people*). L'histogramme de répartition des notes (figure 1, à gauche) montre des notes réparties sur les extrêmes. Cela confirme l'*a priori* que les documents de ce corpus sont très majoritairement subjectifs, étant donné que les critiques sont le plus souvent des téléspectateurs qui veulent donner leur opinions.

Lorsqu'on regarde le corpus de plus près, on remarque qu'un grand nombre de critiques présentent des expressions assez directes tels que « j'adore », « j'aime », « je déteste ». Tandis que d'autres évoquent le non intérêt du sujet (avec des mots comme « nul », « stupidité » ou « vulgaire »). Aussi le langage est bruité, c'est-à-dire qu'on rencontre souvent des mots mal orthographiés, des contractions de mots et parfois un langage "SMS". Quelques documents présentent un caractère multimodale avec des émoticônes.

Critiques issues de JEUXVIDEO.COM Les notations des documents dans le corpus de critiques télévisuelles nous semblant biaisées par la subjectivité des auteurs, nous avons choisi d'acquérir des données d'un site dont les critiques sont un peu plus professionnelles et dont nous pourrions éventuellement tenté d'extraire les caractéristiques évaluatives des produits. La structure d'une critique de jeux vidéo permet en effet une évaluation de l'extraction de ces caractéristiques. Le jeux est présenter dans ces grandes lignes, puis il est donné une note commentée par l'auteur à chaque caractéristique prédéfini (graphisme, jouabilité (facilité d'utilisation), durée de vie, bande son, scénario). Aussi une note générale est donnée avec le résumé des caractéristiques. Néanmoins, nous avons préférez utiliser ce corpus pour faire une validation inter-domaine (note générale vers scénario ; qui selon notre intuition ne doit pas donner d'excellents résultats, car beaucoup des jeux vidéo n'ont pas de scénario).

Pour obtenir un modèle de classificateur d'opinions, les textes d'opinions sont placés en entrée d'une chaîne de traitement qui inclue un changement de représentation des données et l'apprentissage sur cette nouvelle représentation.

2.2 Chaîne de traitements

Nous présentons dans cette partie les traitements que nous appliquons à nos données afin d'en tirer un modèle et de l'évaluer. La figure 2 décris cette chaîne composé en 6 étapes. La première étape (1 et 1bis) est d'appliquer un même pré-traitement au corpus d'entraînement et

3. `www.parlonstv.com`
4. `www.jeuxvideo.com`

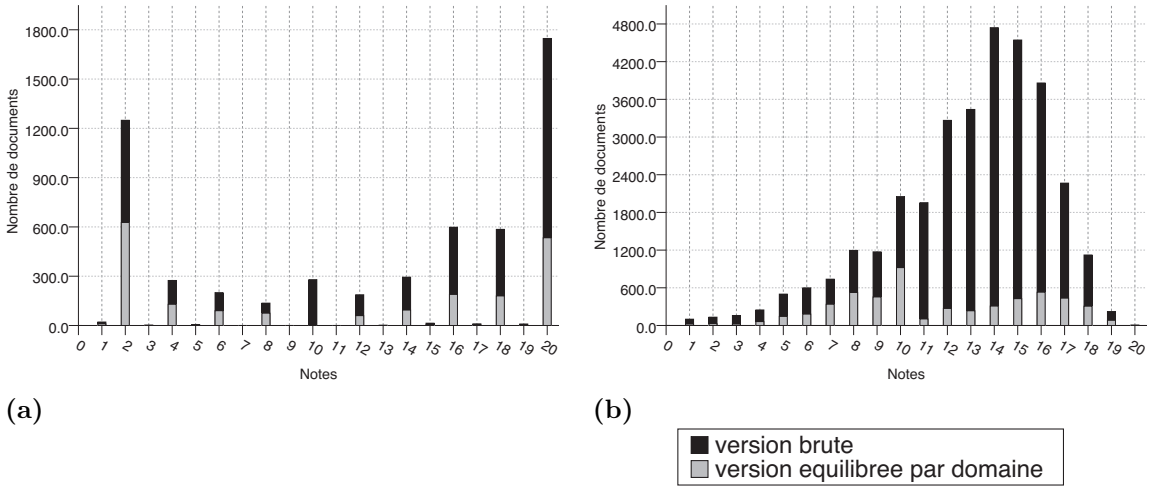


FIGURE 1 – Histogrammes de répartition des notes. Nous montrons en gris l'effet de notre équilibrage des notes sur cette répartition. Le principale constat est la perte de nombreux exemples. **(a)** Les notes du corpus de critiques télévisuelles sont réparties en forme de U, en effet dans la plupart des critiques soit l'opinion est très bonne soit elle est très mauvaise. Nous pensons que la notation est subjective, car les critiques sont des téléspectateurs et critiques amateurs. **(b)** Les notes du corpus de critiques de jeux vidéo sont réparti de manière opposée. Nous pensons que la notation est plutôt objective, étant donné que les critiques sont des professionnels.

au corpus de test. La seconde étape (2) est d'extraire un lexique du corpus d'entraînement, afin de représenter tous les documents avec des attributs identiques (3 et 5). Pour la construction (4) et le test (6) du modèle, nous utilisons l'outil *Weka*[5](Holmes et coll. (1994)) qui fournit un environnement de travail pour l'apprentissage supervisé et proposent de nombreux classificateurs. Suite à l'étape 6, nous utilisons les résultats de classifications pour faire l'évaluation.

2.2.1 Pré-traitement

Afin d'homogénéiser les données nous avons employé des traitements préliminaires sur les textes. Tous ces traitements ne sont pas systématiquement appliqués, certains sont considérés comme des paramètres lors des expériences.

Traitements systématiques Les traitements appliqués systématiquement sont la normalisation des caractères (dont suppression des accents) et la détection des entité nommées. Nous ôtons les accents pour la simple raison que ceux-ci sont souvent oubliés dans les corpus bruités (e.g. *ParlonsTV*). La suppression des accents est un moyen simple de réduire le nombre de dimensions. Les entités nommées sont des mots, ou un groupes de mots catégorisables dans des classes telles que noms de personnes, noms d'organisations ou d'entreprises, noms de lieux, date... Lorsque le corpus est assez varié, il est rare que la même entité apparaisse plus souvent que des mots du vocabulaire du sentiment, mais dans le cas contraire (e.g. dans le domaine de la télévision où l'on site souvent le présentateur ou la chaîne de télévision) certaines entités sont des éléments discriminants entre nos deux classes considérées (positif et négatifs). Afin d'éviter

5. http ://www.cs.waikato.ac.nz/ml/weka/

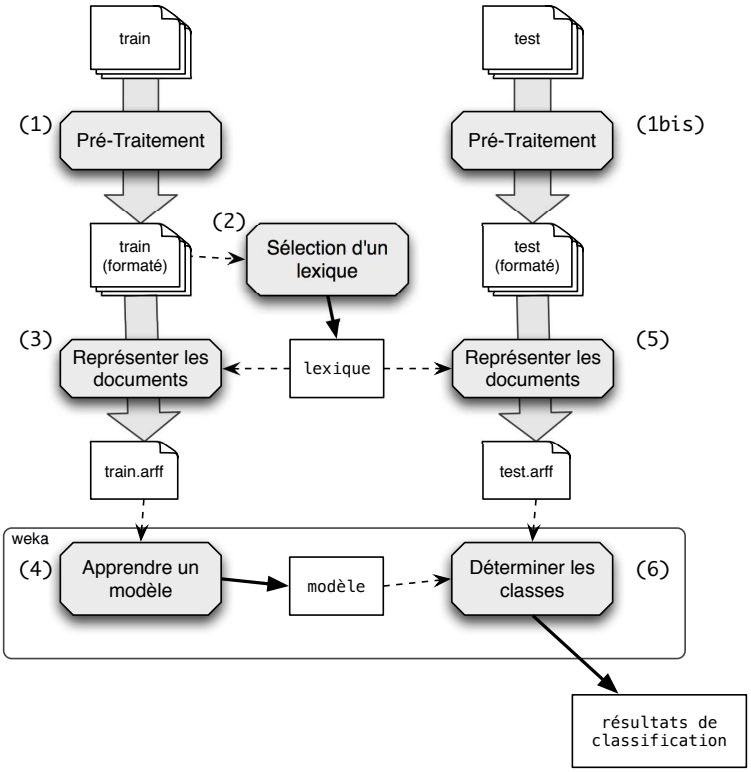


FIGURE 2 – Lors des étapes 1, 2, 3 et 5, le principale problème est de représenter les documents de manière homogènes en apportant assez d'informations au classificateur qui intervient lors des étapes 4 et 6.

ce biais à l'apprentissage (e.g. éviter que "Arthur" soit systématiquement lié à une opinion négative), nous détectons les entités nommées avec un système très basique : une étiquette `_name_` est ajoutée aux mots qui contiennent des chiffres et aux mots en majuscules qui ne sont pas précédés d'une ponctuation finale (nous n'étiquetons jamais le premier mot d'une phrase). Lors de la sélection du lexique, ces entités sont systématiquement écartées. La détection automatique des entités nommées (Fourour (2002); Raymond et Fayolle (2010)) est un problème qui nécessite des connaissances *a priori* (dont des lexiques des entités connues). Bien que les systèmes de détection des entités nommées soient assez génériques, nous ne les utilisons pas car notre détection semble suffisante pour notre problème.

Variables d'expériences Les pré-traitements suivants sont optionnels et sont utilisés pour certaines expériences dans l'idée que chaque terme restant après le pré-traitement est potentiellement un terme du lexique.

- **utilisation d'attributs issues de l'internet social** : De nombreux documents d'opinions disponibles sur la toile présentent des caractéristiques propres à l'internet social. Notamment, on rencontre des émoticônes et des acronymes tels que « :) », « :(» ou encore « lol ». Les émoticônes sont des symboles qui manifestent une émotion ou un état d'esprit, nous ne les ignorons donc pas car il peut s'agir d'indices génériques pour la fouille d'opi-

nions. L'avantage de ces symboles est qu'ils sont multilingues, néanmoins la forme d'un émoticône varie entre occident et l'orient (en effet, les émoticônes asiatiques, notamment au Japon, sont bien plus variés que les émoticônes occidentaux). Il peut donc s'agir d'indices génériques pour la fouille d'opinions. Notons l'existence de travaux sur l'extraction automatique d'émoticônes (Tanaka et coll. (2005)) réutilisés par Yasuhiro et coll. (2006) qui utilisent les émoticônes comme informations contextuelles lors de la classification. Nous reparlons de la valeur contextuelle des émoticônes dans la sous-section 2.3 (expériences). Dans notre chaîne de traitements, nous normalisons les émoticônes 8 bits (qui utilise la table de caractère minimale et commune à tout les pays), à partir d'un base de connaissance construite manuellement considérant 15 catégories dont 5 ont un lien direct avec l'émotion (joie, tristesse, adoration, ennui, colère) et 10 autres qui décrivent plutôt une attitude (tire de langue, clin d'œuil, rire, attitude "cool", surprise, confusion, silence, maladie, attitude clownesque, transpiration ou malaise). Nous tentons aussi de classer notre corpus avec ces seuls indices.

- **suppression des mots vides** : Les mots que nous appelons vides, sont ceux qui sont si communs, qu'ils semblent n'apporter aucune information utile pour la fouille. Ce sont principalement, les déterminants et les auxiliaires. Nous n'éliminons pas systématiquement ces mots pour la principale raison que nous considérons que leur suppression implique l'utilisation d'une connaissance apriorique (liste des mots vides). Nous étudions l'influence de leur suppression, et tentons aussi de les éliminer de manière statistique.
- **réduction lexicale** : Parmi les mots que l'on peut rencontrer dans un corpus textuel, il n'est pas rare d'en rencontrer plusieurs ayant un sens commun ou se déclinant simplement sous une autre conjugaison. Distinguer ces mots apportent des redondances d'informations non corrélées qui se traduisent par une perte sémantique. Certains traitements permettent d'éliminer ces redondances d'informations. Nous étudions l'effet d'une *lemmatisation* qui consiste à mettre les mots sous une forme canonique. (`je aimer` est un exemple d'opinion lemmatisé). Ce principe de réduction utilise un ensemble des lemmes qui associent au mot son entrée dans un dictionnaire.
- **réduction grammaticale** : Dans une phrase, chaque mot a un rôle particulier. La grammaire française classe les mots en catégories, appelées également « parties du discours » (adjectif, adverbe, article, conjonction...). Un étiquetage grammatical des mots (*part-of-speech tagging*) affecte à chaque mot une étiquette définissant sa catégorie dans la phrase. Afin d'étudier les effets d'une réduction grammaticale, nous utilisons l'étiqueteur *TreeTagger* (Schmid (1994)) pour lemmatiser les mots et réduire le texte à un ensemble de « parties du discours » tels que les adjectifs, les verbes et adverbes.
- **conversion des mots en n-grammes** : Pang et coll. (2002) constatent que l'utilisation bi-grammes offre de moins bon résultats que l'utilisation de simples mots (uni-grammes). Nous assimilons l'utilisation des bi-grammes à un pré-traitement transformant les textes en ensembles de bi-grammes (e.g. `_DEBUT_the the_touch touch_screen screen_was was_very very_cool cool_FIN_`).
- **normalisation de la ponctuation** : Au lieu de supprimer les ponctuation, nous les normaliser afin de voir l'influence certaines ponctuations interrogatives et exclamatives.

Une fois ces pré-traitements appliqués, la phase d'analyse statistique du lexique commence.

2.2.2 Sélection statistique du lexique

Un texte peut être représenté comme un vecteur de mots. Un problème au coeur de la fouille d'opinions est de choisir les dimensions de ce vecteur. Pour la suite, nous préférons parler d'éléments du lexique plutôt que de mots, pour généraliser le texte à tout résultat du pré-traitement.

Lors de l'étape de pré-traitement les textes sont représentés comme une suite d'éléments (unigrammes ou des bi-grammes, avec ou sans mots vides). Lors de l'étape de sélection, nous cherchons simplement à réduire le lexique aux éléments les plus pertinents. Cette sélection est faite de manière statistique. Nous n'utilisons aucune autre connaissance a priori que celles exploitées lors du pré-traitement. Une sélection plus efficace du lexique pourrait être faite en utilisant la co-occurence (e.g. PMI de Turney (2002)) de mots graines sur les adjectifs. Néanmoins, nous préférons miser sur des méthodes très génériques et peu sophistiquées.

L'idée est donc de choisir grâce à des mesures statistiques les éléments discriminants entre les classes. Nous utilisons une formule qui associe un score maximale aux éléments les plus discriminants sur l'ensemble des classes. Pour la suite, nous généralisons notre problème à deux classes (*positif* et *négatif*) de façon à s'adapter facilement les formules à d'autre classes (e.g. neutre). Les scores que nous utilisons dans nos tests sont fonctions du nombre de documents de chaque classe c , dans lesquels apparaît l'élément w. Les scores suivant utilisent la fréquence d'apparition du terme w dans une classe c du corpus.

$$tf(w,c) = \frac{nombre\ d'occurences\ de\ w}{nombre\ d'occurences\ total}$$

– Le *Log Entropy maximal* et *TFIDF* maximal sont des adaptations des mesures *Log Entropy* et *TFIDF* utilisé en recherche d'information qui retournent habituellement un score élevé si le terme w est représentatif d'un document. Nous généralisons le document à la classe c. Ces scores sont donc élevés si le terme w correspond bien à une classe c.

$$max_c logentropy(w) = max_c(entropy_{all}(w) * log(tf(w,c)+1))$$

$$max_c tfidf(w) = max_c(tf(w,c) * idf(w))$$

– Un autre moyen de considérer un terme w discriminant est de comparer sa fréquence d'apparition dans une classe c à la moyenne de ses fréquences d'apparition dans les autres classes de l'ensemble $all - c$.

$$\delta_{tf}(w) = \sum_{c \in all} |tf_c(w) - \frac{\sum_{i \in (all-c)} tf_c(w)}{|all - c|}|$$

La figure 3 permet de mieux comprendre quels termes sont choisis avec ce score.

Fréquences pour le terme w.

classe	tf	moyenne des tf pour les autres classes	différence
1	0.5	0.1	0.4
2	0.1	0.3	0.2
3	0.1	0.3	0.2

Fréquences pour le terme w'.

classe	tf	moyenne des tf pour les autres classes	différence
1	0.5	0.3	0.2
2	0.1	0.5	0.4
3	0.5	0.3	0.2

FIGURE 3 – Calcul du δ_{tf} pour les termes w et w' qui sont deux mots discriminants en terme de fréquence entre les 3 classes hypothétiques. w est discriminant car sa présence est caractéristique de la classe 1 tandis que w' car son absence est caractéristique de la classe 2.

Dans certains cas, nous utilisons des poids globaux pour exclure les mots trop fréquents.

– La fréquence inversé

$$idf(w) = log(1 + \frac{nombre\ de\ classe}{nombre\ de\ classe\ où\ w\ apparaît})$$

– L'entropie de Shanon

$$entropy(w) = -\sum_{c \in all} p_c(w) \frac{log(p_c(w))}{log(nombre\ de\ classe)}$$

où

$$p_c(w) = \frac{tf_c(w)}{nombre\ de\ fois\ que\ w\ apparaît}$$

– Une fonction de poids ρ inspirée de l'entropie de Bernouilli, et dont le rôle est d'écarter les mots trop fréquents

$$\rho_k(w) = bern_{\alpha_k}(tf(w))$$

où

$$bern_{\alpha_k}(x) = -x * log(\alpha_k * x) - (1 - x) * log(1 - x)$$

$$\alpha_k = k * (nombre\ de\ classes)^2 * \frac{log(nombre\ total\ de\ mots)}{log(nombre\ de\ classes)}$$

et k détermine l'exclusion des mots trop fréquents et doit être supérieur ou égal à 1. L'entropie de Bernouilli ($bern_1$) est défini entre 0 et 1, et forme un arc entre ces deux extrêmes, la valeur maximum tend donc sur le juste milieu. Lorsqu'on augmente α on tire le point défini en 1 vers le bas et le juste milieu (la valeur maximum) apparaît pour un x plus petit. L'idée pour exclure les mots trop fréquent est redéfinir ce milieu. Pour formuler cette définition de α_k, on admet comme hypothèse que la fréquence des mots discriminants entre les classes, diminue de lorsque le nombre de classes et le nombre de mots présents augmentent.

Pour représenter un document, on utilise les mots du lexique. Une question que l'on se pose est « doit-on considérer le mot par sa simple présence dans le document ou alors par d'autres propriétés ? ». Cette question a été un peu exploré par Pang et coll. (2002). Dans ce rapport, nous appelons ces propriétés des traits.

2.2.3 Traits

Chaque document est représenté par un vecteur de traits relatifs aux éléments du lexique. Ces traits décrivent la présence de l'élément, son nombre d'apparation, ou encore la position dans le document.

Présence Le concept de présence est binaire, soit l'élément du lexique apparaît, soit il est absent. Ce trait a montré son supériorité par rapport aux autres dans des travaux passés (Pang et Lee (2002, 2004)). Lors des expériences sur le corpus de critiques télévisuelles, nous essayons aussi une variante de la présence qui distingue la présence dans le titre et celle dans le texte.

Fréquence Nous testons aussi la fréquence de l'élément dans le document. Cette fréquence est le rapport du nombre de fois que l'élément est rencontré sur le nombre total d'éléments rencontrés dans le document. Il a été vu que ce trait utilisé avec des uni-grammes de mots n'aide pas à produire de meilleurs résultats. Néanmoins, nous préférons vérifier cela sur notre corpus.

Position Dans le domaine du cinéma, l'opinion est souvent statué soit au début, soit à la fin de la critique, le milieu du texte étant le plus souvent réservé à des détails descriptifs (DZICZKOWSKI (2008)). Nous expérimentons l'information des positions des éléments (e.g. début, milieu, fin, début et fin...) dans le document, pour représenter les critiques. Dans notre mise en oeuvre, on construit la liste des positions auxquelles le mot apparaît dans le document. On définit le début du texte comme le premier quart, et la fin comme le dernier quart. C'est aussi une alternative pour concevoir la présence, la fréquence et un minimum de séquentialité.

2.2.4 Classificateurs

Lorsque les documents sont représentés sous formes de vecteurs de traits, on a les données pour apprendre un modèle de classification. On utilise dans ce rapport trois types de classificateurs.

- **classificateurs de type probabiliste** : L'approche probabiliste en classification assigne un document d à une classe $c = argmax_c P(c|d)$. Cette méthode s'inspire de la Loi de Bayes et les probabilités sont fonction des mots contenus dans les documents.

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} \tag{1}$$

  Il existe plusieurs méthodes pour déterminer $P(c|d)$ (Pang et coll. (2002)), la méthode la plus simple est celle utilisée par le classificateur Bayésien Naïf, où les traits sont supposés conditionnellement indépendants :

$$P_{NB}(c|d) := \frac{P(c)P(\prod_{i=1}^{m} P(fi|c)^{n_i(d)})}{P(d)} \tag{2}$$

  où f_i est le $i^{ème}$ mot (supposé conditionnellement indépendante) parmi les m mots choisis (le dictionnaire) et $n_i(d)$ est son nombre d'occurrences dans le document d. Le principal défaut cette méthode est sa naïveté, liée à la probabilité $P(c)$. Par exemple, si on apprend un modèle avec une majorité d'exemples positifs, alors ce modèle classe souvent les données comme des positifs. Pourtant, la technique du Bayésien Naïf est simple et efficace pour la classification des opinions. Nous utilisons l'algorithme *Naive Bayes* de *Weka*.
- **classificateurs de type Séparateur à Vaste Marge** : Tandis qu'une approche probabiliste consiste à modéliser la probabilité qu'un document d appartienne à une classe c, une autre approche consiste en la construction d'un modèle discriminant. Les Séparateurs à Vaste Marge, souvent écourtés à l'acronyme SVM, reposent sur la notion d'hyperplan séparateur et de marge maximale. Considérons un espace multidimensionnel où chaque trait est une dimension. Un hyperplan séparateur entre deux ensembles de points (dans notre cas, des documents de polarités positives pour un ensemble et négatives pour l'autre) est la frontière entre ces deux ensembles. La marge représente la distance entre un des deux ensembles et cet hyperplan. Les SVM sont plus lourds à implémenter que le Bayésien Naïf et n'offrent pas toujours de meilleurs résultats que ce dernier (Pang et coll. (2002); Maurel et coll. (2008)). Le Bayésien Naïf et SVM sont souvent inter-changés afin de choisir celui qui offre les meilleurs résultats. Nous utilisons l'algorithme *SMO* (*Sequential Minimal Optimization*) de *Weka* avec les paramètres par défaut.
- **classificateurs de type Arbre de décision** : Les modèles construits par un Bayésien Naïf ou un SVM sont décris par un ensemble de paramètres regroupé dans une seule formule, ce qui n'est pas très lisible. Les classificateurs de types Arbres de décisions construisent un arbre dont chaque noeud correspond à une décision sur un paramètre. Les avantages des ces classificateurs sont la capacité à sélectionner automatiquement les variables discriminantes

et leur lisibilité. Les arbres de décisions ont aussi la réputation d'offrir des prédictions fiables. C'est pourquoi nous choisissons d'utiliser cette méthode. Nous utilisons l'algorithme *C4.5* (*J48* sous *Weka*), *NBTree* et les forêts aléatoires. *C4.5* est un algorithme de référence pour la construction d'arbres de décisions. *NBTree* est un hybride entre *C4.5* et le Bayésien Naïf (la construction des arbres y est faite en utilisant la propriété de Bayes). Les forêts aléatoires sont des ensembles d'arbres où les décisions sont prises selon des traits choisis aléatoirement.

La chaine des traitements est ainsi décrite. Elle représente bien la dualité représentation des données et classification des vecteurs de données. Dans la sous-section suivante, nous présentons nos résultats d'expériences portant sur les problèmes de représentation (nous faisons varier les pré-traitements, les méthodes de sélection et les traits) et de classification.

2.3 Expériences

Nous présentons ici les résultats concernant notre recherche d'un outil de fouille d'opinion générique. Dans ces expériences nous utilisons les corpus *ParlonsTV* et *JEUXVIDEO.COM*. Lorsque nous parlons du corpus *ParlonsTV* (ou TV), nous faisons référence à une expérience faite avec le sous-corpus *émissions-tv* (697 positifs et 697 négatifs) pour l'entrainement et *séries-tv* (468 positifs et 468 négatifs) pour le test. Lorsque nous parlons du corpus *JEUXVIDEO.COM* (ou JV), nous faisons référence à une expérience faite avec le sous-corpus *NotesGénérale* (1490 positifs et 1490 négatifs) critiques pour l'entrainement et *Scénario* (450 positifs et 450 négatifs) pour le test.

2.3.1 Attributs pour décrire les données

Différents attributs peuvent servir à classer les documents. Susuki et coll. (2006) utilisent les émoticônes et les mots d'exclamations tels que « phew » (souvent orthographié « pfff » par les français) pour leur valeur contextuelle. Nous expérimentons des attributs similaires pour classer les documents du corpus *ParlonsTV*.

Utilisation des émoticônes pour évaluer l'opinion Dans le corpus tiré du site de critiques d'émissions de télévision (*ParlonsTV*), quelques documents contiennent des émoticônes et des acronymes. Nous tentons d'utiliser ces indices simple afin de classer les documents. Pour ces expériences, nous utilisons un lexique constitué d'émoticônes uniquement et faisons varier les traits de classifications décris dans 2.2.3 auxquels nous ajoutons des considérations sur la présence et fréquence dans le titre ou dans le texte. Nous utilisons les traits *pres* (présence du terme dans le document), *pres-tt* (présence du terme dans le titre et présence du terme dans le texte), *freq* (fréquence du terme dans le document, *freq-tt* (idem *pres-tt* avec la fréquence) et *pos* (positions auxquelles apparaissent le terme). La figure 4 montre les résultats que nous obtenons en ne décrivant les documents qu'avec les émoticônes qu'ils contiennent. Les mauvais résultats de la table (a) illustrent simplement le fait que trop peu de documents contiennent des émoticônes. Les résultats intéressants sont sur les tables (b) et (c) de la figure 4. On constate d'abord que *SMO* (l'implementation des Séparateur à Vaste Marge que nous utilisons) surpasse *NB* (*Naive Bayes*) lorsque les données ne sont pas équilibrées, et l'inverse lorsqu'elles sont équilibrées. Une explication à cela est que *NB* fonctionne par probabilité d'apparition dans un corpus où 135 documents sont positifs et seulement 43 négatifs, *NB* classe donc plus de documents négatifs à tord comme positifs. Ensuite, la fréquence d'apparition des émoticônes dans le document semble instructive selon la table (b) où le meilleur score est satisfaisant (80.1%), tandis que la position apporte les pires résultats. Néanmoins la table (c) indique l'importance de la position de l'émoticône dans

Trait	Apprenant	Précision	Rappel
freq	NB	**0.515**	**0.466**
	SMO	**0.66**	0.367
freq-tt	NB	0.515	0.466
	SMO	0.535	0.431
pos	NB	0.653	0.368
	SMO	0.648	0.366
pres	NB	0.653	0.368
	SMO	0.536	0.432
pres-tt	NB	0.653	0.368
	SMO	0.646	**0.365**

(a) On tente de classer 2136 documents (autant de positifs que de négatifs équitablement répartis sur les différents sous-domaines du corpus) en utilisant uniquement les 15 catégories d'émoticônes considérées.

Trait	Apprenant	Précision	Rappel
freq	NB	0.787	0.779
	SMO	**0.801**	**0.794**
freq-tt	NB	0.755	0.753
	SMO	0.771	0.762
pos	NB	**0.657**	**0.661**
	SMO	0.755	0.753
pres	NB	0.746	0.741
	SMO	0.763	0.758
pres-tt	NB	0.683	0.691
	SMO	0.763	0.758

(b) On tente de classer seulement les documents (178) contenant des émoticônes.

Trait	Apprenant	Précision	Rappel
freq	NB	0.632	0.604
	SMO	0.628	0.607
freq-tt	NB	**0.693**	**0.649**
	SMO	0.683	0.635
pos	NB	0.683	0.635
	SMO	0.683	0.635
pres	NB	**0.624**	**0.61**
	SMO	0.624	0.61
pres-tt	NB	0.683	0.635
	SMO	0.683	0.635

(c) On réduit l'ensemble utilisé dans (b) afin d'avoir un équilibre entre les positifs et les négatifs (86 documents).

Trait	Apprenant	Précision	Rappel
freq	NB	0.723	0.693
	SMO	**0.751**	**0.715**
freq-tt	NB	0.679	0.685
	SMO	0.732	0.696
pos	NB	**0.638**	**0.628**
	SMO	0.725	0.689
pres	NB	0.683	0.664
	SMO	0.723	0.693
pres-tt	NB	0.682	0.66
	SMO	0.725	0.689

(d) On ajoute au lexique la categorie des accronymes assimilable à « lol », et on procède de la même manière que pour (b) (264 documents).

FIGURE 4 – Résultats de classification en utilisant les émoticônes comme lexique. Étant donnée le faible nombre de documents faisant apparaître les éléments souhaités, nous montrons la précision et le rappel. Les 4 tableaux présentent les résultats par validation croisée, pour différents sous-ensembles du corpus de critiques télivisuelles.

le texte (avec *pos*, *pres-tt* et *freq-tt*) pour un corpus équilibré. La table (d) présente de moins résultats que la table (c) car l'accronyme « lol » n'a pas une orientation (sentiment positif ou négatif) clairement défini. Pour conclure sur l'utilisation unique des émoticônes, nous considérons que la présence d'un émoticône n'est pas suffisante pour déterminer l'émotion exprimée dans le document.

Derks et coll. (2007) mettent en avant l'influence du contexte social sur la manière d'utiliser et sur la sémantique des émoticônes. Ce contexte peut-être capturé en analysant les mots dans le texte. Nous remarquons d'un autre coté que l'émoticône, comme l'expression du visage dans

a) « Il va en falloir du temps pour se rendre compte que ces programmes se valent tous et se rebeller à notre tour pour exiger de meilleurs émissions à la télévision!On fera l'enfant capricieux en pleurnichant comme une jeune fille en fleur et même la super Mary Poppins n'y pourra rien sinon on sera amener à lui montrer notre côté "homme des cavernes",en lui lançant des piques dans le derrière...:) »

b) « ..jvais vous faire une confidence ;) approcher.....j'ai toujours voulu à l'époque la carte club dorothée, bah jlai jamais eu :'(lool »

Exemple 2 – Deux textes issues du corpus *ParlonsTV*. Ils illustrent le contexte (l'ironie en l'occurence) apportée par les émoticônes.

une communication orale, pose un contexte permettant de capter l'ironie dans un texte. Dans l'exemple 2 a et b, « :) », « ;) » et « lool » marque un ton ironique, tandis que « :'(» indique une certaine empathie. Nous pensons qu'une analyse de la co-occurence des émoticônes avec des expressions indiquant un sentiment permettrait d'améliorer les résultats de classification (Susuki et coll. (2006) utilisent cette hypothèse et améliorent considérablement leur résultats). Ce processus étant complexe et notre corpus (limité au émoticônes) trop petit, nous normalisons seulement les émoticônes lors du pré-traitement. C'est le minimum à faire pour ne pas ignorer ces indices dans le cas où notre classificateur analyse un corpus riches en émoticônes.

Maintenant que nous avons exploités des indices « jouets » manuellement défini (pourtant génériques à notre avis), intéressons nous au problème de la sélection statistique des éléments du lexique.

2.3.2 Étude de l'influence des techniques de sélection du lexique

La sélection des éléments du lexique est un problème clé pour classification d'opinions. Nous expérimentons ici divers méthodes de sélections statistiques.

Nos premières expériences sont faites sur le corpus *ParlonsTV* et évaluons uniquement la représentation. Le processus d'évaluation est le suivant : le lexique est appris sur le sous-corpus $émissions - tv$, et l'évaluation est faite par validation croisée dans le sous-corpus $séries - tv$ représenté selon la présence (*pres*) des mots ce lexique.

Expériences préliminaires Avant de lancer des expériences sur l'influence de la taille du lexique, nous essayons plusieurs méthodes de sélections du lexique pour un lexique de 100 éléments avec l'apprenant *Naive Bayes* (choisi pour sa rapidité). Les scores que nous considérons sont définis dans la section 2.2.2. La figure 5 présente les résultats pour les différentes sélections statistiques. Nous remarquons que le score δ_{tf} défini précédemment marche assez bien, néanmoins le lexique obtenu présente beaucoup de mots vides, il en est de même lorsqu'on pondère les mots avec idf. Nous avons pondéré les mots sélectionnés par δ_{tf} avec le score ρ_k. Le paramètre k à été choisi en plusieurs essais sur le corpus TV en considérant un lexique de 1000 éléments. Dans la figure 6 on constate, les résultats se dégradent lorsqu'on exclus les mots trop fréquents. Cela peut dénoter l'importance des mots vides dans notre corpus, comme tend à le confirmer le résultat que l'on obtient en supprimant les des mots répertorié dans la liste de mots vides ($F_1 score$ de 0.796 contre 0.815 pour 1000 mots considérés). Suite à cette expérience, nous avons fixé k à 20, pour lequel la plupart des mots vides disparaissent du lexique. Notons que la remonté pour $k = 20$ est due à l'exclusion du mot « émission » spécifique au sous-domaine d'apprentissage.

Sélection	tf	$max_c tfidf$	$max_c le$	δ_{tf}	$\delta_{tf} * idf$	δ_{le}	$\delta_{tf} * \rho_{20}$
$F_1 score$	0.688	0.736	0.729	0.763	0.773	0.766	0.762

FIGURE 5 – Résultats de classification sur le corpus TV en utilisant des lexiques (de taille 100) construits avec les sélections décrite dans la sous-section 2.2.2 auxquels nous ajoutons δ_{le}, $\delta_{tf} * idf$ qui sont des hybrides par δ des formules $LogEntropy$ et $TFIDF$.

k	1	10	20	30	40	50	60
$F_1 score$ pour $\delta_{tf} * \rho_k$	0.815	0.805	**0.807**	0.797	0.795	0.757	0.756

FIGURE 6 – Résultats de classification sur le corpus TV en utilisant différents coefficients d'exclusions des mots trop fréquents pour un lexique de 1000 éléments. Remarquons que l'on obtient un $F_1 score$ de 0.796 lorsqu'on utilise une liste de mots vides.

Étude de l'influence de la taille du lexique Plus nous augmentons le nombre d'éléments dans le lexique, plus nous avons de chance de représenter correctement les documents. Les algorithmes d'apprentissage savent la plupart du temps exclure les dimensions inutiles. Nous considérons qu'une bonne sélection du lexique converge vers de bons résultats plus rapidement qu'une mauvaise dans notre processus de validation. Sur la figure 7, on constate que les 20 mots les plus fréquents ne sont pas les plus informatifs (courbe tf), en effet ces mots (qui sont sûrement des mots vides) sont très présents dans les deux classes considérées. Les courbes de $max_c tfidf$ et $max_c le$ (très proche l'une de l'autre) ne font pas beaucoup mieux. Ces courbes soulignent que l'utilisation de δ offre de bons résultats, cela confirme que δ_{tf} permet une bonne analyse des mots discriminants dans le corpus. Nous adaptons donc $tfidf$ et le à δ (figure 8) et remarquons que δ_{tf}, $\delta_{tf} * idf$, δ_{le} offrent des résultats presque identiques et font tous apparaître les mots `tres, bien, emission, bonne, adore, nul, tele, faire, interet, beaucoup` mais aussi des mots vides tels que `j, et, un, n`. $\delta_{tf} * \rho_{20}$ faits disparaître ces mots vides et apparaître d'autres mots tels que `super, sujets, dommage, quoi`, mais laisse disparaître le mot `tres` qui accompagne souvent `bonne` dans l'opinion fréquente « très bonne émissions ». Pour la suite des expérience nous utilisons la sélection δ_{tf}.

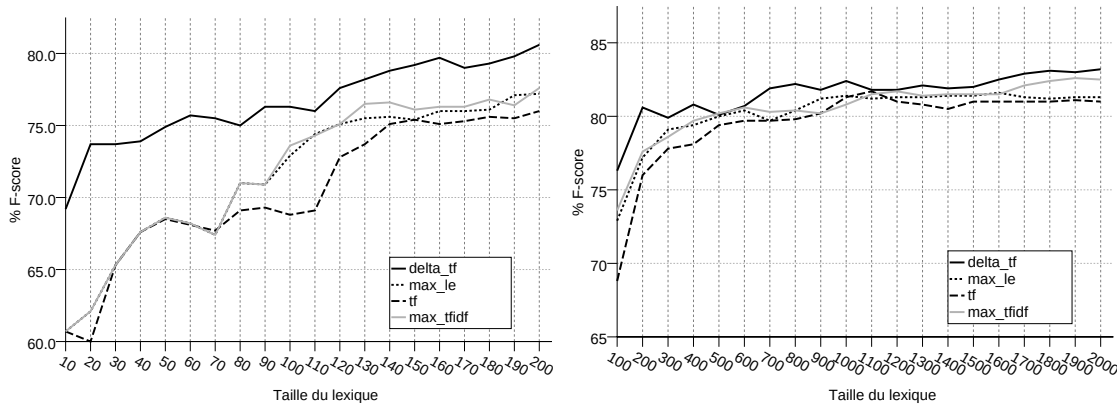


FIGURE 7 – Courbes des résultats de classification en utilisant les lexiques construits avec les selections statistiques tf, $max_c tfidf$, $max_c le$ et δ_{tf}. Les courbes de gauches sont construites sur de petits lexiques et celles de droite sur de grands lexiques.

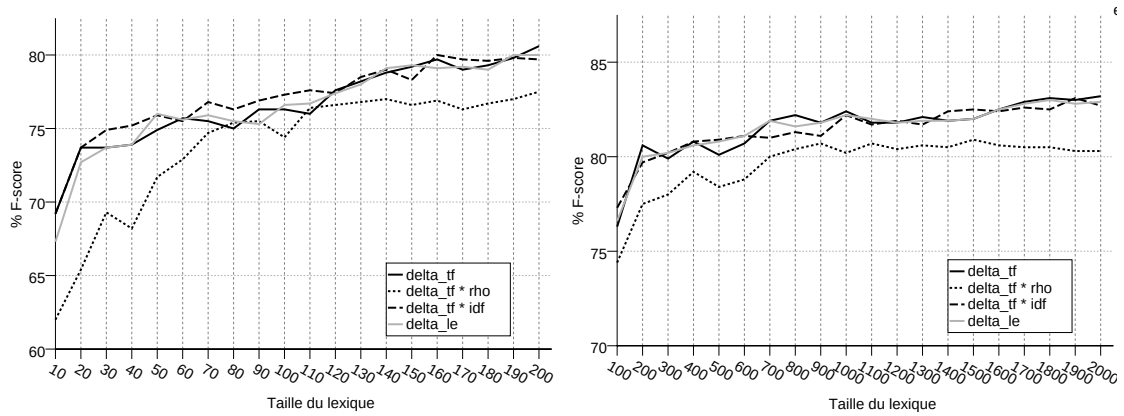
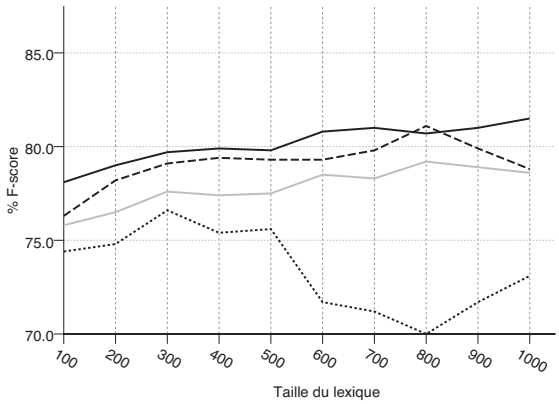


FIGURE 8 – Courbes des résultats de classification en utilisant les lexiques construits avec les selections statistiques δ_{tf}, $\delta_{tf} * idf$, δ_{le} et $\delta_{tf} * \rho_{20}$. Les courbes de gauches sont construites sur de petits lexiques et celles de droite sur de grands lexiques.

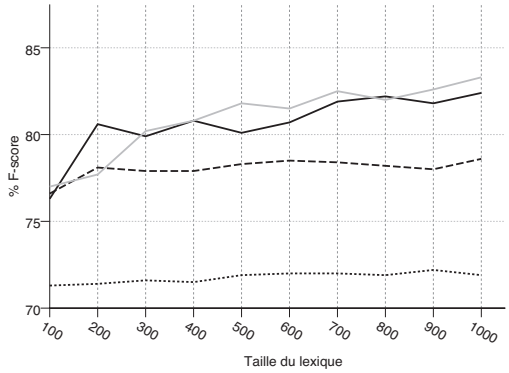
Étude de l'influence de réductions lexicales et grammaticales Une réduction lexicale telle que la *lemmatisation* permet de généraliser les mots, mais en contrepartie elle engendre une perte d'information telle que la façon de parler. En effet, la *lemmatisation* engendre une perte d'information sur le sujet et sur le temps de certains verbes. Nous remarquons qu'une perte d'information sur le temps peut avoir des incidences sur la polarité de l'opinion au même titre que la négation. Deux phrases comme « je pensai que ce présentateur était honnête » et « je pense que ce présentateur est honnête » ont la même représentation, alors que dans la première phrase l'auteur à tendance à émettre un avis opposé à celui de la seconde phrase. La perte d'information sur le sujet se caractérise par la perte de certains raccourcis que notre méthode pourrait assimiler. Par exemple, lorsqu'on considère les quatre phrases « j'aime », « j'adore », « je déteste » et « je n'aime pas », on remarque que `"j"` peut-être un raccourci pour une opinion positive, et `"je"` un raccourci pour une opinion négative.

Lorsqu'on limite les « parties de discours », la perte doit être encore plus importante. Notons d'abord que les pronoms sont de bons indicateurs de subjectivité (d'où leur sélection parmi les mots les plus représentatifs d'une opinion selon δ_{tf}) et que leur exclusion génère un perte d'information assez importante. Aussi, bien que les adjectifs soient de bons indicateurs de sentiment lorsqu'ils sont placés dans un contexte (i.e. associés à une caractéristique de l'entité (Liu (2010)), à un adverbe (e.g. « très ») ou une négation), hors contexte leur valeur est limité (Dave et coll. (2003)). Si on ajoute aux adjectifs, les verbes (e.g. « aimer » et « détester »), et les adverbes (e.g. « beaucoup »), les résultats doivent être meilleurs grâce d'une part à l'apport implicite d'un contexte, et d'autre part à la moins grande diversité des verbes (e.g. aimer et détester) et adverbes associés à un sentiment.

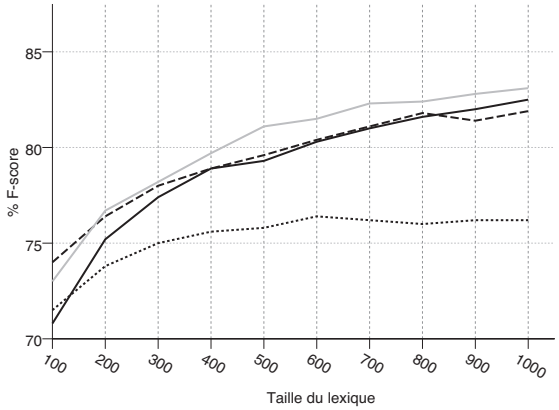
Nous expérimentons ici les influences qu'ont ces réductions lexicales et grammaticales en terme de résultats. Dans la figure 9, nous avons testé les lexiques sur à la fois sur le corpus qui a servi les apprendre (résultats à gauche) et sur le corpus de test (résultats à droite), nous observons que notre réflexion sur les adjectifs s'avère assez juste. Dans la première courbe, nous observons une baisse du $F_1 score$ pour la représentation avec des adjectifs lorsqu'on considère entre 600 et 800 éléments dans le lexique, pendant que les résultats s'améliore pour la représentation qui y ajoute les verbes et adverbes. L'idée de contexte transparaît assez bien à travers



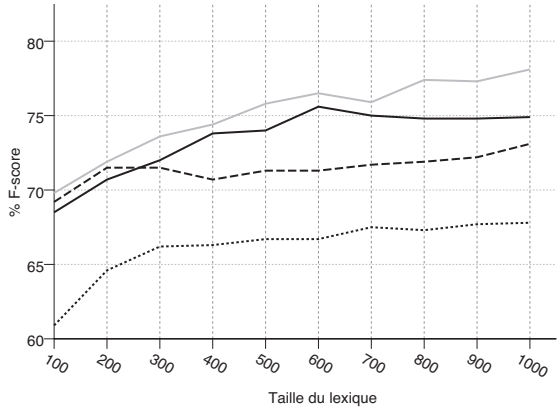
(a) lexique *émissions-tv* vers *émissions-tv*



(a') lexique *émissions-tv* vers *séries-tv*



(b) lexique *Notes générales* vers *Notes générales*



(b') lexique *Notes générales* vers *Scénario*

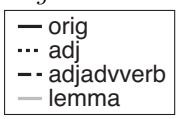
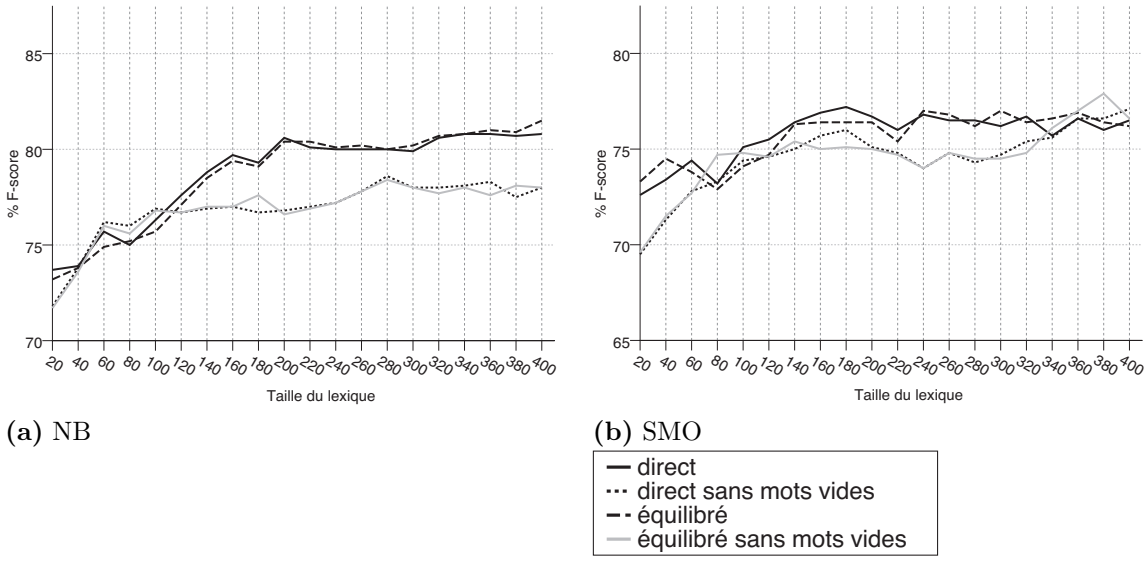


FIGURE 9 – Courbes des résultats de classification en utilisant les lexiques construits avec les sélections statistiques δ_{tf} sur les corpus TV et JV. Ces courbes permettent d'évaluer l'influence des réductions lexicales et grammaticales sur la classification. Nous considérons qu'une représentation générique doit permettre de bons résultats (ou en l'occurrence de meilleurs résultats que les autres) lorsqu'on évalue cette représentation sur le corpus qui a servi à apprendre le lexique que sur le corpus qui sert à le tester.

ces résultats : cette variation correspond à l'introduction d'adjectifs tels que « difficile » qui peuvent être utilisé autant pour un propos objectif (e.g. « les reporters tournent dans des conditions difficiles ») que pour un propos négatif (e.g. « difficile d'accrocher à cette émission »). On constate que la représentation lemmatisée offre de bons résultats lorsqu'on change sensiblement de domaine, tandis que les utilisations des mots originaux et du trio adjectifs-adverbes-verbes représentent mieux les corpus utilisés pour apprendre les lexiques. De plus, on observe que les représentations qui ignorent les noms et pronoms personnels sont les deux pires (adjectifs et adjectifs-adverbes-verbes, voir à droite dans la figure 9). Liu (2010) admet clairement que les noms ont une importance capitale afin d'apporter un contexte aux adjectifs. Dans la figure 11,

où sont exclus entre autres les pronoms personnels, on constate aussi de moins bon résultats. Les conclusions que l'on peut tirer de ces expériences sont l'importance du contexte (dont les noms et les pronoms personnels). Les bons résultats de la *lemmatisation* indiquent que ce traitement permet une bonne adaptabilité au contexte. Pourtant, pour la suite des expériences, nous continuons à utiliser les mots originaux, afin de ne pas être dépendant d'un pré-traitement utilisant des ressources *a priori*.

Étude de l'influence des mots vides Dans la plupart des expériences de classification d'opinions utilisant des uni-grammes, les mots vides sont systématiquement retirés du lexique. Il est considéré que ces mots n'apportent pas d'informations importantes pour la classification d'opinions. Nous pensons, au contraire, que les mots vides peuvent être des indices génériques pour la fouille d'opinions et cherchons à éliminer des mots trop présents dans le corpus tels que « émission » et dont le sens est trop spécifique au domaine. L'un des effets recherché du score que nous utilisons ($\delta_{tf} * \rho$, voir sous-section 2.2.2) est l'exclusion des mots vides qui n'aident pas à la classification d'opinions. Dans la figure 11, on compare un lexique où les mots vides sont exclues à partir de ressources *a priori* (liste de mots vides pour le français), avec un lexique où ils sont exclus de manière statistique ($\delta_{tf} * \rho$), et un lexique où les mots vides sont conservés. On observe que notre pondération ρ_{20} offre les moins bon résultats, néanmoins, lorsque taille du lexique augmente, ces résultats converge vers de meilleurs résultats qu'avec exclusion des mots vides à partir de ressources *a priori*. Les résultats obtenus lorsqu'on conserve les mots vides confirment l'avis d'utilité des mots vides pour la classification d'opinions.



(a) NB **(b)** SMO

FIGURE 10 – (TV) Courbes des résultats de classification en utilisant les lexiques construits avec δ_{tf}, le trait de classification est la présence et les apprenants sont NB (a) et SMO (b), ces courbes montrent l'influence l'équilibrage du lexique .

Étude de l'influence de l'équilibre du lexique Pendant la classification, chaque élément du lexique correspond à un indice positif ou négatif. Il n'est pas impossible que, au moment de la sélection du lexique, une majorité d'éléments soit des indices plus relatifs à une classe qu'à

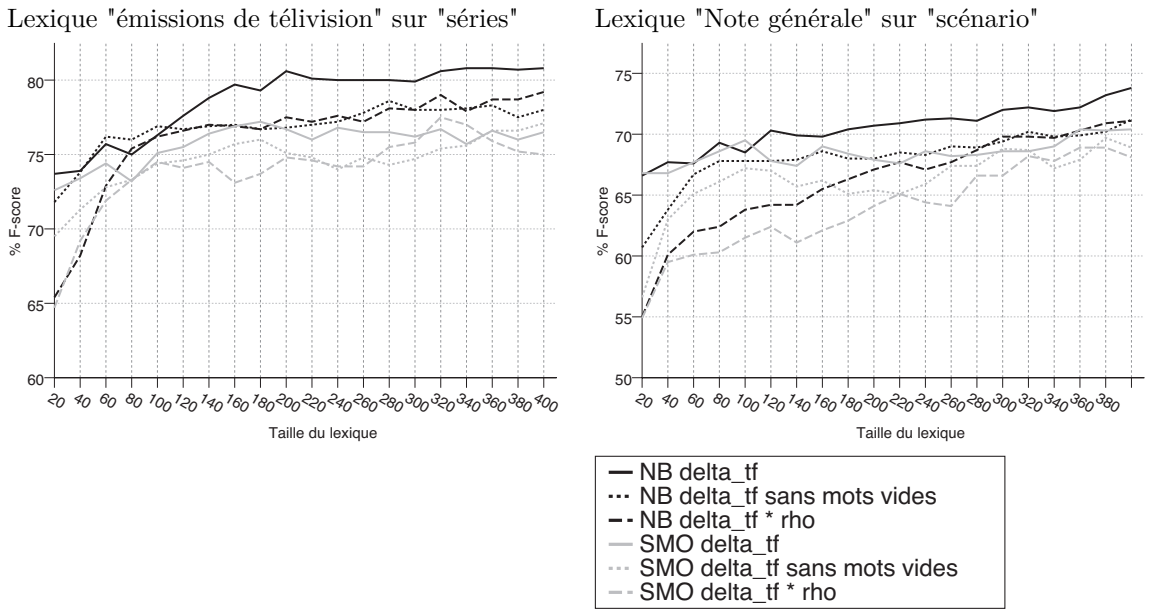


FIGURE 11 – Courbes des résultats de classification en utilisant les lexiques construits avec δ_{tf}, le trait de classification est la présence et les apprenants sont NB (en noir) et SMO (en gris), ces courbes montrent l'influence des mots vides.

une autre. Nous nous sommes posé la question de l'influence d'un équilibrage du lexique. Afin d'équilibrer ce lexique, les éléments sont associés à la classe dans laquelle ils apparaissent le plus fréquemment, et sont donc pris de manière à répartir équitable les indices selon la classe. Les résultats, visibles sur la figure 10, montrent que cette influence est négligeable. Rappelons que les corpus sur lesquels nous avons appris nos lexiques sont équilibrés entre les classes, nous n'assurons donc rien sur l'importance de cette équilibre dans le cas où des exemples positifs et négatifs ne sont pas équitablement distribués.

Utilisation des bi-grammes Pang et coll. (2002) indiquent que l'utilisation de bi-grammes, n'apportent pas de meilleur résultat qu'avec des uni-grammes. Nous vérifions cela sur nos corpus. Les bi-grammes apportent l'information de la séquentialité en contrepartie d'une augmentation exponentielle du nombre d'éléments à considérer (le sous-corpus *émissions-tv* contient 8000 uni-grammes contre 42000 bi-grammes, et le sous-corpus *Notes Générales* contient 10000 uni-grammes contre 60000 bi-grammes). La table 1 offre un comparatif de l'utilisation des bi-grammes et des uni-grammmes avec un même pourcentage d'éléments utilisé, en faisant intervenir le paramètre le ponctuation. Nous nous intéressons principalement au cas où l'on ne considère pas la ponctuation. On constate que sur le corpus *ParlonsTV*, les uni-grammes offrent de bien meilleurs résultats tandis que les bi-grammes. Le lexique de bi-grammes construit présentent certaines redondances telles que `-_DEBUT__tres-` `-tres_bien-` et `-bien__FIN_-` qui au lieu d'ajouter des informations, fait perdre la généralisation de l'adjectif « bien ». Sur le corpus *JEUXVIDEO.COM*, les résultats entre uni-grammes et bi-grammes sont très proches. Nous expliquons cela par la diversité de langage qu'offre le corpus *JEUXVIDEO.COM* ; les opinions y sont, en effet, dites de manière implicite et sont centrées sur un avis plus objectif. À partir de nos résultats et ceux de Pang et coll. (2002); Dave et coll. (2003), on peut formuler l'hypothèse (à

vérifier) que l'utilisation des uni-grammes capte bien les opinions personnelles où le critique laisse apparaître ses sentiments (e.g. « J'adore cette émission »), et que l'utilisation des bi-grammes peut-être utilisé sur des corpus où les critiques évitent de formuler leur opinion personnelle (voir exemple 3), ce qui est souvent le cas des critiques professionnels. Nous déconseillons les bi-grammes à cause de la redondance d'information et la perte de généralisation que nous avons constaté sur le corpus *ParlonsTV*.

Pour ce qui est de l'influence de la ponctuation, on remarque des variations non significatives des résultats, bien que les ponctuations « . », « ; » et « ? » figurent parmi les 10 éléments (uni-grammes) les plus discriminants dans le corpus *ParlonsTV*. Nous considérons que seul le point d'interrogation à une orientation (négative), et pensons que l'apparition des autres ponctuation est liée à des circonstances telles que l'oubli de ponctuation dans les critiques les plus virulentes.

Ratio du lexique	unigram ou bigram	ponctuation	corpus	nombre d'éléments	$F_1 score$
0.02	bigram	sans	TV	424	0.756
		avec		432	0.756
0.04	bigram	sans	TV	847	0.779
		avec		864	0.772
0.04	unigram	sans	TV	339	0.806
		avec		339	0.811
0.04	bigram	sans	JV	2401	0.731
		avec		2325	0.748
0.04	unigram	sans	JV	399	0.739
		avec		399	0.738

TABLE 1 – Résultats de classification en utilisant des bi-grammes

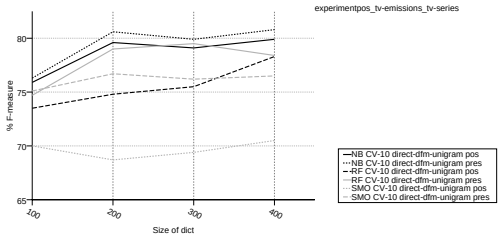
Dans la sous-section précédente nous avons explorer le problème de la sélection du lexique. Nous achevons notre exploration du problème de représentation en nous demandant quels sont les traits (parmi ceux que nous avons décris dans 2.2.3) les plus efficace pour représenter une opinion.

Quel trait donne les meilleurs resultats ? Nous finalisons notre problème de représentation du document en nous intéressant aux traits de classification qui offre les meilleurs résultats. Nous expérimentons les traits décrits dans 2.2.3 [que nous nommons ici *pres* (présence de l'élément dans le document), *freq* (fréquence de l'élément dans le document) et *pos* (positions auxquelles apparaissent le terme)] avec les apprenants *SMO* (Séparateur à Vaste Marge), *NB* (*Naive Bayes*) et *RF* (forêts aléatoire).

Dans les figures 12 et 13, le constat est unanime (même si l'efficacité du trait est liée au choix du classificateur), la présence surpasse les autres de traits de classification (Pang et coll. (2002)). Rappelons ici, les résultats de la première expérience utilisant d'autres attributs que des mots où la fréquence est le trait à privilégier.

Lors des expériences précédentes sur la représentation des données, nous apprenons une représentation (un lexique que nous représentons avec les bons traits) sur un sous-domaine (e.g. Note Général) et testons leur efficacité par validation croisée sur un autre sous-domaine (e.g. Scénario). Dans l'expérience suivante nous souhaitons juger l'apprenant le plus efficace pour un tel changement de sous domaine, au lieu de faire une validation croisée sur le corpus de test, nous apprenons ici un modèle sur le corpus d'entraînement que nous validons par test sur l'autre corpus.

Lexique "émissions de télivision" sur "séries"



Lexique "Note générale" sur "scénario"

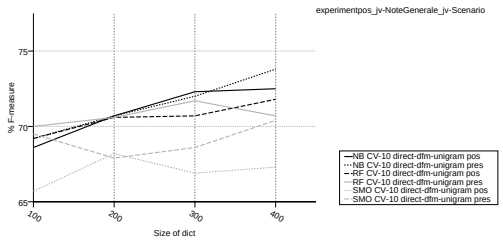
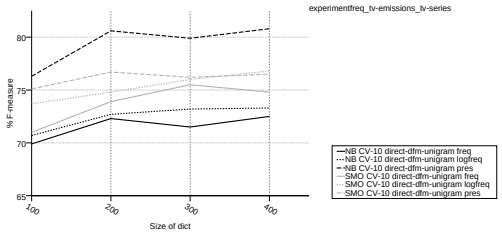


FIGURE 12 – pos (TV) Courbes des résultats de classification en utilisant les lexiques construits avec δ_{tf}, le trait de classification est la présence et les apprenants sont NB (a) et SMO (b), ces courbes montre l'influence l'équilibrage du lexique .

Lexique "émissions de télivision" sur "séries"



Lexique "Note générale" sur "scénario"

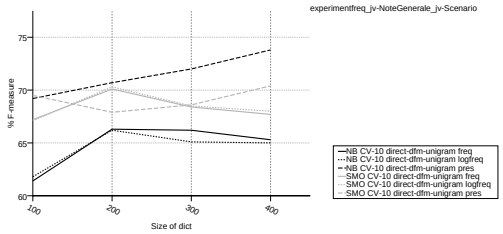


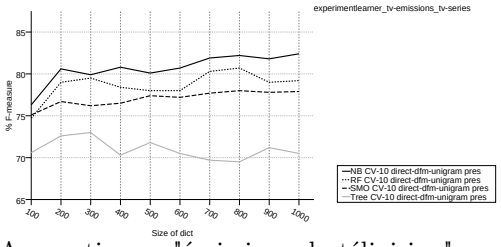
FIGURE 13 – freq (TV) Courbes des résultats de classification en utilisant les lexiques construits avec δ_{tf}, le trait de classification est la présence et les apprenants sont NB (a) et SMO (b), ces courbes montre l'influence l'équilibrage du lexique .

2.3.3 Apprenant inter-domaine

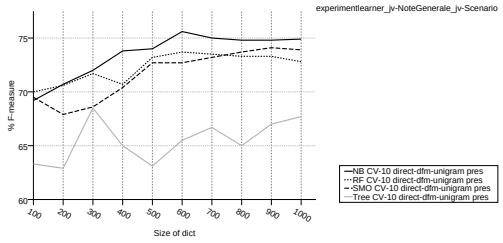
Dans le domaine de la fouille d'opinions, les Séparateurs à Vaste Marge (SVM) ont la réputation de donner de bons résultats (Pang et coll. (2002)). Nous souhaitons dans cette expérience le vérifier, car ces classificateurs sont plus lents que *Naive Bayes* ou que les forêts aléatoires.

L'expérience confronte les classificateurs *SMO* (une implémentation de SVM), *NB* (*Naive Bayes*), *RF* (forêts aléatoires) et *C4.5* (une implémentation populaire des arbres de décisions). La figure 14 nous montre les courbes des résultats de classification en fonction de la taille du lexique. Nous mettons en correspondance les résultats obtenus en apprenant le modèle sur le corpus de test (comme dans les expériences sur la représentation), avec les résultats qui nous intéressent (apprentissage "intégral" du modèle sur l'ensemble d'entraînement). On remarque d'abord l'allure décroissante des courbes qui signifie que les éléments du lexique ont des valeurs (en terme de polarité) sensiblement différentes d'un sous-domaine à un autre. Ceci révèle bien un problème de portabilité. Les meilleurs apprenants dans ce test de portabilité sont *SMO* et *RF* tandis *NB* donne de meilleurs résultats dans un domaine fixe. La réputation des Séparateurs à Vaste Marge semble donc bien fondée. Globalement, les forêts aléatoire offre de bons résultats qui passent bien le domaine. Ceci s'explique facilement : le problème lors de l'apprentissage d'un modèle est le choix des dimensions dont l'importance varie selon le domaine, les forêt aléatoires font ce choix de manière aléatoire, le modèle est donc peu dépendant du corpus d'apprentissage.

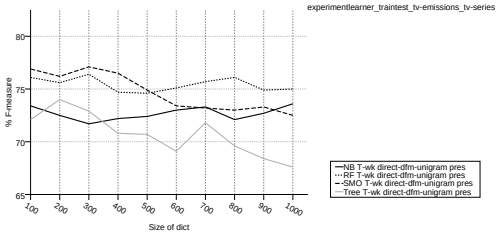
Représentation "émissions de télivision" sur "séries"



Représentation "Note générale" sur "scénario"



Apprentissage "émissions de télivision" sur "séries"



Apprentissage "Note générale" sur "scénario"

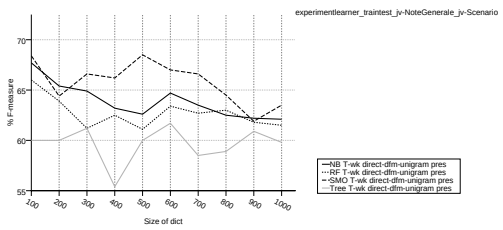


FIGURE 14 – Quel est le meilleur apprenant ? (légende : NB en trait plein, RF en petit pointillés, SMO en pointillés plus large et C4.5 en gris)

Influence des notes moyennes Le but de notre classificateur est de séparer les données en deux classes. Les données à la limite des deux classes introduisent des données ambiguës qui perturbent le classificateur. Nous observons les variations du $F_1 score$ obtenu sur le corpus de base, et modifions les corpus en excluant des documents. Pang et Lee (2004) utilisent un détecteur de subjectivité pour améliorer l'apprentissage de leur modèle, ici, l'expérience ne s'intéresse pas à l'exclusion de documents objectifs (ou factuels) mais à l'exclusion des opinions mitigées (ou neutre). Pour cela, nous excluons les notes moyennes du corpus. Dans la table 2, on constate d'abord la réduction du corpus introduit par ce filtrage des notes moyennes. Pour les données de *JEUXVIDEO.COM*, cela a de lourdes conséquence : lors de l'exclusion des notes de 8 à 12, le nombre de documents d'entraînement est divisé par trois. Lorsqu'on exclut la note 10, le résultat ne changent pas pour le corpus TV (où, nous le rappelons, la majorité des notes se trouvent sur les extrêmes), alors que celui pour le corpus JV diminue de 4%. Ceci peut s'expliquer par le fait qu'un annotateur a tendance à considérer cette note comme une borne supérieure à la classe négative (c'est-à-dire qu'on a exclu un grand nombre d'exemples négatifs). L'exemple 3 est une opinion notée 10 issue du corpus JV. On remarque bien que le critiques met en avant les défauts du produit. Pour des expériences similaires au nôtres, nous encourageons à considérer cette note comme reflétant un avis positif. Lorsqu'on exclut les notes de 8 à 12, on obtient des résultats sensiblement meilleurs (gain de 4% pour le corpus TV et 2% pour le corpus JV, ce dernier est discutable, vu de la diminution du corpus de test). On constate avec cette expérience que l'exclusion des avis mitigés (pour une classification en deux classes *positif* et *négatif*) n'est pas nécessaire avec un système de notation en général assez bien compris des critiques tel que la notation sur 20.

Dans cette partie, nous avons d'abord utilisé des indices simples tels que les émoticônes pour classer les opinions. Ensuite, nous avons exploré le problème de la sélection du lexique et mis en avant l'efficacité de la sélection δ_{tf}. Nous avons évaluer la présence d'un élément du lexique

« Kriss Kross **ne révolutionnera pas** le concept du jeu de cartes. Il apporte, certes, un peu d'originalité et de fraîcheur, mais **n'est pas d'un intérêt suffocant**. **Le pire** dans tout ça, c'est quand même **le prix** : 179 F, c'est carrément **trop cher**, surtout qu'il y a quelques **bugs**. »

Exemple 3 – Avis d'un critique de jeux vidéo, dont la note est 10. Les groupes de mots qui reflète un avis négatif sont mis en gras.

Notes exclues	Corpus	Nombre d'exemples	Nombre de textes à classer	$F_1 score$
aucune	TV	1394	936	0.734
aucune	JV	2980	900	0.677
10	TV	1216	808	0.734
10	JV	2150	568	0.633
8, 9, 10, 11, 12	TV	1158	722	0.769
8, 9, 10, 11, 12	JV	1098	258	0.697

TABLE 2 – Résultats de classification en fonctions des notes exclues du corpus, en utilisant un lexique de 100 éléments (uni-grammes sélectionnés avec δ_{tf}), et un modèle de classification (*Naive Bayes*) appris sur les exemples. Après exclusions des notes moyennes, d'autres notes sont exclues afin de conserver l'équilibre entre les deux classes.

comme étant le meilleur trait de classification. Et dans la sous-section précédente, nous avons mis en avant le problème de portabilité entre des sous-domaines (e.g. *Notes Générale* et *Scénario* appartenant tout deux au domaines des jeux vidéo). Nous avons mis en valeur que les SVMs (Séparateur à Vastes Marges) et les forêts aléatoires offre les meilleurs résultats dans ce problème et donc mis en avant leur généricité.

Rappelons que tous ces tests ont été faits grâce à des exemples annotés en utilisant les notes données par les auteurs des critiques. Pang et Lee (2008) indiquent que cette méthode d'annotation est biaisée par la subjectivité du critique, une annotation manuelle faite par des gens de confiance est préférable, néanmoins l'annotation manuelle est un processus coûteux.

Dans la partie suivante, nous tentons de réduire le coût d'annotation en d'exploiter les exemples non-annotées. La technique employé est l'*active learning* et part du principe qu'avec un minimum de connaissances, on peut atteindre, voire dépasser les résultats que l'on obtiendrai en ayant ingurgité la totalité des connaissances.

3 Active Learning

3.1 État de l'art

L'annotation manuelle de textes d'opinions peut être un processus extrêmement coûteux. En effet, une annotation fiable nécessite un accord entre plusieurs personnes. L'idée est donc de profiter des documents non annotés pour en tirer un maximum d'informations. Dans une approche *active learning*, un algorithme demande à un expert d'annoter les exemples dont il a le plus besoin pour apprendre. Ces exemples sont choisis de manière à réduire le nombre de données annotées. Il existe plusieurs types d'*active learning* (Dagan et Engelson (1995)), nous considérons le cas où l'apprenant choisit des exemples parmi un ensemble de données non annotées (*selective sampling*), on parle dans ce cas de *pool-based active learning*.

Le principe d'un algorithme d'*active learning* est le suivant. On commence par un ensemble

(graine) d'exemples annotés, bien que cet ensemble initial ne soit pas essentiel pour appliquer l'algorithme. Ensuite le processus est itératif : à chaque itération un (Baram et coll. (2004)) ou plusieurs (*batch mode active learning* (Tong et Koller (2002); Zhu et coll. (2008); Boiy et Moens (2009); Hoi et coll. (2009))) exemples sont choisis, puis annotés par des humains et ajoutés à l'ensemble d'apprentissage afin de réentraîner le classificateur jusqu'à ce qu'une condition d'arrêt soit atteinte. Les deux points clés de l'*active learning*, sont la sélection des exemples (qui doivent être les plus informatifs pour le classificateur) et la détermination du critère d'arrêt (Zhu et coll. (2008)). Tandis que le second point est souvent ignoré, le premier a été largement exploré.

3.1.1 Méthodes de sélection des exemples

Le problème de la sélection des exemples les plus informatifs pour le classificateur connais plusieurs solutions. Les techniques les plus adaptées à notre problème sont l'*uncertainty sampling* (les exemples choisis sont ceux pour lesquels le classificateur est le moins certain), le *committee-based sampling* (les exemples choisis sont ceux qui génèrent un désaccord entre plusieurs classificateurs) et le *Kernel Farthest First* (les exemples choisis sont ceux les plus éloignés des données existante). Aussi certains auteurs combinent plusieurs méthodes afin d'améliorer la sélection des exemples.

Uncertainty sampling Cette technique concerne la sélection des exemples à annoter pour lesquels le classificateur utilisé est le moins sûr. On attend de l'annotation de ces exemples que les exemples similaires seront mieux classés. La mesure de l'incertitude (*uncertainty* en anglais) dépend du classificateur utilisé. Cette mesure est simple à acquérir pour des classificateurs probabilistes, ainsi que pour les arbres de décisions, grâce à l'indication du degré de certitude pour chaque exemple apportée par ces classificateur. Pour les Séparateurs à Vaste Marge, l'incertitude peut-être mesurée comme la distance d'un exemple à l'hyperplan séparateur (Tong et Koller (2002)). En ne sélectionnant pas les exemples pour lesquels le classificateur est déjà sûr, on diminue les redondances d'informations. Pour résumer, une sélection par incertitude donne la priorité à l'apprentissage soit des exemples les moins similaires à ceux présents dans le corpus, soit des exemples ambiguës. Cette stratégie a donc pour rôle d'explorer le corpus.

Une méthode de sélection opposée, appelé *relevance sampling* (le principe est d'ajouté les exemples pour lesquels le classificateur prétend être le plus sûr), est utilisée par Boiy et Moens (2009) afin de préciser la classification.

Committee-based sampling L'idée est la même que pour l'*uncertainty sampling*, on donne la priorité au données ambiguës. La principale différence est que la sélection des ces données est faite par un consensus de classificateurs (Dagan et Engelson (1995)). La mesure que nous utilisons ici est le désaccord entre les classificateurs.

Kernel Farthest First Une alternative aux algorithmes utilisant les résultats de classifications est l'approche géométrique dans un hyper-espace. Dans la méthode du *kernel farthest first* (Baram et coll. (2004)) l'exemple le plus éloigné de l'ensemble des exemples déjà annotés est choisi pour être annoté. Cette méthode est utilisée par Baram et coll. (2004); Boiy et Moens (2009) pour faire de l'*active learning* avec *SVM*.

Combinaisons de plusieurs méthodes Dans la littérature, on constate que certains algorithmes de sélection des exemples sont mieux adaptées à un problème qu'à un autre (Baram et coll. (2004)). Boiy et Moens (2009) combinent le *relevance sampling* et l'*uncertainty sampling*, afin de conserver un équilibre entre les classes grâce à la connaissance apportée par le *relevance sampling* tout en ajoutant des exemples incertains. Baram et coll. (2004) combinent plusieurs

algorithmes (chacun spécialisés pour un problème différent) en utilisant un algorithme de *Multi Armed Bandit* (Auer et coll. (2002) ; le principe de cet algorithme est de prendre en compte les avis de tous les algorithmes et de récompenser ceux qui offre une meilleure répartition des classes) pour obtenir un algorithme résolvant tous les problèmes.

3.1.2 Évaluation des méthodes

Afin d'évaluer l'*active learning*, on compare généralement les résultats obtenus par une sélection aléatoire des exemples avec ceux obtenus avec la méthode de sélection testée. La figure 15 illustre les courbes théorique des résultats obtenus en faisant une sélection aléatoire (PASSIVE) et ceux obtenus avec la méthode d'*active learning* (ACTIVE). Quelques auteurs (e.g. Tong et Koller (2002)) testent la précision obtenue pour un nombre fixé d'exemples ou d'itérations. D'autres auteurs (e.g. Roy et McCallum (2001)) se contente de visualiser la courbe des précisions obtenues en fonction du nombre d'exemples ajoutés pour démontrer les avantages de leur methode en terme de vitesse d'apprentissage. Baram et coll. (2004) propose une mesure qui permet d'évaluer l'efficacité d'une méthode de sélection des exemples sur l'ensemble des itérations. Cette mesure, illustrée dans la figure 15, quantifie la déficience d'une fonction de requête par rapport aux résultats obtenus avec l'aléatoire (correspondant à la sélection passive standard). Étant donné un ensemble d'exemples non-annotés échantillonnés en n instances. Soit PASSIVE la sélection aléatoire et ACTIVE la méthode de sélection à évaluer. Pour chaque $1 \leq t \leq n$, on définit $Score_t(PASSIVE)$ et $Score_t(ACTIVE)$ comme étant la moyenne des performances obtenues par les sélections correspondantes d'un ensemble d'apprentissage de t instances.

$$Def_n(ACTIVE) = \frac{\sum_{t=1}^{n}(Score_n(PASSIVE)) - Score_i(ACTIVE))}{\sum_{t=1}^{n}(Score_n(PASSIVE)) - Score_i(PASSIVE))}$$

La déficience mesure la performance globale d'un *active learner* jusqu'à l'instance n. Dans la figure 15, la déficience correspond à au rapport de surfaces $\frac{B'-B"}{A}$. Le numérateur est l'aire entre le plafond et la courbe des scores obtenus avec l'*active learner*, en considérant la surface surface sous le plafond comme positive et la surface au-dessus comme négative. Le dénominateur permet de rendre une score moins dépendant au problème. Baram et coll. (2004) ne mettent pas assez en avant que l'active learning, au delà de réduire le nombre d'exemples à annoter, aide aussi à améliorer les performances, en affirmant que la déficience est non-négative. Bien que le cas où $B' < B"$ (impliquant $B' - B" < 0$) soit quasiment impossible, nous préférons éviter une telle affirmation.

3.2 *Active Learning* pour la fouille d'opinions

À notre connaissance, seuls Boiy et Moens (2009) utilisent l'*active learning* dans un problème de fouille d'opinions assez similaire au notre. Dans notre approche, l'*active learning* est utilisé pour apprendre un modèle assez générique avec un minimum d'exemples annotés.

Nous interprétons le processus itératif de l'*active learning* de la manière suivante.

1. sélectionner les premiers exemples annotés (graine)
2. jusqu'à ce qu'un nombre maximal d'exemples annotés, répéter les étapes qui suivent :
   (a) évaluer le $F_1 score$ sur le corpus de test
   (b) utiliser l'algorithme de sélection pour prendre 20 nouveaux exemples annotés dans l'ensemble d'entraînement, ces exemples sont ajoutés à l'ensemble des données annotés
   (c) (récompenser si l'on utilise une combinaison d'algorithmes)

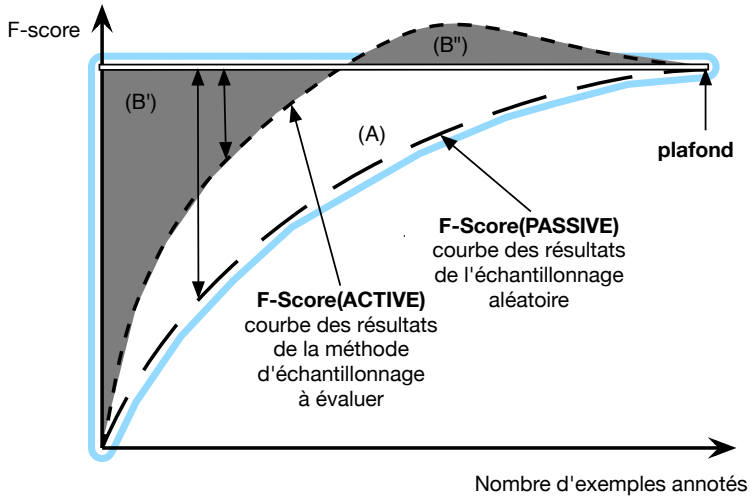


FIGURE 15 – La définition en image de déficience d'un *active learner* ou ACTIVE correspond à un bon *active learner*. L'axe des ordonnées dans l'illustration de Baram et coll. (2004) est la précision (*true accuracy* en fait), nous choisissons d'utiliser le $F_1 score$ afin d'uniformiser nos résultats. Le plafond est le score maximale obtenu par une sélection passive, et correspond à l'annotation de toute les exemples de l'ensemble des données. La déficience est définie dans ce cas comme le rapport de la surface B (caractérisée par $B' - B$") par la surface A (entourée en bleu clair).

Pour chaque itération, on sélectionne un paquet de n exemples annotés, on parle dans ce cas de *batch mode active learning*. Cette façon de faire est réputée efficace (Hoi et coll. (2009)) et est adaptée à notre problème de classification binaire.

Pour commenter les méthodes de sélections et les expériences, nous formalisons les concepts suivants. On admet que deux qualités souhaitées d'une méthode de sélection sont la rapidité avec laquelle l'apprentissage converge vers les meilleurs résultats, et le dépassement du plafond. Nous appelons ces qualités respectivement **attaque** et **supériorité**. Nous proposons de mesurer l'attaque en utilisant la déficience sur les premières itérations. Tandis que la supériorité (que nous ne mesurons pas dans ce rapport) correspond à l'aire B". Lors de la sélection des données dans un hyper-espace, on peut se représenter deux manières de choisir les données : la première consiste à choisir les données les plus éloignées afin de découvrir de nouvelles données, la seconde à choisir les données assez proches pour préciser le modèle (Boiy et Moens (2009)). Nous parlons respectivement d'**exploration** et d'**optimisation**.

3.2.1 Les méthodes de sélections

Lors de notre exploration du problème d'*active learning*, nous avons adapté les méthodes de sélection décrites dans 3.1.1.

Uncertainty sampling Nous utilisons la prédiction (comprise entre 0 et 1) de *Weka* pour les classificateurs probabilistes afin d'utiliser l'*uncertainty sampling*. Afin d'améliorer la pertinence des exemples sélectionnés, nous proposons deux méthodes :

– **équilibrage** : Boiy et Moens (2009) remarquent que l'équilibre (autant de positifs que de négatifs) des exemples choisis a une grande importance et utilisent le *relevance sampling* pour découvrir la classe des exemples et ainsi rééquilibrer le corpus d'apprentissage. Pourtant, il n'est pas souhaitable d'intégrer des exemples redondants. Au lieu de nous intéresser à l'équilibre effectif des classes, nous nous intéressons à l'équilibre des classes prédites. L'idée est qu'avec un classificateur probabiliste, si on fait une erreur de prédiction et que l'équilibre est perturbé, alors le classificateur donnera de préférence une prédiction correspondant à la classe majoritaire. Il y a donc de grande chance qu'au tour suivant les exemples pour lesquels le classificateur est moins sûr appartiennent à cette classe. Par conséquent, l'opération d'équilibrage sélectionne naturellement des exemples de la classe minoritaire dont la prédiction est plus fiable.
– **réflexion** : L'*uncertainty sampling* est sensé sélectionner des exemples que le classificateur n'est pas sûr de bien classer. Si plusieurs exemples similaires ont une faible prédiction et que l'un deux est ajouté à l'ensemble d'apprentissage, alors les autres exemples doivent être mieux prédits par le nouveau modèle. Lorsque ce n'est pas le cas et que au contraire la prédiction d'un exemple, alors il semble prioritaire d'inclure cet exemple dans l'ensemble. Soit u_n l'incertitude (1 - prédiction) d'un exemple e lors de la $n$ième itération, la formule que nous utilisons pour donner cette priorité ($0 \leq p_n \leq 1$) est la suivante :

$$p_n(e) = u_n(e) * \frac{(u_n(e) - \frac{\sum_{i=1}^{n-1} u_i(e)}{n-1} + 1)}{2}$$

Avec cette formule, nous considérons à la fois la variation de l'incertitude et l'incertitude actuelle.

Committee-based sampling La mise en oeuvre d'une sélection par consensus nécessite deux éléments clés : un comité de classificateurs et une fonction qui permet de mesurer le désaccord. Afin de constituer un comité nous faisons varié les méthodes de sélection du lexique, les traits et les apprenants. Pour quantifier le désaccord nous réutilisons la formule d'entropie (Dagan et Engelson (1995); Zhu et coll. (2008)). Soit k la taille du comité et $n_i(e)$ le nombre de membres du comité prédisant l'exemple e dans la classe i, le désaccord $D(e)$ est le suivant :

$$D(e) = -\sum_i (\frac{n_i(e)}{k} log(\frac{n_i(e)}{k}))$$

Lorsque l'on dispose de la valeur de prédiction pour chaque membre du comité, on peut utiliser cette formule en redéfinissant $n_i(e)$ comme la somme des prédictions des membres du comité prédisant l'exemple e dans la classe i.

Discovery sampling Dans notre problème nous choisissons dynamiquement les dimensions (liées au lexique) en fonction du corpus d'entraînement. Nous proposons une interprétation de *KFF* pour ce problème. La sélection *discovery sampling* choisit les documents qui contiennent les mots les plus nouveaux (qui apparaissent pour la première fois ou très rarement).

Combinaison des méthodes Notre idée de base pour la combinaison de méthodes, est de combiner les points forts de chaque algorithme dans un meta-algorithme. La combinaison idéale doit être capable de combiner l'attaque d'une méthode avec la supériorité d'une autre méthode, mais aussi de s'adapter aux problèmes que les algorithmes savent résoudre (Baram et coll. (2004)).

Lorsqu'on combine plusieurs méthodes dans notre *batch mode active learning* (sélection de plusieurs exemple), on peut : soit choisir de les exemples les plus pertinents de chaque algorithme (il s'agit plus d'un mélange que d'un combinaison), soit utiliser une politique de récompense (ou punition).

Une façon d'effectuer un mélange de x algorithmes on prend simplement les $\frac{x}{n}$ premiers exemples choisis par chaque algorithme. Lorsqu'on combine des méthodes par récompense (voir *Multi Armed Bandit*), il s'agit de déterminer le poids de chaque méthode l'instant donné, nous utilisons l'algorithme de Baram et coll. (2004) pour cela.

L'influence des premiers exemples La sélection des premiers exemples (la graine) est, dans la littérature, soit faite manuellement, soit aléatoirement. Les auteurs (e.g. Boiy et Moens (2009)) qui s'appuient sur graine manuellement choisi, tente de construire une connaissance minimale suffisante pour classificateur. Nous proposons deux approches, basée sur l'apprentissage non-supervisée, pour une sélection automatique de ces premiers exemples. La première est basée sur le *clustering* de textes, l'idée est d'avoir un ensemble d'apprentissage avec des contenus divers. Ce genre d'approche nécessite *a priori* une graine de taille assez conséquente (au moins 50) afin d'être sûr de disposer des deux classes. Si ce n'est pas le cas nous supposons que la méthode de sélection rattrapera le retard.

La seconde est basée sur les méthodes d'apprentissage non-supervisée existantes dans la littérature de la fouille d'opinions. En utilisant l'information mutuelle (Turney (2002)) de "bon" et "mauvais", on choisi des exemples dont le degré de confiance est assez élevé (e.g. « bon programme »), et on distribue les textes équitablement entre les deux classes. Cette seconde approche a pour avantage de ne pas forcément dépendre d'un annotateur pour la graine, et offre un modèle de classification juste mais minimal dés les premiers exemples.

3.3 Expériences

Dans cette partie, nous présentons nos résultats d'expériences d'*active learning* pour la fouille d'opinion. Nous explorons les méthodes sur un corpus utilisé par beaucoup de gens du domaine (Pang et coll. (2002); Boiy et Moens (2009)) et sur notre corpus *ParlonsTV* (on utilise toujours les commentaires des notes générales pour classer ceux sur le scénario). Nous évaluons les méthodes avec les concept de déficience, attaque et supériorité présentés dans les sous-sections 3.1 et 3.2. Lors de notre évaluation, l'*active learning* est effectué sur un corpus d'entraînement (où théoriquement les documents ne sont pas annotés) et l'évaluation est faite sur un corpus de test (où théoriquement les documents sont annotés) avec la sélection statistique du lexique δ_{tf} et l'apprenant *Naive Bayes* vus dans la sous-section 2.2. Le nombre de mots étant variable d'une itération à l'autre, nous choisissons une taille de dictionnaire variable (2% du nombre total de mots rencontrés pendant l'itération). Dans nos expériences, on considère qu'une attaque est bonne lorsqu'elle est proche de 0, même si cette définition semble étrange, elle permet de conserver la déficience pour les premières itérations.

Premier test avec un corpus standard Nous réutilisons le corpus de critiques de cinéma de Pang et Lee (2008). Ce corpus contiens 2000 évaluations de cinéma (1000 positives, 1000 négatives), caractérisées par un vocabulaire varié (typiquement mélange de mots qui décrivent le film et de mots qui décrivent l'évaluation de ce film). Pour évaluer, on a découper le corpus de manière équilibrée en deux corpus de 1320 documents pour l'entraînement et de 600 documents pour le test.

Nous avons d'abord comparer l'*uncertainty sampling* avec une sélection hypothétique des exemples de manière équilibrée (à chaque tour on ajoute 10 positifs 10 négatifs dans l'ordre

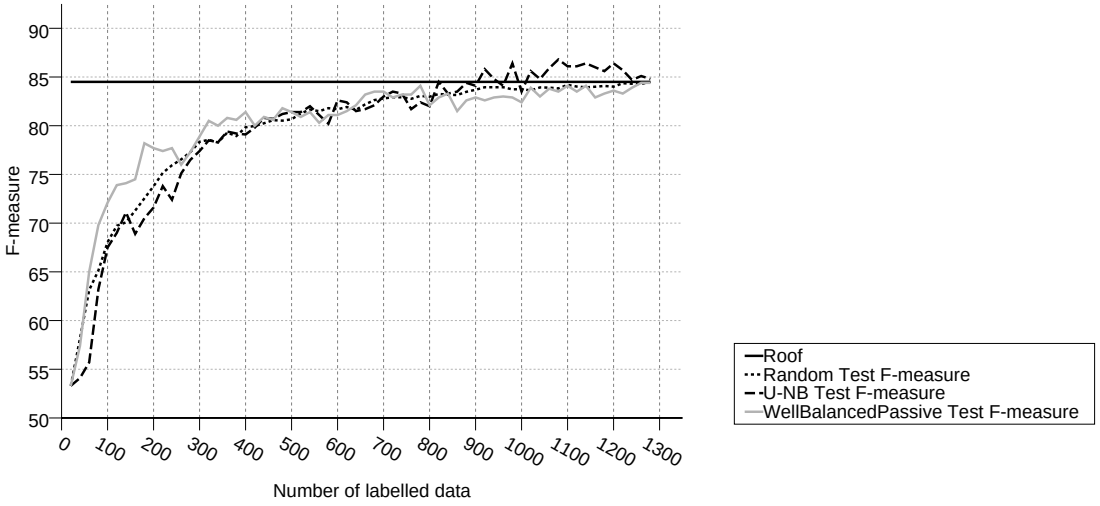


FIGURE 16 – Déficience sur le corpus du Pang (on a coupé le corpus en 1320 train 660 test)

Cinéma

Méthode	Attaque
B-Passive	0.83
U-RF	1.00
U-NBTree	1.04
CoTrait	1.04
CoLex	1.06
Dis	1.09
CoCla	1.11
U-NB	1.12
U-Tree	1.55
r*U-RF	0.91
Br*U-RF	1.16
BU-NBTree	0.93
Br*U-NBTree	0.96
BU-NB	1.06
r*U-NB	1.06
Br*U-NB	1.06
r*U-NBTree	0.97
BU-Tree	2.44
r*CoLex	0.94
r*CoTrait	0.99
r1U-NBTree	0.99
r5U-NBTree	0.99

ParlonsTV

r*U-NBTree	0.46
U-NB	0.47
BU-NB	0.68
r*Colex	0.72
Colex	0.88
U-NBTree	0.90
BU-NBTree	0.91
B-Passive	1.05
MAB	1.11
CoTrait	1.12
U-RF	1.32
Melange	1.37
Dis	1.69

TABLE 3 – Attaque (déficience pour les 14 première itérations (280 exemples)) des différents algorithmes de sélections. sur le corpus du Pang (on a coupé le corpus en 1320 train 660 test)

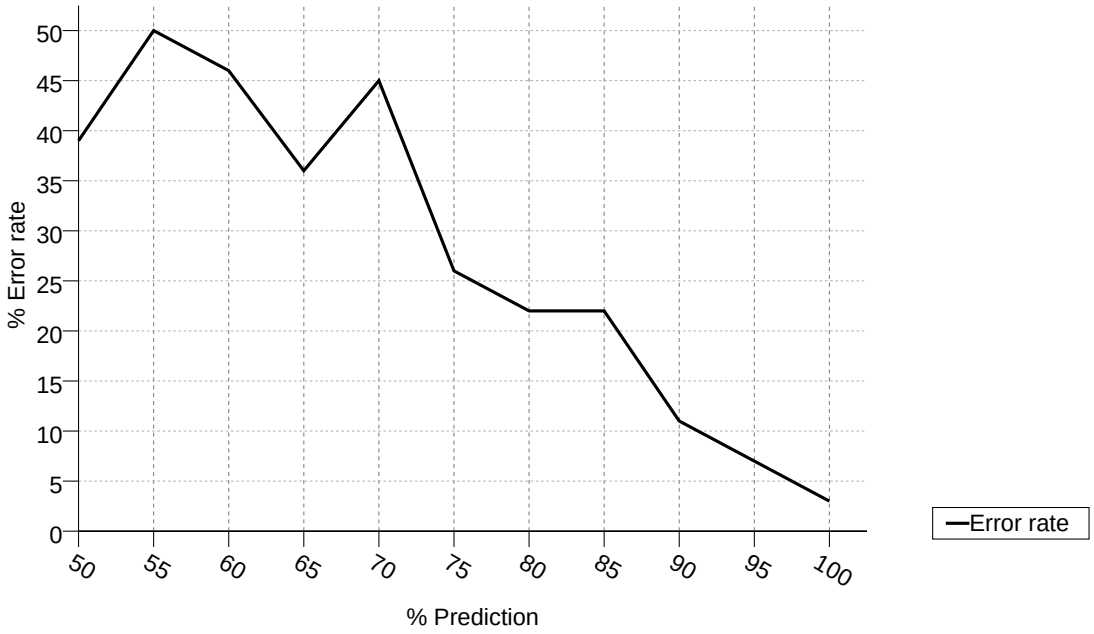


FIGURE 17 – Le taux d'erreur en fonction de prédiction de *Naive Bayes* (lexique de 100 avec deltatf)

d'apparition des critiques dans le dossier disponible pour ces critiques). Dans les expériences cette sélection est appelée *B-Passive* et nous appelons *U-NB* l'*uncertainty sampling* utilisé avec *Naive Bayes*. Il y a peu de chance pour que *B-Passive* soit la meilleure sélection, néanmoins il est très probable qu'une sélection équilibrée des documents donne de bons résultats. Ce que met en évidence la première partie (les 14 premières itérations) la figure 16, c'est l'importance de l'équilibre de données (pour notre classificateur de test en tout cas, en l'occurence *Naive Bayes*). Lors des premières itérations, l'*uncertainty sampling* semble fonctionner trés mal, en effet, la courbe se trouve sous celles des sélections passives. Néanmoins pendant les 20 dernières itérations, on constate que l'*uncertainty sampling* dépasse le plafond. Notre méthode hypothétique nous servira de *baseline*, sa déficience totale est .89 et 0.83 pour les 14 premières itérations. La déficience totale de l'*uncertainty sampling* est 0.99 et 1.12 pour les 14 premières itérations. Le but pour la suite est d'améliorer ces résultats.

Méthodes de sélection Nous évaluons ici nos méthodes et meta-méthodes de sélection. Dans l'expérience, nous nommons *U-NB*, *U-RF*, *U-Tree*, *U-NBTree* les méthodes d'uncertainty sampling en utilisant les prédictions respectives des classificateurs *Naive Bayes*, forêts aléatoires, *C45* (arbre de décision) et *NBTree* (voir 2.2.4) ; les méthodes de *committee-based sampling* commence par *Co*, nous utilisons *CoLex* (qui confronte des classificateurs qui utilisent δ_{tf}, $max_c logentropy$ et $max_c tfidf$), *CoTrait* (qui confronte la présence et la fréquence) et *CoCla* (qui confronte *Naive Bayes*, *SMO*, et les forêts aléatoire) ; *Dis* correspond au *Discovery sampling*. Les méthodes dont le nom commence par *B* correspondent sont équilibrées (à l'exception de *B-Passive*) de la manière décrite dans la sous-section 3.2.1. Les méthodes dont le nom commence par *r* correspondent à celle qui utilise la réflection (expliqué dans 3.2.1). Cette méthode considérant une mémoire des prédictions passée, nous tentons de faire varié cette mémoire : avec $r*$ cette mémoire se souvient

de toute les itérations ; avec *r5* elle se souvient des 5 dernières ; avec *r1* elle ne se souvient que de la dernière. Les méthodes de combinaisons discutées sont indiquées par *MAB* (algorithme de Baram et coll. (2004)) et *Melange*. On tente de combiner *U-NB*, *U-NBTree* et *CoTrait*.

La table 3 montre la déficience de nos algorithmes pour les 14 premières itérations que nous avons testé sur les corpus de Pang et Lee (2008) et *ParlonsTV*.

On s'intéresse d'abord au techniques les plus basiques. Notre *discovery sampling* n'offre pas de très bon résultat sur le corpus *ParlonsTV*, mais fonctionne sensiblement mieux sur les critiques de cinéma. Nous pensons que cette méthode est intéressante pour les corpus où le langage est très varié, mais dans un corpus où le langage ne l'est pas (e.g. *ParlonsTV*) cette méthode est inefficace. Selon nos résultat l'incertitude des techniques probabilistes (*NB* et *NBTree*) permettent une assez bonne sélection. *U-NBTree* s'en sort mieux la où *U-NB* ne s'en sort pas. On remarque de bons résultats pour *U-NB* sur le corpus *ParlonsTV*. Lorsqu'on observe la figure 17, on remarque le lien logique entre la fiabilité de la prédiction et l'efficacité de l'*uncertainty sampling*. Pour ce qui des bons résultats de la prédiction des forêts aléatoires, ils sont discutables à cause du coté aléatoire de cette méthodes.

Passons au méthodes plus évoluées. Le comité qui offrent les meilleurs résultats est *CoLex*. Ce n'est pas étonnant, car ce comité a pour avantage de d'analyser les données sous différentes dimensions. Remarquons rapidement que les combinaisons offrent de mauvais résultats. Ces méthodes font un compromis entre plusieurs algorithmes, si certains algorithmes offrent de mauvais résultats, alors la combinaison en pâtira. Nos améliorations de l'*uncertainty sampling* offre de meilleurs résultats dans presque tous les cas (voir table 3). Même lors avec d'autres méthodes de prédictions, tels que les *committee-based sampling* les résultats sont meilleurs que les originaux. Une explication est que notre intuition est bonne (voir 3.2.1). La tentative de combiner l'équilibrage et la réflexion ne semble pas offrir d'amélioration très conséquente. Nous avons donc pas chercher à expérimenter plus ces méthodes.

Sélection des premiers exemples Dans 3.2.1, nous évoquons des solutions pour sélectionner les premiers exemple de l'ensemble d'apprentissage. Nous avons un peu expérimenté la sélection basée sur le *clustering* (en utilisant l'*uncertainty sampling* pour la sélection lors de l'*active learning*), et obtenons de mauvais résultats (le $F_1 score$ initial est 0.35 et il diminue lors des 4 premières itérations). La principale explication à cet échec est le manque d'équilibre entre les classes. Nous pensons que la méthode basée sur la classification non-supervisée des opinions offrira un meilleur ensemble initial.

4 Conclusions

Dans ce rapport, nous avons expérimenté des techniques génériques pour faire de la fouille d'opinions. Nous avons vu dans l'ordre, les problèmes de représentation des documents puis rapidement celui du choix du classificateur à utiliser et finalement nous avons utilisé l'*active learning* dans l'espoir d'augmenter la généricité de notre approche. Nous avons d'abord explorer le problème d'une sélection du lexique par apprentissage supervisé, et fait ressortir une simple formule qui permet choisir les mots discriminants d'après leur fréquences d'apparition dans les classes considéré (positif et négatif). Nous avons aussi tenté d'éliminer les mots vides (et en général les mots trop fréquents trop spécifiques au domaine) de manière statistique. En marge des mots traditionnels, nous avons tenté d'utiliser des émoticônes dans la classification. Certaines conclusions de Pang et coll. (2002), telles que l'efficacité de la représentation utilisant la présence des *uni-grammes* et le potentiel des Séparateurs à Vastes Marges (qui dans les expériences de selection du lexique offraient de résultats moins bon que *Naive Bayes* et qui dans les expériences

tentant de faire de la validation inter-domaine offre des résultats encourageant), ont été confirmées. *Naive Bayes* nous a donné de bons résultats dans de nombreux cas, sauf lorsque le domaine d'apprentissage des données est différent de celui des données de test. Aussi, pour ce problème, la technique des forêts aléatoires semble construire un modèles assez générique.

Grâce à l'*active learning*, nous sommes capables d'obtenir de meilleurs résultats en réduisant le nombre des exemples à annotées. De plus, nous mettons en relief deux manières d'améliorer les méthodes de sélection par incertitude (telles que *uncertainty sampling* et *committee-based sampling*). L'utilisation de ces méthodes permet donc la construction d'un classificateur pour la fouille d'opinions assez générique. Les perspectives à ce sujet sont de nombreuses améliorations liées à l'architecture du classificateur.

Perspectives Nous pensons qu'une architecture en cascade (Boiy et Moens (2009)) de plusieurs classificateurs, capables respectivement de détecter la subjectivité, d'utiliser les émoticônes pour améliorer la classification (Yasuhiro et coll. (2006)), et faisant finalement une agrégation entre plusieurs classificateurs (DZICZKOWSKI (2008)), pourrait offrir de bons résultats. Le domaine de l'*active learning* est aussi très prometteur, néanmoins pour sa mise en pratique le problème du critère d'arrêt (Zhu et coll. (2008)) est à explorer en priorité.

Références

(2009). *Défi Fouille de Texte : Analyse multilingue d'opinion*, Paris, France.

Aue, A. et Gamon, M. (2005). Customizing sentiment classifiers to new domains : a case study. Technical report, Microsoft Research.

Auer, P., Cesa-Bianchi, N., Freund, Y., et Schapire, R. E. (2002). The nonstochastic multiarmed bandit problem. *SIAM Journa on Computing*, 32(1) :48–77.

Baram, Y., El-yaniv, R., et Luz, K. (2004). Online choice of active learning algorithms. *Journal of Machine Learning Research*, 5.

Boiy, E. et Moens, M.-F. (2009). A machine learning approach to sentiment analysis in multilingual web texts. *Inf. Retr.*, 12(5) :526–558.

Dagan, I. et Engelson, S. P. (1995). Committee-based sampling for training probabilistic classifiers. Dans *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 150–157. Morgan Kaufmann.

Dave, K., Lawrence, S., et Pennock, D. M. (2003). Mining the peanut gallery : opinion extraction and semantic classification of product reviews. Dans *WWW*, pages 519–528.

Derks, D., Bos, A. E., et Von Grumbkow, J. (2007). Emoticons and social interaction on the internet : the importance of social context. *Computers in Human Behavior*, 23(1) :842–849.

DZICZKOWSKI, G. (2008). *Analyse de sentiments : système autonome d'exploration des opinons exprimées dans les critiques cinématographiques*. Thèse de doctorat, Ecole Nationale Supérieurres.

Esuli, A. et Sebastiani, F. (2006). Sentiwordnet : A publicly available lexical resource for opinion mining. Dans *Proceedings of the 5th Conference on Language Resources and Evaluation*, pages 417–422.

Fourour, N. (2002). Nemesis, un système de reconnaissance incrémentielle des entités nommées pour le français. *TALN*.

Hatzivassiloglou, V. et Wiebe, J. M. (2000). Effects of adjective orientation and gradability on sentence subjectivity. Dans *Proceedings of the 18th conference on Computational linguistics*, pages 299–305, Morristown, NJ, USA. Association for Computational Linguistics.

Hoi, S. C. H., Jin, R., et Lyu, M. R. (2009). Batch mode active learning with applications to text categorization and image retrieval. *IEEE Trans. on Knowl. and Data Eng.*, 21(9) :1233–1248.

Holmes, G., Donkin, A., et Witten, I. H. (1994). Weka : a machine learning workbench.

Hu, M. et Liu, B. (2004). Mining opinion features in customer reviews. Dans *AAAI'04 : Proceedings of the 19th national conference on Artifical intelligence*, pages 755–760. AAAI Press / The MIT Press.

Kaji, N. et Kitsuregawa, M. (2007). Building lexicon for sentiment analysis from massive collection of html documents. Dans *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Karlgren, J. et Cutting, D. (1994). Recognizing text genres with simple metrics using discriminant analysis. Dans *Proceedings of the 15th conference on Computational linguistics*, pages 1071–1075, Morristown, NJ, USA. Association for Computational Linguistics.

Liu, B. (2010). Sentiment analysis and subjectivity. *Handbook of Natural Language Processing, Second Edition*.

Maurel, S., Curtoni, P., et Dini, L. (2008). L'analyse des sentiments dans les forums. Dans *Atelier Fodop?*, pages 9–22.

Pang, B. et Lee, L. (2004). A sentimental education : sentiment analysis using subjectivity summarization based on minimum cuts. Dans *ACL 2004*, page 271, Morristown, NJ, USA. Association for Computational Linguistics.

Pang, B. et Lee, L. (2008). Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2) :1–135.

Pang, B., Lee, L., et Vaithyanathan, S. (2002). Thumbs up ? : sentiment classification using machine learning techniques. Dans *EMNLP 2002*, pages 79–86, Morristown, NJ, USA. Association for Computational Linguistics.

Polanyi, L. et Zaenen, A. (2006). Contextual valence shifters. *Computing Attitude and Affect in Text : Theory and Applications*, pages 1–10.

Raymond, C. et Fayolle, J. (2010). Reconnaissance robuste d'entités nommées sur de la parole transcrite automatiquement. Dans *Traitement Automatique des Langues Naturelles*, Montréal, Canada.

Roy, N. et McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. Dans *ICML '01 : Proceedings of the Eighteenth International Conference on Machine Learning*, pages 441–448, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Schmid, H. (1994). Treetagger - a language independent part-of-speech tagger. http ://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/.

Tanaka, Y., Takamura, H., et Okumura, M. (2005). Extraction and classification of facemarks. Dans *IUI '05 : Proceedings of the 10th international conference on Intelligent user interfaces*, pages 28–34, New York, NY, USA. ACM.

Tong, S. et Koller, D. (2002). Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2 :45–66.

Turney, P. D. (2002). Thumbs up or thumbs down ? : semantic orientation applied to unsupervised classification of reviews. Dans *ACL 2002*, pages 417–424, Morristown, NJ, USA. Association for Computational Linguistics.

Vernier, M., Monceaux, L., Daille, B., et Dubreil, E. (2009). Catégorisation sémantico-discursive des évaluations exprimées dans la blogosphère. Dans *TALN 2009*.

Wiebe, J. M. (1994). Tracking point of view in narrative. *Comput. Linguist.*, 20(2) :233–287.

Yasuhiro, S., Takamura, H., et Okumura, M. (2006). Application of semi-supervised learning to evaluative expression classification. 3878 :502–513.

Zhu, J., Wang, H., et Hovy, E. (2008). Multi-criteria-based strategy to stop active learning for data annotation. Dans *COLING '08 : Proceedings of the 22nd International Conference on Computational Linguistics*, pages 1129–1136, Morristown, NJ, USA. Association for Computational Linguistics.