TALN 2011, Montpellier, 27 juin – 1er juillet 2011

Classification en polarité de sentiments avec une représentation textuelle à base de sous-graphes d'arbres de dépendances

Alexander Pak, Patrick Paroubek
alexpak@limsi.fr, pap@limsi.fr
Université de Paris-Sud, Laboratoire LIMSI-CNRS, Bâtiment 508,
F-91405 Orsay Cedex, France

Résumé. Les approches classiques à base de n-grammes en analyse supervisée de sentiments ne peuvent pas correctement identifier les expressions complexes de sentiments à cause de la perte d'information induite par l'approche « sac de mots » utilisée pour représenter les textes. Dans notre approche, nous avons recours à des sous-graphes extraits des graphes de dépendances syntaxiques comme traits pour la classification de sentiments. Nous représentons un texte par un vecteur composé de ces sous-graphes syntaxiques et nous employons un classifieurs SVM état-de-l'art pour identifier la polarité d'un texte. Nos évaluations expérimentales sur des critiques de jeux vidéo montrent que notre approche à base de sous-graphes est meilleure que les approches standard à modèles « sac de mots » et n-grammes. Dans cet article nous avons travaillé sur le français, mais notre approche peut facilement être adaptée à d'autres langues.

Abstract. A standard approach for supervised sentiment analysis with n-grams features cannot correctly identify complex sentiment expressions due to the loss of information incurred when representing texts with bag-of-words models. In our research, we propose to use subgraphs from sentence dependency parse trees as features for sentiment classification. We represent a text by a feature vector made from extracted subgraphs and use a state of the art SVM classifier to identify the polarity of a text. Our experimental evaluations on video game reviews show that using our dependency subgraph features outperforms standard bag-of-words and n-gram models. In this paper, we worked with French, however our approach can be easily adapted to other languages.

Mots-clés : analyse de sentiments, analyse syntaxique, arbre de dépendances, SVM.

Keywords: sentiment analysis, parsing, dependency tree, SVM.

1 Introduction

L'approche « sac de mots » est un des premiers modèles de représentation textuelle, qui est de nos jour encore souvent utilisé pour l'analyse de sentiments. Le texte y est représenté comme un ensemble de n-grammes sans prise en considération de leur ordre d'apparition dans le texte, ni des relations qui les relient au sein du texte. Des approches classiques en apprentissage automatique (Naive Bayes or SVM) utilisent ensuite cette représentation pour construire des systèmes de classification en sentiments des textes. L'exactitude [1] de ce genre d'approche peut être très élevée, tout particulièrement lorsque l'on utilise des techniques avancées de sélection de traits, en conjonction avec des lexiques additionnels extraits de textes identifiés au préalable comme porteur d'opinion. Cependant, nous sommes convaincus que des modèles capables d'identifier des expressions plus complexes de sentiments, allant au delà de la simple reconnaissance de construction comme « bon film » ou « jeu déplorable », doivent permettre d'obtenir de meilleurs systèmes de classification. Un des problèmes de l'approche sac de mots réside dans la perte d'information lors de la construction de la représentation des textes, vus comme des collections de termes dissociés. Or les relations qu'entretiennent les mots au sein du texte sont souvent très importantes dans la détermination précise du degré ou de la polarité d'un sentiment. Si nous considérons la phrase : « Ce film est **mauvais** », elle exprime de manière évidente un sentiment négatif et un système de classification standard à base d'unigrammes n'aura pas de mal à la classer comme négative, pourvu qu'il ait été suffisamment entraîné sur des données appropriées. Dans le cas d'un énoncé un peu plus complexe comme : « Ce film n'est **pas mauvais** », un modèle unigramme simple sera probablement mis en échec, mais un modèle utilisant des bigrammes sera lui capable de détecter l'occurrence de « pas mauvais » comme un terme à connotation positive. Considérons maintenant un exemple encore plus complexe comme : « Ce film est **étonnamment pas si mal** » et là les systèmes à base d'unigrammes et de bigrammes vont probablement se tromper. Dans cet exemple, il faudrait qu'ils soient associés à un traitement plus sophistiqué de la négation.

En plus d'être incapables de prendre en compte toutes les expressions de négation, les modèles n-grammes sont incapables de représenter les dépendances longue distance. Un modèle de bigrammes pourra identifier « J'ai apprécié » comme un motif à connotation positive dans l'énoncé « **J'ai apprécié** le film », mais pas dans « **J'ai** beaucoup **apprécié** le film ». Nous pensons qu'il faut recourir à d'autres modèles que le modèle sac de mots si nous voulons progresser dans l'identification automatique des sentiments en utilisant une classification plus fine, qui rende par exemple compte de l'intensité d'un sentiment en plus de sa polarité, car les modèles sac de mots ne nous fournissent pas assez d'information.

Pour aller au delà des modèles sac de mots, nous proposons d'utiliser les arbres de dépendances issus de l'analyse syntaxique des phrases pour générer des sous-graphes, qui serviront à représenter un texte. Un arbre de dépendances est une représentation graphique associée à une phrase, dans laquelle les nœuds correspondent aux mots de la phrase et les arcs représentent des relations syntaxiques entre les nœuds comme : objet, sujet, modifieur etc. La Figure 1 représente un arbre de dépendance syntaxique pour la phrase « Je n'aime pas beaucoup le poisson ». Une telle représentation des phrases est parfaitement adaptée à l'analyse de sentiment voire même à la fouille d'opinion car :

– À partir de l'arbre de dépendances, nous pouvons facilement identifier le sous-graphe contenant la négation « ne $\xrightarrow{NEGAT}$ aime ».
– Nous pouvons identifier les marqueurs d'intensité : « beaucoup $\xrightarrow{VMOD\_POSIT1}$ aime »
– De même pour la source d'une expression d'opinion : « Je $\xrightarrow{SUBJ}$ aime » et la cible d'une expression opinion : « aime $\xrightarrow{OBJ}$ poisson »

Comme pour les modèles à base de n-grammes, notre approche utilise un paramètre de taille pour calibrer les sous-graphes extraits des arbres de dépendance pour représenter un texte. Nous posons que la taille d'un sous-graphe est égale au nombre de ses arcs. Ainsi, un sous graphe de taille 1 contiendra un arc et deux nœuds, un sous graphe de taille 2 contiendra 2 arcs et 3 nœuds, etc. La Figure 2 contient la représentation de l'énoncé « J'aime bien le poisson » au moyen de sous-graphes de taille 2.

Dans la section suivante, nous expliquons en détails comment obtenir la représentation à base de sous-graphes d'un texte à partir des arbres de dépendances syntaxiques qui lui sont associés. Ensuite, nous montrons comment utiliser cette représentation pour indexer des critiques de jeux vidéo et pour entraîner un classifieur en polarité de sentiments à base de SVM. Nous présentons notre protocole d'évaluation et les résultats obtenus par notre modèle

1. accuracy

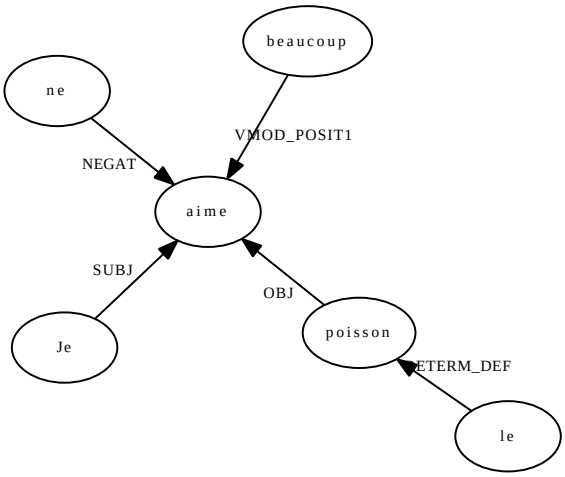

FIGURE 1 – Un arbre de dépendance syntaxique pour la phrase « Je n'aime pas beaucoup le poisson ». Les nœuds représentent des mots , les arcs des relations entre les mots. Le mot «pas» ne figure pas explicitement dans le diagramme, car il est encodé par la relation de négation.

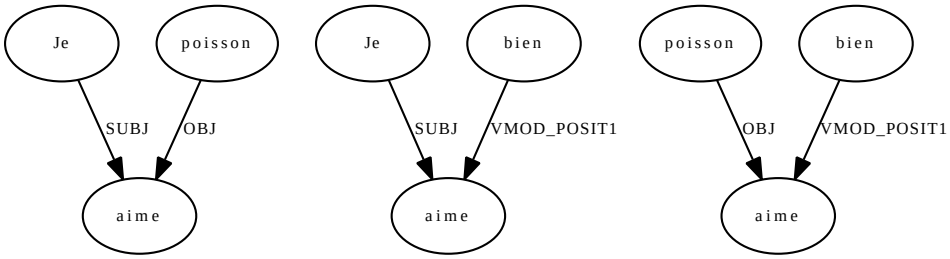

FIGURE 2 – Une représentation de la phrase « J'aime bien le poisson » avec des sous-graphes de taille 2. La relation « déterminant » contenant le nœud « le » a été écartée.

dans la Section 3, une présentation des recherches antérieures dans le domaine dans la Section 4 et la conclusion sur nos travaux en section 5.

2 Notre approche

2.1 Représentation à base de sous-graphes de dépendances

Nous utilisons la sortie en dépendances typées de l'analyseur syntaxique *Xerox Incremental Parser* (XIP) (Aït-Mokhtar *et al.*, 2002) pour construire l'arbre de dépendances de la phrase. La Table 1 contient un exemple de dépendances produites par XIP.

SUBJ(VERB :aime, PRON :Je)
OBJ(VERB :aime, NOUN :poisson)
VMOD_POSIT1(VERB :aime, ADV :beaucoup)
DETERM_DEF(NOUN :poisson, DET :le)
NEGAT(VERB :aime)

TABLE 1 – Les dépendances produites par XIP pour la phrase : « Je n'aime pas beaucoup le poisson »

Chaque ligne de la sortie de XIP contient une unique dépendance qui correspond à une description des relations grammaticales entre les mots de la phrase (de Marnee & Manning, 2008). Chaque dépendance peut être vue

comme un triplet <Type, Source, Cible>, où *Type* détermine la relation grammaticale (ex. sujet, objet, etc.) entre la *Source* et la *Cible*. La source et la cible sont représentés comme des mots associés à leur étiquette grammaticale. XIP produit aussi des relations unaires, que nous catégorisons en deux types distincts, avec pour chacun un traitement spécifique :

1. **Négations** (ex. `NEGAT(VERB:aime)`) Nous transformons une relation unaire en relation ternaire par l'ajout de la particule 'ne' comme cible. Ainsi, nous obtenons : `NEGAT(VERB:aime, NEG:ne)`

2. **Entités** XIP reconnaît et étiquette les entités telles que les noms de personnes, dates, temps, noms de lieux etc. Ces informations n'étant pas utiles pour la détection des sentiments, elles sont ignorées.

Nous écartons aussi la relation `SEQNP`, qui indique les énumérations dans les phrases ; ceci afin de réduire la taille de l'index, la suppression de cette relation n'ayant pas d'impact notoire sur nos résultats.

De l'ensemble de relations produit par XIP pour chaque énoncé, nous voulons obtenir un arbre dans lequel chaque nœud possède un sens complètement déterminé. Dans notre exemple, un nœuds comme « ne » n'a pas de sens intrinsèque et le nœud « aime » possède un sens partiel (il lui manque la prise en compte de la négation dans son interprétation). Par conséquent, nous avons besoin de fusionner certains nœud et de retirer certaines relations. Nous avons décidé de réduire le nombre de relations avec lesquelles travailler, car XIP produit plus de 90 types de relations (une liste complète est présentée en 5.).

2.1.1 Réduction du jeu de relations

Nous avons simplifié le jeu de relations de dépendances en ne considérant que les classes génériques en appliquant les règles d'assimilation de la Table 2.

NMOD_* -> NMOD	les modifieurs de nom (ante et post posés)
VMOD_* -> VMOD	les différents modifieurs de verbe
SUBJ_* -> SUBJ	les différents sujets
OBJ_* -> OBJ	les différents compléments d'objet directs
DEEPSUBJ* -> SUBJ	le sujet profond est assimilé au sujet de surface

TABLE 2 – Règles de simplification des relations de dépendances.

En outre, lors de la construction de l'arbre de dépendance, nous excluons certains arcs qui ne sont pas indispensables à notre analyse :
– Les déterminants, ainsi « le $\xrightarrow{DETERM\_DEF}$ film » devient « film », mais nous conservons les quantificateurs (`DETERM_NUM`, `DETERM_QUANT`, `DETERM_QUANT_DEF`, `DETERM_QUANT_DEM`).
– Les pronoms possessifs, ainsi « mon $\xrightarrow{DETERM\_POSS}$ livre » devient « livre ».
– Les relations modifieur de nom `NMOD`, lorsque la source et la cible sont tous deux des noms, ainsi « livre $\xrightarrow{NMOD\_POSIT1}$ cuisine » devient « livre ».
Au final, le jeu de relation de dépendances que nous considérons pour notre analyse est donné en Table 3.

ADJMOD	modifieur d'adjectif
ADVMOD	modifieur d'adverbe
DETERM_NUM	déterminant numérique
DETERM_QUANT	quantificateur
NMOD	modifieur de nom
OBJ	complément d'objet direct
SUBJ	sujet (de surface ou profond)
VMOD	modifieur de verbe

TABLE 3 – Jeu de relations de dépendances final.

2.1.2 Combinaison de nœuds

Nous utilisons les règles suivantes pour combiner les nœuds :
– nœuds liés par la relation de négation (NEGAT), ainsi l'arc « ne $\xrightarrow{NEGAT}$ aime » devient un nœuds simple « ne aime »
– verbes auxiliaires et principaux (AUXIL), ainsi l'arc « a $\xrightarrow{AUXIL}$ aimé » devient un nœuds simple « a aimé »
– verbes passifs, réfléchis et composés (AUXIL, AUXIL_PASSIVE, REFLEX, OBJ_SPRED, COORDITEMS_SC)

Un exemple d'arbre issu de l'application des règles précédentes est donné dans la figure Figure 3.

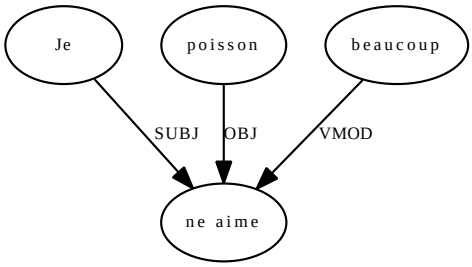

FIGURE 3 – Arbre de dépendance obtenu pour la phrase « Je n'aime pas beaucoup le poisson », après combinaison des nœuds et réduction des arcs.

Finalement, la phrase est représentée par un le jeu de tous les sous-graphes possibles pour une taille S, où S est égal au nombre d'arc des sous-graphes. Dans nos expériences, nous avons utilisé $S = 1, 2, 3$.

2.1.3 Nœud universel

La majorité des expressions de sentiment ont la même structure grammaticale. Par exemple, dans les expressions suivantes : « J'aime le poisson » et « J'aime le film » seul l'objet diffère tandis que le reste de la construction reste le même. Nous aimerions entraîner notre système à reconnaître ces expressions. Pour cela, nous avons ajouté un nœud universel, représentant la classe de tous les mots, dans les sous-graphes (Arora *et al.*, 2010).

Pour chaque sous-graphe obtenu à l'étape précédente, nous générons une permutation des sous-graphes contenants plusieurs nombres (de 0 à $S - 1$) de nœuds universels. Pour ce faire, nous remplaçons tout à tour chaque nœud d'un sous-graphe par un nœud universel, sauf pour les verbes, les adjectifs et les adverbes car ils peuvent exprimer des sentiments. Par ailleurs, nous interdisons d'avoir deux nœuds universels adjacents. Un exemple de l'emploi des nœuds universels avec la phrase « Je $\xrightarrow{SUBJ}$ aime $\xrightarrow{OBJ}$ poisson » est décrit dans la Figure 4.

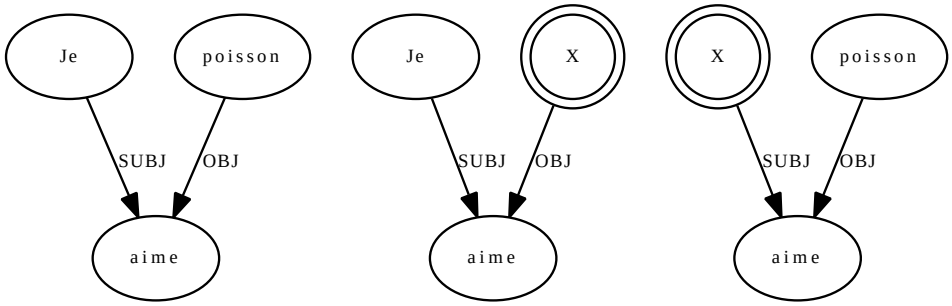

FIGURE 4 – Sous-graphes avec nœud universel (X) obtenus pour « J'aime le poisson »

2.2 Construction des vecteurs de traits

Nous représentons un texte donné T comme un vecteur de traits $T = \{w_1, w_2, \cdots, w_K\}$, où w_i est le poid d'un sous-graphe i dans le texte T et K est le nombre de sous-graphes dans T. Nous utilisons le schéma de pondération delta TFIDF lissé, car il a permis d'obtenir la meilleure performance dans des recherches antérieures (Paltoglou & Thelwall, 2010).

$$w_i = tf_i \cdot \Delta idf_i \tag{1}$$

$$\Delta idf_i = log \frac{N_1 \cdot df_2 + 0.5}{N_2 \cdot df_1 + 0.5} \tag{2}$$

où N_1 et N_2 représentent le nombre total de documents de classe 1 et 2, df_1 et df_2 sont des classes de fréquences du graphe i (c.a.d. le nombre de documents de classes 1 et 2 dans lesquelles le graphe apparaît). Dans notre cas, les classes 1 et 2 sont des documents positifs et négatifs.

3 Expériences et résultats

3.1 Les données

Nous utilisons des critiques de jeux vidéo issues du projet DOXA [2], dont le but est la construction d'une plateforme industrielle de fouille d'opinion.

Le corpus est constitué de critiques de jeux vidéo provenant de 8 sites dédiés [3]. Le corpus et ses annotations sont décrites dans (Paroubek *et al.*, 2010). Les annotations synthétisent les sentiments exprimés par les auteurs des critiques au niveau du document et du paragraphe (définis arbitrairement comme un empan de texte d'environ 100 mots). Un exemple d'annotation est fourni dans la table 4.

Attribut	Valeur
catégorie sémantique	une liste de 1 à 5 catégories d'opinion DOXA, ex. *« recommandation_suggestion »*
polarité	$-, \pm, +$, neutre
intensité	faible-moyen, fort
thème	la cible de l'expression d'opinion sélectionnée dans une taxonomie du domaine considéré (une liste de 1 à 5 concepts)
lien	lorsque plusieurs catégories sémantiques et plusieurs thèmes sont présents, le lien peut être fait entre certaines opinions s'ils sont plus particulièrement associés.
justification	référence au paragraphe/segment de texte qui représente au mieux l'opinion annotée

TABLE 4 – Annotation d'opinon DOXA au niveau document et paragraphe.

Dans les annotations DOXA, la polarité d'un sentiment est exprimée au moyen d'une échelle de six valeurs : *neutre, très-négatif, faible-moyen-négatif, mixte, faible-moyen-positif, fort-positif*. Nous avons sélectionné tous les documents ayant une polarité positive (*forst-positif* et *faible-moyen-positif*), ainsi que tous les documents avec une polarité négative (*fort-négatif* et *faible-moyen-négatif*) que nous avons répartis dans deux classes distinctes. Nous n'avons pas utilisé les documents annotés comme *neutre* (pas d'expression de sentiment) ni ceux annoté *mixte* (qui contiennent à la fois des expressions positives et négatives, résultant en une interprétation mitigée). Notre corpus contient donc 387 documents considérés à teneur positive et 250 à teneur négative. Nous avons ensuite divisé le sous corpus des documents positifs en deux parties : un corpus d'entraînement et un corpus de tests, en sélectionnant pour ce dernier, tous les documents qui ont été annotés par deux annotateurs. Le sous-corpus négatif a subit le même découpage. La Table 5 résume les caractéristiques de notre corpus.

2. https ://www.projet-doxa.fr/index.php

3. www.ecrans.fr, www.gamehope.com, www.gamepro.fr, www.jeuxactu.com, www.jeuxvideo.com, www.jeuxvideo.fr, www.play3-live.com

Classe	Entraînement	Tests
Positif	334	53
Negatif	197	35
Total	531	88

TABLE 5 – Nombre de documents par classe

3.2 Évaluation

Nous avons entraîné un classifieur SVM à base de n-grammes (Pang *et al.*, 2002) avec un schéma de pondération delta TFIDF (Paltoglou & Thelwall, 2010), que nous utilisons pour obtenir une mesure de performance de base. Les négations ont été traitées en attachant la particule de négation successivement au mot qui la précède et au mot qui la suit lors de la génération des n-grammes (Pak & Paroubek, 2010). Nous avons généré trois types de classieurs, respectivement à base de n-grammes, de bigrammes et de trigrammes.

De manières similaire, pour notre modèle à base de sous-graphes de dépendances, nous avons utilisé trois types de modèles, utilisant respectivement des sous-graphes de taille 1, 2 et 3.

Aussi bien pour le modèle à n-gramme que pour notre modèle à sous-graphes de dépendances, nous avons utilisé une implémentation libre de classifieur SVM issue de la librairie LIBLINEAR (Fan *et al.*, 2008), avec des valeurs de paramètre par défaut et un noyau linéaire. Le classifieur a d'abord été entraîné sur un jeu de 531 documents puis évalué sur un ensemble de 88 documents. L'exactitude moyenne et la précision moyenne (Manning & Schütze, 1999) ont été choisies comme mesures d'évaluation.

$$exactitude = \frac{vp + vn}{vp + vn + fp + fn} \tag{3}$$

$$precision = \frac{vp}{vp + fp} \tag{4}$$

où `vp` est le nombre de documents classés correctement comme positifs (*vrais positifs*), `vn` est le nombre de document classés correctement comme étant négatifs (*vrais négatifs*), `fp` est le nombre de document incorrectement identifiés comme positifs (*faux positifs*) et `fn` est le nombre de document incorrectement identifiés comme négatifs (*faux négatifs*).

3.3 Résultats

Les résultats de l'évaluation sont donnés dans la Table 6. Les mentions unigramme, bigramme et trigramme correspondent respectivement aux trois modèles de base n-gramme, tandis que les mentions subgraph-1, subgraph-2 et subgraph-3 correspondent à nos modèles à sous-graphes de dépendances, respectivement de taille 1, 2 et 3.

Modèle	Exactitude moy. (%)	Précision moy. (%)	Préc$_{pos}$ (%)	Préc$_{neg}$ (%)
unigramme	73.86	69.57	90.57	48.57
bigramme	72.73	69.11	86.79	51.43
trigramme	64.77	60.08	83.02	37.14
subgraph-1	**78.41**	**74.80**	**92.45**	**57.14**
subgraph-2	64.77	61.05	79.25	42.86
subgraph-3	60.23	59.22	64.15	54.29

TABLE 6 – Comparaison des mesures d'exactitude et de précision pour des modèles unigramme, bigramme et trigramme par rapport à nos modèles à sous-graphes de dépendances de tailles 1, 2 et 3.

Comme le montre la table de mesures, la meilleure valeur d'exactitude est obtenue avec un modèle à sous-graphes de dépendances de taille 1 (78.41%). Quant à elle, la meilleur valeur d'exactitude pour les modèles à n-grammes est obtenue avec un modèle unigramme (73.86%). Les performances des modèles n-gramme se dégradent au fur et à mesure que l'ordre du modèle augmente. Le même phénomène se produit avec les modèles à base de sous-graphes de dépendances : l'exactitude diminue avec l'accroissement de la taille des sous-graphes. D'après nous,

ce phénomène provient de la taille des données, qui n'est pas suffisante pour que les modèles d'ordres supérieurs soient confrontés à suffisamment d'exemples d'apprentissage.

Puisque dans nos données, nous avons plus de documents d'opinion positive, la précision moyenne de classification est meilleure pour ces derniers. La combinaison des modèles unigramme, bigramme and trigramme en vue d'obtenir de meilleurs résultats de classification n'a pas répondu à nos attentes de manière significative. De la même manière, la combinaison des modèles à base de sous-graphes de dépendances de différentes tailles n'a pas produit d'amélioration significative non plus.

Dans les Figure 5 et 6 nous présentons les 10 sous-graphes les plus fréquents de taille 1 et les 5 sous-graphes les plus frèquents de taille 2 (selectionnés avec le score Δidf) respectivement pour les classes de documents positifs et négatifs.

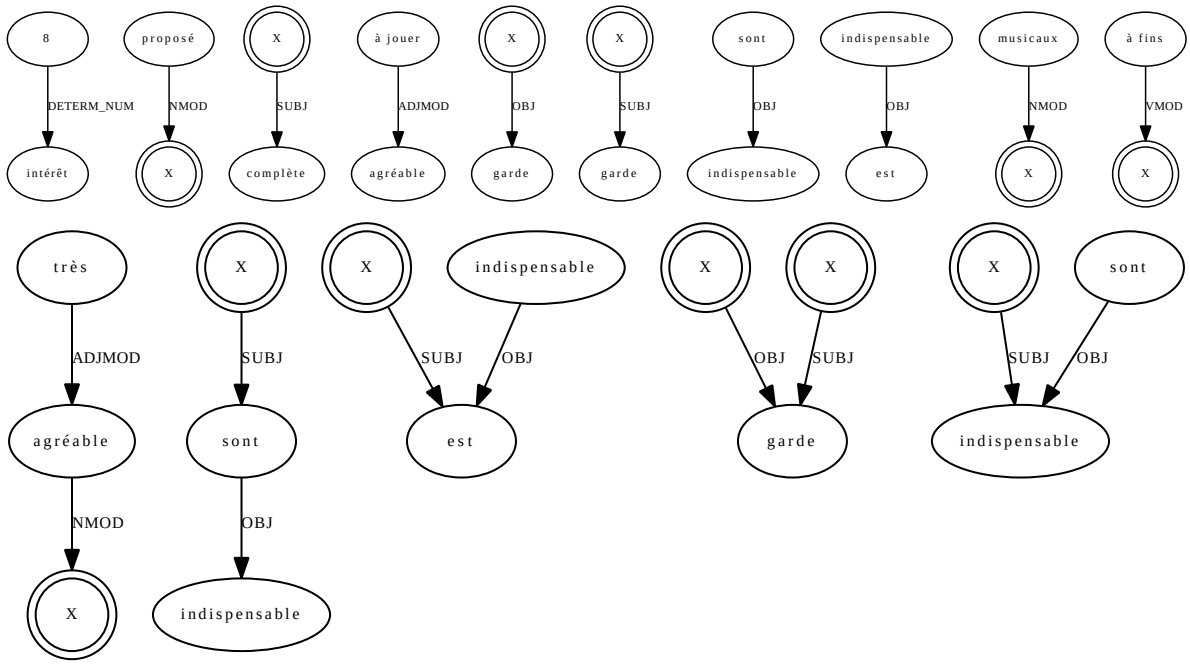

FIGURE 5 – Sous-graphes extraits des critiques de jeux vidéo positives

4 Travaux apparentés

Une première expérience par (Pang *et al.*, 2002), utilisant la représentation « sac de mots » avec des traits binaires et des classfieurs SVM, est devenue une base pour de nombreux travaux dans le domaine de la classification des sentiments. Les auteurs ont amélioré leur système dans (Pang & Lee, 2004) en utilisant un détecteur de subjectivité basé sur la notion de coupe minimale dans un graphe. L'utilisation d'un détecteur de subjectivité a permi de diminuer le bruit et se concentrer uniquement sur les phrases exprimant des sentiments. Cette méthode a amélioré la précision de 82.7% à 86.4%. Par la suite, de nombreux travaux ont utilisé des techniques avancées et des lexiques additionels pour augmenter l'espace de trait ou bien pour affiner la sélection des traits pertinents, améliorant ainsi la précision de la classification. (Whitelaw *et al.*, 2005) utilise des groupes d'appréciation, comme *« very good »* (très bon) ou « not terribly funny » (pas vraiment drôle) dans le cadre de la théorie de l'appréciation (*Appraisal theory*) en combinaison avec le modèle « sac de mots » et a obtenu une précision de 90.2% sur le jeu de données de critiques de films. (Aue & Gamon, 2005) a utilisé les SVM avec une sélection de traits par registre de probabilité et a obtenu une précision de 90.2% sur le même jeu de données.

L'arbre de dépendances des phrases a été largement utilisé dans le domaine de l'analyse de sentiments. Une recherche récente par (Arora *et al.*, 2010) a noté les problèmes de la représentation habituelle des textes par une approche « sac de mots ». Les auteurs suggéraient d'utiliser leur algorithme pour extraire les traits de sous-graphe

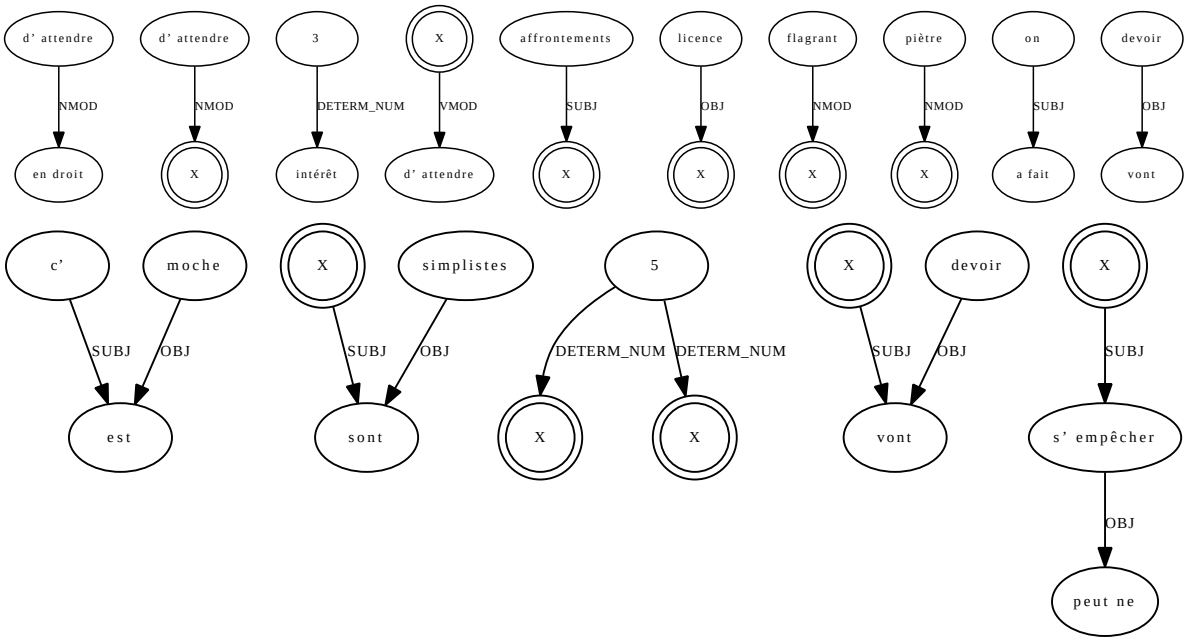

FIGURE 6 – Sous-graphes extraits des critiques de jeux vidéo négatives

par la programmation génétique. Cependant, les traits obtenues n'étaient pas utilisées pour remplacer le modèle n-gramme classique, mais plutôt comme un jeu de traits complémentaire. Un travail récent par (Nakagawa *et al.*, 2010) utilise un arbre de dépendances pour obtenir des traits qui sont utilisées pour entraîner un classifieur CRF pour la détection de la polarité des sentiments. Dans (Zhuang *et al.*, 2006), les auteurs utilisent des arbres de dépendances pour extraire les paires trait-opinion, ou le premier membre de la paire est un terme trait (ex. *« movie »*/film) et le second est un porteur d'opinion (ex. *« masterpiece »*/chef d'œuvre). Les arbres de dépendances sont utilisés afin d'établir les relations entre les mots traits et les mots-clés d'opinion. Dans (Chaumartin, 2007), l'arbre de dépendance est utilisé pour normaliser des titres vers des formes grammaticalement correctes, avant analyse des sentiments. Dans (Meena & Prabhakar, 2007), les auteurs utilisent l'arbre de dépendances et WordNet pour effectuer une analyse en sentiments.

5 Conclusion

Avec l'explosion du nombre de blogs et le développement des réseaux sociaux, la fouille d'opinion et l'analyse de sentiments sont devenus des domaines d'intérêt pour la recherche. Un travail pionnier sur la classification supervisée en sentiments à base de n-grammes ayant produit des résultats prometteurs, de nombreux chercheurs ont développé ce type de modèle. Cependant, l'approche « sac de mots » pour représenter un texte ne permet pas de prendre en compte des expressions complexes de sentiments et ne se prête que difficilement à l'utilisation de modèles sophistiqués de sentiments, qui nécessitent d'identifier entre autres, l'intensité d'une opinion ou la source/cible d'une expression d'opinion. Clairement, un nouveau type de modèle est nécessaire afin d'obtenir de meilleures performances en classification automatique de sentiments et en fouille d'opinion. Dans nos travaux, nous avons développé une nouvelle représentation à base de sous-graphes extraits des arbres de dépendances syntaxiques. Nous représentons un texte comme une collection de sous-graphes, où les nœuds sont des mots (ou des classes de mots) et les arcs des dépendances syntaxiques entre ceux-ci. Une telle représentation évite la perte d'information associée à l'emploi de modèles « sac de mots » pour représenter un texte, ces derniers étant basés uniquement sur des collections de n-grammes de mots. Nous avons testé notre modèle sur un ensemble de critiques de jeux vidéo, développé dans le cadre du projet DOXA sur la fouille d'opinion. Ainsi nous avons pu montrer qu'un classifieur SVM utilisant des traits construits à partir des sous-graphes extraits des arbres de dépendances, donne de meilleurs résultats que les systèmes traditionnels à base d'unigrammes. L'exactitude la plus élevée que nous ayons mesurée sur des textes en français est de 75%. Nous pensons que cette mesure peut

encore être améliorée par l'utilisation de techniques avancées de sélection de traits ou l'utilisation de lexiques dédiés à l'analyse de sentiments et d'opinion.

Remerciements

Ces travaux ont reçu le soutien financier du projet DOXA du pôle de compétitivité CAP-DIGITAL.

Références

AÏT-MOKHTAR S., CHANOD J.-P. & ROUX C. (2002). Robustness beyond shallowness : incremental deep parsing. *Nat. Lang. Eng.*, **8**, 121–144.

ARORA S., MAYFIELD E., PENSTEIN-ROSÉ C. & NYBERG E. (2010). Sentiment classification using automatically extracted subgraph features. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, p. 131–139, Morristown, NJ, USA : Association for Computational Linguistics.

AUE A. & GAMON M. (2005). Customizing Sentiment Classifiers to New Domains : a Case Study. In *Proc. International Conference on Recent Advances in NLP*.

CHAUMARTIN F.-R. (2007). Upar7 : a knowledge-based system for headline sentiment tagging. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, p. 422–425, Morristown, NJ, USA : Association for Computational Linguistics.

DE MARNEE M.-C. & MANNING C. D. (2008). Stanford typed dependencies manual. *http ://nlp.stanford.edu/software/dependencies_manual.pdf*.

FAN R.-E., CHANG K.-W., HSIEH C.-J., WANG X.-R. & LIN C.-J. (2008). Liblinear : A library for large linear classification. *J. Mach. Learn. Res.*, **9**, 1871–1874.

MANNING C. D. & SCHÜTZE H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA, USA : MIT Press.

MEENA A. & PRABHAKAR T. V. (2007). Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis. In *Proceedings of the 29th European conference on IR research*, ECIR'07, p. 573–580, Berlin, Heidelberg : Springer-Verlag.

NAKAGAWA T., INUI K. & KUROHASHI S. (2010). Dependency tree-based sentiment classification using crfs with hidden variables. In *Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, p. 786–794, Morristown, NJ, USA : Association for Computational Linguistics.

PAK A. & PAROUBEK P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC1́0)*, Valletta, Malta : European Language Resources Association (ELRA).

PALTOGLOU G. & THELWALL M. (2010). A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, p. 1386–1395, Morristown, NJ, USA : Association for Computational Linguistics.

PANG B. & LEE L. (2004). A sentimental education : Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, p. 271–278.

PANG B., LEE L. & VAITHYANATHAN S. (2002). Thumbs up ? : sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, p. 79–86, Morristown, NJ, USA : Association for Computational Linguistics.

PAROUBEK P., PAK A. & MOSTEFA D. (2010). Annotations for opinion mining evaluation in the industrial context of the doxa project. In N. C. C. CHAIR), K. CHOUKRI, B. MAEGAARD, J. MARIANI, J. ODIJK, S. PIPERIDIS, M. ROSNER & D. TAPIAS, Eds., *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta : European Language Resources Association (ELRA).

WHITELAW C., GARG N. & ARGAMON S. (2005). Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, p. 625–631, New York, NY, USA : ACM.

ZHUANG L., JING F. & ZHU X.-Y. (2006). Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, p. 43–50, New York, NY, USA : ACM.

Annexe A. Liste des dépendences produites par XIP

```
ADJMOD
    ADJMOD_POSIT1
    ADJMOD_PROPQUE

ADVMOD
AUXIL_PASSIVE

CONNECT
    CONNECT_REL
    CONNECT_SUBJ

COORD
    COORDITEMS
    COORDITEMS_SC

COREF_POSIT1_REL
    COREF_REL

DATE*
    DATE_PERIODE*

DEEPOBJ
DEEPSUBJ
    DEEPSUBJ_PASSIVE
    DEEPSUBJ_PROPQUE

DETERM
    DETERM_DEF
    DETERM_DEM
    DETERM_INT
    DETERM_NUM
    DETERM_POSS
    DETERM_QUANT
    DETERM_QUANT_DEF
    DETERM_QUANT_DEM

LIEU*
    LIEU_BATIMENT*
    LIEU_CONTINENT*
    LIEU_IMPERSO*
    LIEU_PAYS*
    LIEU_PAYS_REGION*
    LIEU_QUARTIER*
    LIEU_REGION*
    LIEU_REGION_VILLE*
    LIEU_VILLE*

NEGAT*
    NEGAT_SUBJ

NMOD
    NMOD_NUM
    NMOD_POSIT1
    NMOD_POSIT2
    NMOD_POSIT3
    NMOD_PROPQUE
    NMOD_REL

OBJ
    OBJ_COORD
    OBJ_COORD_SPRED
    OBJ_PROPQUE
    OBJ_PROPQUE_COORD
    OBJ_PROPQUE_SPRED
    OBJ_REL
    OBJ_SPRED
    OBJ_SUBJ

ORG*
    ORG_BATIMENT_LIEU*
    ORG_ENTREPRISE*

PERSONNE*
PREPMOD
PREPOBJ_REL
SEQNP**
SUBJ
    SUBJ_COORD
    SUBJ_IMPERSO
    SUBJ_IMPERSO_COORD
    SUBJ_IMPERSO_PASSIVE
    SUBJ_PASSIVE
    SUBJ_PASSIVE_COORD
    SUBJ_PASSIVE_PROPQUE
    SUBJ_PASSIVE_REL
    SUBJ_PROPQUE
    SUBJ_REFLEXIVE
    SUBJ_REL
    SUBJ_REL_COORD
    SUBJ_SUBJ

SUBJCLIT
    SUBJCLIT_PASSIVE

URL*
VMOD
    VMOD_COORD
    VMOD_COORD_SPRED
    VMOD_IMPERSO
    VMOD_POSIT1
    VMOD_POSIT1_SUBJ
    VMOD_POSIT1_SUBORD
    VMOD_POSIT2
    VMOD_PROPQUE
    VMOD_REL
    VMOD_SPRED
    VMOD_SUBJ
    VMOD_SUBORD
```

Les relations sont marquées d'un astérisque (*), une séquence de relation `SEQNP` est marquée par (**).