Automatic, adaptive, and applicative sentiment analysis

Alexander Pak

Graduate School of Computer Science
University of Paris-Sud

Automatic, Adaptive, and Applicative Sentiment Analysis

by Alexander Pak

A thesis submitted in fulfillment of the requirements for the degree of Philosophy Doctor in Computer Science

defended on June 13, 2012 before the committee:

President:	François Yvon	Université Paris-Sud
Reviewers:	Béatrice Daille	Université de Nantes
	Yves Lepage	University of Waseda
Examiners:	Suresh Manandhar	University of York
	Patrick Gallinari	Université Pierre et Marie Curie
Advisor:	Patrick Paroubek	Université Paris-Sud

ABSTRACT

Sentiment analysis is a challenging task today for computational linguistics. Because of the rise of the social Web, both the research and the industry are interested in automatic processing of opinions in text. In this work, we assume a multilingual and multidomain environment and aim at automatic and adaptive polarity classification. In particular, we propose a method for automatic construction of multilingual affective lexicons from microblogging to cover the lack of lexical resources. We propose a novel text representation model based on dependency parse trees to replace a traditional n-grams model. Finally, we investigate the impact of entity-specific features on classification of minor opinions and propose normalization schemes for improving polarity classification. The effectiveness of our approach has been proved in experimental evaluations that we have performed across multiple domains (movies, product reviews, news, blog posts) and multiple languages (English, French, Russian, Spanish, Chinese) including official participation in several international evaluation campaigns (SemEval'10, ROMIP'11, I2B2'11).

RESUMÉ DE THESE

L'analyse de sentiments est un des nouveaux défis apparus en traitement automatique des langues avec l'avènement des réseaux sociaux sur le WEB. Profitant de la quantité d'information maintenant disponible, la recherche et l'industrie se sont mises en quête de moyens pour analyser automatiquement les opinions exprimées dans les textes. Dans cet ouvrage, nous nous plaçons dans un contexte multilingue et multi-domaine pour explorer la classification automatique et adaptative de polarité. Plus particulièrement, nous proposons dans un premier temps de répondre au manque de ressources lexicales par une méthode de construction automatique de lexiques affectifs multilingues à partir de microblogs. Nous proposons ensuite, pour une meilleure analyse, de remplacer le traditionnel modèle n-gramme par une représentation à base d'arbres de dépendances syntaxiques. Finalement, nous étudions l'impact que les traits spécifiques aux entités nommées ont sur la classification des opinions minoritaires et proposons une méthode de normalisation des décomptes d'observables, qui améliore la classification de ce type d'opinion. Nos propositions ont été évaluées quantitativement pour différents domaines d'applications (les films, les revues de produits commerciaux, les nouvelles et les blogs) et pour plusieurs langues (anglais, français, russe, espagnol et chinois), avec en particulier une participation officielle à plusieurs campagnes d'évaluation internationales.

État de l'art

La première partie de la thèse présente l'état de l'art en fouille d'opinion et analyse de sentiments. L'analyse de sentiments est un domaine récent en traitement automatique des langues, qui confronte les chercheurs à toute la complexité de la langue naturelle. Si le but recherché en traitement automatique des langues est d'être capable de traiter avec des ordinateurs le langage humain, l'objectif en analyse de sentiments et de pourvoir reconnaître les émotions humaines, telles qu'elles sont exprimées dans les textes. Dans le chapitre 1 nous exposons le thème de notre recherche ainsi que le plan de notre thèse. Dans le chapitre 2, nous définissons les concepts clés à la base de nos travaux. Les termes *analyse de sentiments* (sentiment analysis) et *fouille d'opinion* (opinion mining) sont souvent utilisés de manière interchangeable. D'après Pang and Lee (2008), le vocable analyse de

sentiments est préféré dans le domaine du traitement automatique des langues, tandis que le terme fouille d'opinion a été lui adopté par la communauté de la recherche d'information. Bien que ces deux termes concernent des champs d'investigation très proches, qui pourraient même être considérés comme une seule et même entité, nous avons choisi pour nos travaux d'utiliser le terme analyse de sentiments, que nous distinguons de la fouille d'opinion.

Nous postulons que l'**opinion** est l'expression d'un individu à propos d'un objet ou d'un sujet particulier. Nous qualifions la personne qui s'exprime comme le **porteur d'opinion** (opinion holder) et le sujet de l'expression comme la **cible de l'opinion** (opinion target). Ainsi le terme **fouille d'opinion** se réfère au champ du traitement automatique des langues qui étudie les opinions. Nous distinguons les opinions des **faits**, qui sont des informations avérées, comme le sont en particulier les informations que l'on désigne par le terme sens commun. Notre définition de l'opinion, nécessite que lui soient associés un porteur et une déclaration de ce dernier, précisant sa position par rapport à la cible, sinon ce n'est pas une opinion. Par exemple, l'énoncé « j'ai froid » ("I am cold"), n'est pas, d'après nous, une expression d'opinion, car il n'y a pas à proprement parler de positionnement exprimé par rapport à un objet ou un sujet, mais plutôt l'expression d'un fait. Par contre, l'énoncé « j'ai l'impression qu'il fait froid dans cette pièce » est une expression d'opinion, avec comme cible la température de la pièce. De la même manière, l'énoncé « l'économie est en récession » ("Economy is in recession") n'est pas une expression d'opinion, car le porteur n'est pas mentionné explicitement, mais l'énoncé « Le ministre croit que l'économie est en récession » ("The minister believes that the economy is in recession") est une expression d'opinion, avec « le ministre » comme porteur de l'opinion.

Nous définissons le **sentiment** (sentiment) comme le jugement que porte un individu sur un objet ou un sujet, ce jugement étant caractérisé par une **polarité** (polarity) et une **intensité** (intensity). L'**analyse de sentiments** (sentiment analysis) est le champs du traitement automatique des langues qui étudie les sentiments. Pour nous, une polarité est soit positive, soit négative, soit un mélange de ces deux valeurs, tandis que l'intensité montre le degré de positivité ou de négativité, et varie de faible à forte. De notre définition, il ressort qu'un **sentiment est un type particulier d'opinion dotée d'une polarité**. Ainsi nous opposons les sentiments aux faits et aux expressions de neutralité face à un objet ou un sujet particulier.

Par **action bénéfique** (beneficial action), nous entendons une action qui profite au possesseur de la cible de l'opinion. Par exemple, dans le cas des critiques de film, la cible de l'opinion sera un film particulier, le possesseur de la cible sera une compagnie cinématographique et l'action bénéfique sera l'achat d'un billet pour une séance ou l'achat d'un DVD. En politique, la cible de l'opinion pourra être un candidat à une élection et l'action bénéfique, un vote pour ce

candidat[1]. Ainsi, **la polarité d'un sentiment sera dite positive si l'opinion est en faveur de l'action bénéfique et elle sera dite négative si elle s'y oppose**. L'intensité d'un sentiment mesure dans ce cas le degré de soutien ou d'opposition à l'action bénéfique. Notons, que le soutien ou l'opposition n'ont pas besoin d'être explicites. Par exemple, écrire une bonne critique pour un film, n'implique pas nécessairement d'inciter explicitement le lecteur à aller voir le film ou à acheter le DVD; le contenu positif de la critique étant une motivation suffisante en soi, qui va susciter, par voie de conséquence, l'achat du film.

1. Dans ce cas, la cible est aussi le possesseur.

Dans le chapitre 3, nous passons en revue les tâches communément effectuées en fouille d'opinion et analyse de sentiments : analyser la subjectivité, détecter les opinions, classer selon la polarité, identifier le porteur et la cible des expressions d'opinion, résumer les opinions, détecter l'ironie, détecter les « fausses » opinions (spams).

Le chapitre 4 est consacré au thème central de nos travaux de thèse : **le classement en polarité** (polarity classification); nous y présentons en détails la problématique scientifique qui concerne essentiellement l'analyse du discours, le traitement des négations, le traitement des métaphores, l'adaptation au domaine et le multilinguisme. Dans ce chapitre, nous faisons aussi un tour d'horizon des données expérimentales et des cadres évaluatifs qui existent pour les algorithmes de classement polaire.

Le chapitre 5 présente les approches existantes pour le classement en polarité, en distinguant les deux grand courants qui sont d'une part les méthodes à base de lexique et d'autre part les méthodes statistiques. Les premières utilisent un lexique affectif pour déterminer la polarité d'un texte, tandis que les secondes mettent en œuvre l'apprentissage automatique sur des textes de polarité connue pour construire des modèles de reconnaissance de cette polarité. Les méthodes à base de lexique sont coûteuses car elles nécessitent un gros travail de la part d'experts pour construire le lexique. A l'opposé les méthodes statistiques sont beaucoup plus faciles à mettre en œuvre mais donnent en général de moins bons résultats, avec la réserve que la qualité des performances augmente avec l'augmentation de la taille des données d'apprentissage.

Nos travaux

La seconde partie de la thèse est dédiée à la présentation de notre contribution à l'analyse de sentiments. Nous avons concentré nos efforts sur les deux dernières problématiques scientifiques présentées dans le chapitre 4, à savoir : l'adaptation au domaine et le multilinguisme. Nous ne voulons pas dépendre d'une ressource spécifique à un domaine particulier, comme par exemple une ontologie « métier ». Nous voulons aussi être, autant que faire se peut, indépendant de la langue, en proposant des algorithmes facilement transportable vers de nou-

velles langues cibles. C'est pourquoi nous trouvons au cœur de nos travaux, un classifieur à base d'apprentissage automatique, qui n'a besoin que de données d'apprentissage dans la langue cible. Nous faisons en effet l'hypothèse, qu'il est beaucoup plus facile de collecter du matériaux d'apprentissage dans différentes langues, plutôt que de porter des ressources lexicales vers une nouvelle langue cible.

Les Microblogs

Dans le chapitre 6, nous montrons quel potentiel constituent les microblogs pour l'analyse de sentiments. Une mode récemment apparue sur Internet a engendré une explosion du nombre de sites permettant de diffuser des microblogs, ces petits messages dont la taille maximale est restreinte à un texte très court. A tel point que c'est devenu en quelques années un des principaux type de communication sur Internet. La quantité très importante d'information présente sur les sites de microblogs les rend attractifs en tant que source de données pour la fouille d'opinion et l'analyse de sentiments. Nous utilisons Twitter, la plus grande plateforme de microblogs à ce jour, comme une source de données multilingues pour l'analyse de sentiments. Dans le chapitre 6, nous montrons comment nous avons obtenu un jeu de données étiqueté avec des annotations décrivant les sentiments exprimés dans les blogs, de manière automatique, en utilisant les émoticônes[2] comme des annotations bruitées. Nous relatons ensuite comment nous avons utilisé ces données dans 3 types de tâches :

2. Ces esquisses de visages représentant une émotion ou un états d'esprit particulier au moyen de quelques caractères qui sont fréquemment utilisés dans les communications sur Internet.

1. construction d'un lexique affectif pour différentes langues,

2. classement en polarité de critiques de jeux vidéo en français,

3. désambiguïsation d'adjectifs ambigus exprimant des sentiments en chinois.

Notre intention était de ne pas utiliser d'outil linguistique sophistiqué, afin de préserver à notre approche son caractère indépendant de la langue cible, ou du moins facilement transposable à une autre langue. Nous avons évalué notre approche en effectuant des expériences par comparaison des résultats avec ceux obtenus en utilisant un corpus annoté manuellement, ou un lexique construit par des experts ainsi qu'en participant à la campagne d'évaluation internationale SemEval 2010[3]. Lors de ces évaluations notre système a obtenu des performances comparables à celles d'un classifieur supervisé, qui lui nécessite de disposer de données d'apprentissage annotées. Notre méthode est entièrement automatique et n'a besoin d'aucune autre ressource langagière construite à la main.

3. Cette participation est présentée en détails au chapitre 9).

Les d-grammes

Les modèles n-grammes sont un moyen traditionnel de représentation des textes, souvent utilisé en analyse de sentiments. Cependant, nous pensons que la difficulté intrinsèque de la tâche appelle à l'utilisation de nouveaux modèles mieux adaptés à la capture des opinions. C'est pourquoi nous proposons dans le chapitre 7 un nouveau modèle s'inspirant des n-grammes, mais construits à partir des triplets constitutifs des dépendances syntaxiques. Nous avons appelé ce nouveau modèle : d-gramme. De nos expériences, il ressort que l'approche à base de d-grammes contient plus d'information pertinente pour l'analyse de sentiments que les simples modèles à sac-de-mots. Prenons comme exemple l'énoncé « La bande son était affreuse » :

$$S = \text{“The soundtrack was awful”}$$

Un graphe de dépendances syntaxiques possible pour cet énoncé est présenté dans la figure 7.2.

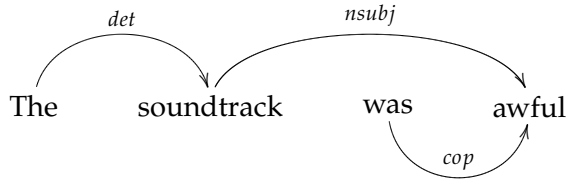


Figure 1: *Le graphe de dépendances syntaxiques produit par l'analyseur syntaxique de Stanford pour l'énoncé “The soundtrack was awful”*

À partir des dépendances de ce graphe nous construisons les d-grammes suivants :

$$\begin{aligned}\text{dgrams}(S) = \{&(\text{The}, \text{det}, \text{soundtrack}),\\ &(\text{soundtrack}, \text{nsubj}, \text{awful}),\\ &(\text{was}, \text{cop}, \text{awful})\}\end{aligned}$$

À fin de comparaison, voici les représentations à base respectivement d'unigrammes et de bigrammes pour le même énoncé :

$$\begin{aligned}\text{unigrams}(S) = \{&(\text{The}),\\ &(\text{soundtrack}),\\ &(\text{was}),\\ &(\text{awful})\}\end{aligned}$$

$$\begin{aligned}\text{bigrams}(S) = \{&(\text{The}, \text{soundtrack}),\\ &(\text{soundtrack}, \text{was}),\\ &(\text{was}, \text{awful})\}\end{aligned}$$

Les modèles d-grammes présentent l'avantage d'être capable de trouver les dépendances à longue distance et d'apporter une information plus pertinente pour le rattachement syntaxique des mots entre

eux. Fait important pour la fouille d'opinion, les modèles d-grammes facilitent aussi le repérage des négations.

Pour nos évaluations nous avons utilisé le jeu de données Cross-Lingual Sentiment[4] et construit par Prettenhofer and Stein (2010). Ce jeu de données est composé de critiques de produits commerciaux publiées sur Amazon dans 4 langues. Nous avons utilisé les critiques en anglais et en français. Elles sont réparties en 3 domaines selon le type de produit : livres, musique et DVDs. Pour chaque domaine nous avons 2.000 critiques positives et 2.000 négatives, pour un total de 24.000 documents utilisés dans cette expérience. Pour évaluer notre modèle nous avons utilisé une implémentation de machine à vecteurs supports linéaire de la bibliothèque LIBLINEAR Fan et al. (2008) et une implémentation personnelle d'un classifieur Bayésien naïf . Nous avons effectué un validation croisée à 10 replis pour estimer l'exactitude moyenne (average accuracy). Les résultats de l'expérience montrent l'efficacité de l'approche d-gramme par rapport aux modèles traditionnels à base d'unigrammes ou de bigrammes. Nous en concluons donc que notre méthode est générale et indépendante du domaine d'application ou de la langue cible.

4. Cross-Lingual Sentiment disponible à l'url : http://www.uni-weimar.de/cms/medien/webis/research/corpora/webis-cls-10.html

Améliorer les stratégies de pondération

Le chapitre 8 présente le problème de l'écrasement des statistiques des opinions minoritaires par celles des opinions majoritaires dans les approches classiques et la solution que nous proposons pour résoudre ce problème. De nos jours, il est très facile de rassembler de très grosses quantités de textes contenant des opinions à partir d'Internet, en allant les chercher sur les réseaux sociaux, les sites de critiques de produits commerciaux, les forums etc. Il est possible à partir de ces corpus de construire assez facilement un système de classement en polarité, qui soit capable de classer des documents de même nature que ceux du corpus original, avec un niveau acceptable d'exactitude. Cependant le système ainsi obtenu comporte un biais en faveur des opinions majoritairement exprimées dans le corpus d'apprentissage. Si maintenant, nous utilisions ce système pour déterminer la polarité de nouvelles critiques d'un produit pour lequel les critiques positives sont majoritaires dans le corpus d'apprentissage, il est fortement probable que toutes les nouvelles critiques ainsi analysées seraient aussi considérées comme positives, car le texte de ces critiques contiendra certainement les mêmes traits considérés comme indicateurs de positivité (par exemple le nom du produit, la marque ou la référence au modèle du produit) que les critiques du même produit dans le corpus d'apprentissage. Et ce, quel que soit le domaine applicatif considéré, par exemple pour les films, ces traits indicateurs de positivité seraient le titre, le nom des acteurs principaux, le nom du réalisateur, du producteur etc. Si nous appelons la cible de l'opinion l'*entité* (entity), ces traits peuvent être considérés comme étant *spécifiques à l'entité* (entity-specific). Para-

doxalement, ce biais est un facteur d'amélioration de l'exactitude du système de classement en polarité, car la distribution entre les critiques positives et négatives d'un produit est en général la même entre le corpus d'apprentissage et le corpus de test sur lequel on applique l'algorithme de classement. Si un produit est bon, il recevra aussi plus de critiques positives dans ce nouveau corpus et réciproquement.

Cependant, nous pouvons être amené à souhaiter disposer d'un système de classement en polarité, qui non seulement possède globalement (en moyenne) une bonne exactitude, mais qui soit aussi capable de déterminer correctement la polarité d'une critique minoritaire, c'est à dire des critique, qui malgrè les louanges de la majorité notent très négativement un produit ou au contraire mentionnent des points positifs pour un produit considéré comme mauvais. Le fonctionnement d'un tel système, s'approche plus de celui d'un expert, qui prend une décision objective à partir des informations mentionnées dans la le texte de la critique, sans se laisser influencer a priori par les opinions que la majorité entretient pour ce produit.

Pour prouver nos dires, nous avons utilisé deux jeux de données standards: des critiques de film et de produits commerciaux qui ont été utilisé par le passé pour des recherche en analyse de sentiment Maas et al. (2011), Blitzer et al. (2007), Duh et al. (2011). Pour chacun des deux jeux de donnée, nous avons construit une version *biaisée* (biased), en regroupant d'abord les critiques en fonction de leur entité cible (un film ou un produit particulier) et ensuite en sélectionnant des groupes avec une distribution déséquilibrée entre les critiques positives et négatives. Nous montrons que les approches traditionnelles, c'est-à-dire utilisant des machines à vecteurs supports et des n-grammes de traits, sont beaucoup moins performantes pour classer les critiques minoritaires que pour classer les critiques majoritaires dans notre jeu de données biaisé. Pour améliorer le classement des critiques minoritaires, nous devons réduire l'importance des termes qui pourraient introduire un biais dans le classement, ce que nous proposons de faire au moyen de deux mesures : la *fréquence moyenne d'un terme* et la *proportion d'entité*.

Fréquence moyenne d'un terme

La fréquence moyenne d'un terme (agv.tf) est le nombre moyen de fois qu'un terme apparaît dans un document :

$$\text{avg.tf}(g_i) = \frac{\sum_{\{d|g_i \in d\}} \text{tf}(\text{g}_\text{i})}{\|\{d|g_i \in d\}\|} \tag{1}$$

où $\{d|g_i \in d\}$ est l'ensemble des documents qui contient g_i

La normalisation à base de fréquence moyenne d'un terme est basée sur l'observation que les auteurs de critiques ont tendance à utiliser un vocabulaire riche quand ils expriment leur attitude par

rapport à un film ou un produit. Ainsi, les termes expriment des sentiments comme *remarquable* (outstanding) ou *adorable* (lovingly) ont une fréquence moyenne proche ou égale à 1, tandis que les termes non subjectifs ont une fréquence moyenne plus élevée. En particulier, cela est vrai pour les termes spécifiques à l'entité comme les titres de film, les noms d'acteurs, les marques et les noms de modèles qui sont souvent mentionnés plusieurs fois au sein d'un même document. Afin de normaliser le vecteur représentatif d'un document qui associe à chaque terme présent dans le document un poids représentatif de son importance, nous divisons chaque poids par la fréquence moyenne du terme correspondante (avg.tf) :

$$w(g_i)^* = \frac{w(g_i)}{\text{avg.tf}(g_i)} \tag{2}$$

Proportion d'entité

La proportion d'entité (ep) est la proportion des occurrences d'un terme par rapport aux différentes entités comparativement à la fréquence des documents :

$$\text{ep}(g_i) = \log\left(\frac{\|\{e|g_i \in e\}\|}{\|\{d|g_i \in d\}\|} \cdot \frac{\|D\|}{\|E\|}\right) \tag{3}$$

où $\{e|g_i \in e\}$ est l'ensemble des entités qui contient g_i dans leurs critiques, $\|D\|$ est le nombre total de documents, $\|E\|$ est le nombre total d'entités.

La normalisation de proportion d'entité favorise les termes qui apparaissent dans les critiques de nombreuses entités mais dans peu de documents. Nous distinguons trois types de termes :

1. Les termes spécifiques à une entité, tels que les noms de produits ou les titres de film, qui sont associés à peu d'entités et donc devraient apparaître dans peu de documents. La valeur de ep devrait être proche de celle de la constante de normalisation $\frac{\|D\|}{\|E\|}$ (nombre moyen de documents par produit).

2. Les termes subjectifs, tels que « remarquable » (“outstanding”) ou « adorable » (“lovingly”), qui devraient apparaître associés à beaucoup de produits et dans un nombre relativement restreint de documents, car les auteurs utilisent un vocabulaire varié. La valeur de ep sera plus grande que la constante de normalisation.

3. Les mots-outils, tels que les déterminants et les prépositions, devraient apparaître dans presque tous les documents, et donc associés à presque tous les produits. La valeur de ep sera proche de celle de la constante de normalisation.

Pour normaliser le vecteur représentatif d'un document, nous multiplions chaque poids associé à un terme par la proportion d'entité associée à l'objet de la critique.

$$w(g_i)^* = w(g_i) \cdot \text{ep}(g_i) \tag{4}$$

Toutes nos expériences réalisées avec des versions spécialement préparées de jeux de données standards ont montré une amélioration des performances de l'exactitude de classification pour les critiques minoritaires. Cependant nous avons quand même observé une légère baisse dans la mesure d'exactitude globale, car il est toujours plus bénéfique pour un classement en polarité de suivre la tendance majoritaire d'opinion.

Au final, c'est toujours le développeur d'un système de classement en polarité qui devra choisir entre un système biaisé dont la performance globale sera légèrement meilleure et un système capable d'identifier correctement les textes d'opinion minoritaire en fonction des visées applicatives qui président à la création de son système. Des applications possibles de notre procédure de normalisation sont l'analyse des retours clients, afin de détecter très tôt les critiques pour supprimer la source du mécontentement ou la détection des signaux faibles (rumeurs), en particulier pour les application sécuritaires, des types d'application qui ont besoin d'une classification à grain fin des documents.

Applications

Un des aspects important de nos travaux concerne leur application directe à des problèmes concrets. Dans la partie 3 de notre thèse, nous relatons notre participation à différente campagnes d'évaluation, qui nous a permis de tester notre approche. Plus particulièrement, nous avons participé aux campagnes internationales suivantes :

- SemEval'10 : désambiguïsation d'adjectifs ambigus exprimant des sentiments en chinois (chapitre 9)
- ROMIP'11 : classement en polarité de critiques de produits commerciaux en russe (chapitre 10)
- I2B2'11 : détection d'émotions dans des notes de suicide (chapitre 11)

SemEval'10

Le jeu de données de la campagne SemEval 2010 Wu et al. (2010) est constitué de courts textes en chinois contenant des adjectifs pris dans une liste fermée, et dont le sentiment associé doit être désambiguïsé en fonction du contexte. Dans notre approche, nous utilisons la plate-forme de microblogs Twitter pour collecter des messages à teneur émotionnelle et construire deux sous-corpus contenant respectivement : les messages à teneur positive et ceux à teneur négative, comme nous l'avons décrit dans le chapitre 6. Notre système de classement en sentiments construit avec ces données, utilise une approche bayésienne naïve multinomiale. Puisque les textes à analyser étaient courts, nous fait l'hypothèse que la polarité à associer à l'adjectif était la même polarité que celle du document entier.

Dans nos expériences, nous avons utilisé deux jeux de données : un jeu d'essai contenant 100 phrase en chinois et un jeu de test de 2.917 phrases, tous deux fournis par les organisateurs de la campagne d'évaluation. Les mesures d'évaluation utilisées pour cette campagne sont la micro et la macro exactitude. Il faut noter que notre approche peut être appliquée sans changement à n'importe quelle autre langue à condition de disposer de suffisamment de données d'apprentissage issues de Twitter. Nous avons obtenu respectivement 64% de macro et 61% de micro exactitude, à la tâche de SemEval 2010, ce qui est une performance inférieure à celle de la plupart des autres participants (nous sommes 6$^{\text{èmes}}$ sur 7 participants), mais notre système est entièrement automatique et n'a recourt à aucun lexique construite manuellement.

ROMIP'11

ROMIP est une campagne internationale d'évaluation annuelle en recherche d'information qui a débuté en 2002 Dobrov et al. (2004). Pour la campagne de 2011, les organisateurs ont ajouté une piste sur l'analyse de sentiments dont le but était le classement en opinion de textes écrits par des consommateurs. Un jeu de données composé de critiques de produits commerciaux issues du services en ligne de recommandation Imhonet et de l'aggrégateur de produits Yandex.Market a été fourni aux participant pour entraîner leurs systèmes. Le jeu de données contenait des critiques pour trois types de produits : les appareils photo numériques, les livres et les films.

L'analyse de sentiment est une tâche difficile, même pour les langues pour lesquelles les ressources linguistiques sont nombreuses comme c'est le cas pour l'anglais. En plus des traitements relativement basiques comme l'étiquetage morpho-syntaxique, des outils d'analyse du langage plus sophistiqués comme des analyseurs de discours ou des lexiques spécifiques sont nécessaires à certaines approches actuelles. C'est pourquoi il est très difficile d'adapter des méthodes initialement développées pour d'autres langues au russe, en particulier celles développées pour l'anglais. L'un des rares système de classement en sentiment développé pour la langue russe Pazelskaya and Solovyev (2011) est un système à base de règles, qui utilise un lexique affectif construit manuellement ainsi qu'un étiquetage morpho-syntaxique et des informations syntaxiques au niveau lexical. Cependant, à notre connaissance, il n'existe pas de ressource publique pour l'analyse de sentiments en russe et développer une approche à base de lexique serait bien trop coûteuse car il faudrait partir de rien.

Pour résoudre ce problème, nous avons décidé d'employer une approche indépendante de la langue qui n'ait pas besoin d'analyse du traitement du langage sophistiquée ni de lexique dédié, qui rappelons le, n'existe pas pour le russe. C'est pourquoi, nous avons employé un système à base de machine à vecteurs supports avec des traits con-

struits sur des n-grammes, des étiquettes morpho-syntaxiques et une analyse syntaxique en dépendances. Nous avons du entraîner un analyseur syntaxique en dépendances sur le corpus national russe (Russian National Corpus) [5]. De plus nous avons effectué une étude sur la pondération des termes et la composition du corpus pour optimiser les performances de notre système qui a été classé d'après les mesures de performance officielles, quatrième dans la piste de classification binaire pour le domaine des appareils photo numériques, troisième dans la piste à trois classes pour le domaine des films et premier dans la piste à cinq classes tous domaines confondus,

5. **Russian National Corpus**: http://www.ruscorpora.ru/en/

I2B2'11

La seconde piste de la campagne d'évaluation I2B2 2011 avait pour but la reconnaissance des opinions exprimées dans un corpus de notes de suicide, en étiquetant les phrases à l'aide d'une ou plusieurs des quinze catégories suivantes : *instructions* (instructions), *information*(information), *désespoir* (hopelessness), *culpabilité* (guilt), *reproche* (blame), *colère* (anger), *chagrin* (sorrow), *peur* (fear), *maltraitance* (abuse), *amour* (love), *reconnaissance* (thankfulness), *espoir* (hopefulness), *bonheur-tranquilité* (happiness-peacefulness), *fierté* (pride), *pardon* (forgiveness).

Nous avons contribué au développement d'un système combinant des règles manuelles et une approche d'apprentissage automatique pour détecter les émotions. Notre objectif était de créer un système qui possède la précision des systèmes à base de règles, secondé par des algorithmes d'apprentissage automatique pour améliorer le rappel et les capacités de généralisation à de nouvelles données. Notre contribution a concerné l'apprentissage automatique, pour lequel nous avons entraîné un système de classement à vecteurs supports utilisant différents traits extraits des corpus d'apprentissage. Nous avons utilisé la bibliothèque LIBLINEAR Fan et al. (2008) avec un noyau linéaire et un paramétrage par défaut. Pour la classification multi-étiquettes, nous avons utilisé une stratégie en parallèle, c'est-à-dire que nous avons entraîné indépendamment un système de classement pour chaque émotion. Chaque système de classement fournit pour chaque phrase une indication de présence ou d'absence de l'émotion qu'il a été entraîné à détecter. Ainsi nous pouvons obtenir pour chaque énoncé, de 0 à 15 étiquettes d'émotion. La liste des traits utilisés pour l'apprentissage comprenait : des n-grammes, des d-grammes (chapitre 7), des étiquettes morpho-syntaxiques, des traits de la base *General Inquirer* (décrite dans le chapitre 5), des traits de la base ANEW (décrite dans le chapitre 5) et des traits heuristiques complémentaires (par ex. la position de la phrase dans la note).

Au final, notre algorithme de classement était le suivant :

1. D'abord nous avons entraîné un détecteur d'annotation, pour distinguer les phrases qu'il fallait annoter des phrases qui resteraient dépourvues d'annotation. Les traits utilisés ont été : les

étiquettes morpho-syntaxiques et les traits *General Inquirer*.

2. Ensuite, les phrases qui étaient supposées recevoir des annotations étaient traitées par un détecteur de subjectivité, afin de séparer les phrases objectives de celles subjectives. Les traits utilisés ont été : les traits heuristiques complémentaires, les étiquettes morpho-syntaxiques et les traits *General Inquirer*.

3. Parmi les phrases objectives, nous avons identifié celles qui contenaient des *information* et celles qui contenaient des *instructions*. Les traits utilisé ont été : les unigrammes, les bigrammes, les traits *General Inquirer*, les graphes de dépendances syntaxiques.

4. Les phrases subjectives ont été réparties en deux classes, celles qui contenaient des émotions positives et celles qui contenaient des émotions négatives. Les traits utilisés ont été : les étiquettes morpho-syntaxiques et les traits ANEW.

5. Les phrases à connotation négative ont ensuite été réparties entre les 7 classes suivantes: *chagrin* (sorrow), *désespoir* (hopelessness), *maltraitance* (abuse), *culpabilité* (guilt), *reproche* (blame), *peur* (fear), *colère* (anger). Les traits utilisé ont été: les unigrammes, les bigrammes, les d-grammes, et les traits *General Inquirer*.

6. Les phrases avec une polarité positive ont été elles réparties entre les 6 classes suivantes: *fierté* (pride), *espoir* (hopefulness), *amour* (love), *bonheur-tranquilité* (happiness-peacefulness), *reconnaissance* (thankfulness), *pardon* (forgiveness). Les traits utilisés ont été : les unigrammes, les bigrammes, les d-grammes et les traits *General Inquirer*.

Afin d'affiner le paramétrage de la partie apprentissage automatique de notre système de classement, nous avons effectué une validation croisée à 10 replis sur le corpus d'apprentissage. Les mesures de performance officielles étaient: la micro-moyenne en précision, en rappel , et en F-mesure (score F1). La moyenne officielle des F-score était 0.4875, le plus mauvais F-score 0.2967 et le meilleur 0.6139. L'approche à base de règles seule a obtenu : F-score = 0.4545, précision = 0.5662, rappel = 0.3797, tandis que notre meilleur paramétrage de l'approche combinant règle et apprentissage automatique a obtenu : F-score = 0.5383, précision = 0.5381 et rappel = 0.5385. Notre approche combinée a été classée sixième sur vingt-six.

Conclusion

La reconnaissance des sentiments dans les textes oblige les chercheurs à se confronter à de nombreuse questions d'analyse du langage, telles que l'analyse du discours, la résolution des coréférences, la reconnaissance des métaphores etc. Dans nos travaux, nous nous sommes

intéressés au classement en polarité, une des tâches fondamentales de l'analyse de sentiments qui vise à classer les documents en fonction de l'attitude qu'a le détenteur d'une opinion envers la cible de l'opinion. Même dans un cadre simplifié, cela reste une tâche difficile, d'autant plus que l'on fonctionne dans un environnement multilingue ou multi-domaines. Notre approche cherche à créer un système d'analyse de sentiments automatique et adaptatif qui soit indépendant de la langue et du domaine d'application concerné.

Notre contribution porte sur les éléments suivants :

- Nous avons montré comment utiliser les microblogs comme une source de données multilingue pour la fouille d'opinion et l'analyse de sentiments. Dans nos expériences, nous nous sommes servi de Twitter pour collecter un corpus de messages exprimant des opinions.

- Nous avons proposé une méthode automatique pour étiqueter les messages de Twitter comme positif ou négatif en considérant les émoticônes présentes dans les messages, comme des étiquettes bruitées. Nous avons ainsi obtenu un ensemble de messages positifs et négatifs pour quatre langues : 5,2 millions de messages en anglais, 672.800 en espagnol, 287.400 en français et 7.800 en chinois. Pour l'anglais, nous avons collecté un ensemble additionnel de 100.000 messages considérés comme neutres en polarité à partir des messages publiés sur Tiwtter par les journaux.

- Nous avons effectué une analyse linguistique du corpus collecté et observé que la distribution des étiquettes morpho-syntaxiques est différente entre le sous-corpus de messages subjectifs et le sous-corpus de messages objectifs, ainsi qu'entre le sous-corpus de messages de polarité positive et celui contenant les messages de polarité négative. Nous proposons d'utiliser la distribution des étiquettes morpho-syntaxiques comme trait supplémentaire pour le classement en polarité et la détection de subjectivité.

- Nous avons proposé une méthode pour la construction automatique de lexiques affectifs à partir de Twitter. Nous avons ainsi construit des lexiques pour l'anglais, l'espagnol et le français qui ont été évalués en vérifiant la corrélation des informations qu'ils contenaient avec le contenu de la base ANEW, considérée comme un standard du domaine.

- Nous avons utilisé les lexiques produits précédemment pour le classement en polarité de critiques de jeux vidéo en français dans le cadre des évaluations effectuées pour le projet DOXA. Les résultats de l'évaluation ont montré que les performances de notre approche sont comparables à celles obtenues avec une approche à base d'apprentissage automatique supervisé utilisant des n-grammes, alors que notre approche n'a pas besoin

de corpus d'apprentissage, car elle se satisfait des ressources extraites automatiquement à partir de Twitter.

- Nous avons proposé un nouveau mode de représentation des textes pour l'analyse de sentiments, que nous avons baptisé d-grammes et qui est basé sur les graphes de dépendances syntaxiques. Des évaluations effectuées avec trois analyseurs syntaxiques en dépendances différents, sur un jeu de données multi-domaines de critiques de produits en anglais et en français, ont montré que notre modèle de d-grammes permet d'obtenir une meilleure exactitude en classement de polarité; avec une amélioration de score pouvant aller jusqu'à 4,4% par rapport à l'approche traditionnelle à base de n-grammes.

- Nous avons exhibé une faiblesse de l'approche traditionnelle pour le classement supervisé en polarité pour ce qui concerne le classement des opinions minoritaires. Nous avons montré que les systèmes de classement ont tendance à s'appuyer sur les traits spécifiques aux entités cibles des opinions et, par voie de conséquences, qu'ils sont biaisés en faveur des opinions majoritaires.

- Nous avons proposé deux mesures pour normaliser les poids qualifiant l'importance d'un terme pour un classement en opinion : la fréquence moyenne d'un terme et la proportion d'entité. Des évaluations effectuées sur deux jeux de données en anglais, concernant cinq domaines applicatifs (films, livres, DVDs, électroménager, électronique) ont montré une amélioration des performance de classification des opinions minoritaires pouvant aller jusqu'à 12,5%.

Les approches proposées pour l'analyse de sentiment automatique et adaptative ont été testées avec succès dans les campagnes d'évaluation internationales suivantes :

- ROMIP 2011 : concernait le classement en polarité sur des critiques de produits commerciaux en russe. Parmi vingts systèmes participants, notre système a été classé quatrième dans la tâche de classification binaire pour le domaine des appareil photo électroniques, troisième dans la piste à trois classes pour le domaine des films, et premier dans la piste à cinq classes tous domaines confondus.

- SemEval 2010 : qui portait sur la désambiguïsation d'adjectifs ambigus exprimant des sentiments en chinois. Avec notre approche indépendante du langage, nous avons obtenu 64% de macro et 61% de micro exactitude (accuracy) et avons été classé sixième sur sept participants.

- I2B2 2011 : qui traitait de la détection des émotions dans des notes de suicide. Notre système a été classé sixième sur 26 avec

une F-mesure de 0.5383, qui était bien supérieure à la moyenne officielles des scores obtenus qui était de 0.4875.

Nous pensons que notre approche peut être facilement combinée avec d'autres travaux du domaine de la fouille d'opinion et l'analyse de sentiments parce que nos algorithmes sont facilement portables vers de nouveaux domaines applicatifs ou de nouvelles langues.

Bien que nous nous soyons concentrés dans cette thèse uniquement sur la classification de polarité, pour nos travaux futurs, nous envisageons le développement d'une approche combinée pour le problème de la fouille d'opinion et l'analyse de sentiments qui tienne compte de manière conjointe de tous les paramètres informationnels d'une expression d'opinion, c'est à dire l'expression, sa cible et sa source explicite ou implicite.

Les travaux présentés dans cette thèse ont fait l'objet à ce jour de publications dans 2 revues internationales, 5 conférences, 4 ateliers internationaux, et 1 chapitre de livre.

ACKNOWLEDGEMENT

I would like to thank my thesis advisor, Patrick Paroubek, for his trust and guidance during my research. Patrick has become not only my supervisor, but also a friend, helping and giving advices about life in France.

I would like to express my gratitude to the thesis committee members: Béatrice Daille, François Yvon, Patrick Gallinari, Suresh Manandhar, Yves Lepage, for their useful comments and challenging questions.

I would like to thank my colleagues, Anne-Lyse, Nicolas, Driss, Houda, Béatrice, and all those who helped me during the studies and made me feel less like a foreigner.

I thank my friends, my mom, my Anna, for their love and support.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

1

Sentiment analysis is quite a recent field of computational linguistics. It presents a challenging task for researchers. If the goal of computational linguistics is to process human language, the aim of sentiment analysis is to process human emotions, expressed in written text. A more formal definition can be found in a detailed survey[1] by Pang and Lee (2008, p. 6) "Opinion mining and sentiment analysis":

A sizeable number of papers mentioning "sentiment analysis" focus on the specific application of classifying reviews as to their polarity (either positive or negative) [. . .] However, nowadays many construe the term more broadly to mean the computational treatment of opinion, sentiment, and subjectivity in text.

1. "Opinion mining and sentiment analysis" survey is available at http://www.cs.cornell.edu/home/llee/opinion-mining-sentiment-analysis-survey.html

The Financial Times defines sentiment analysis as:[2]

A linguistic analysis technique where a body of text is examined to characterise the tonality of the document.

2. Financial Times definition of sentiment analysis: http://lexicon.ft.com/Term?term=sentiment-analysis

In our opinion, the Wikipedia provides one of the best definition[3]:

Sentiment analysis or opinion mining refers to the application of natural language processing, computational linguistics, and text analytics to identify and extract subjective information in source materials. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. The attitude may be his or her judgement or evaluation ([. . .]), affective state ([. . .]), or the intended emotional communication ([. . .]).

3. Wikipedia definition of sentiment analysis: http://en.wikipedia.org/wiki/Sentiment_analysis

The growing interest in processing emotions and opinions expressed in written text is partly due to the raise of the Web 2.0[4] and the user generated content. With the appearance of blogs, online forums, social networks Internet users received more ways to express themselves. An average blogger makes numerous posts about the likes/dislikes, thoughts on the politics, movie and book reviews, opinions on brands and products. Given the scale of the Internet, all this information becomes a valuable data for researchers and foremost for the industries in different fields readily available at no cost.

4. The Web 2.0 is characterized by rich internet applications, web-oriented architecture, and social web

Political scientists can use this information to determine which political candidate or party receives the most support. Sociologists

can estimate people's demand. Market researchers have probably the biggest interest in accurate and automatic processing of user opinions. They need to know the current trend and respond to it, making their products desirable for the broader audience possible. The security applications of sentiment analysis include antisocial behavior monitoring and forensic linguistics. Medical natural language processing (NLP) can use it for psychological studies. Lastly, public safety can benefit from opinion mining for risk prevention, in particular in health management.

An automatic and accurate sentiment analysis is of course difficult to achieve, because human language is too vague and ambiguous to be processed easily by a machine. Even humans have difficulties for identifying emotions and sentiments in texts. To simplify their task, researchers focus on subproblems that include: subjectivity detection, polarity classification, identification of opinion holder and target. In this work, we focus on polarity classification which aims at determining user attitude expressed in a given text towards an entity (opinion target). Even though it is probably the most well studied problem of sentiment analysis and not necessarily always the most frequent phenomena (Daille et al., 2011), it is still a challenging task. In Part I, we will cover the state of the art.

All the possible applications of sentiment analysis, that we have listed so far, contain in fact several domains. For example, market research may be needed to be done for movies, books, consumer electronics, cloth etc. Each of these application domains may need a special treatment. A sentiment analysis system should be adaptive to application and domain change. This is one of the problems we are trying to solve in this work. Another problem we tackle is multilingualism, i.e. we are trying to develop methods that will work in any given language. Finally, our system should be automatic, as our contribution lies within computational linguistics. We present our contribution in Part II.

It is important, that the solution we present can be applied to real world problems. Therefore it should be tested in a realistic environment. In Part III, we describe our participation in different evaluation campaigns, where we have tested our approach. Finally, we conclude our work and give directions to our future research in Part IV.

PART I

SENTIMENT ANALYSIS

This part presents theoretical background necessary for our research. We start by a broad overview of the research area known as opinion mining and sentiment analysis. We give our own definitions of the main concepts in order to reduce a possible ambiguity in later discussions. Finally, we focus on the task of polarity classification for it is our main research topic and review the existing approaches.

DEFINITIONS AND TERMINOLOGY

2

The terms *sentiment analysis* and *opinion mining* are often used interchangeably. According to Pang and Lee (2008) *sentiment analysis* is more often used by NLP researchers while *opinion mining* was adopted by the information retrieval (IR) community. Although we admit that these research fields are closely related and perhaps should be considered as one, in our research we use the term sentiment analysis that we distinguish from opinion mining. Let us start by defining some basic concepts before explaining in what manner these two terms differ.

2.1 Opinions, sentiments, and friends

While these are the key concepts of the research, there is no single convention on what to consider as an opinion or a sentiment. Many researchers however try to give definitions for opinions, sentiments, and emotions. Others consider them as a single concept focusing more on their properties (polarity, intensity etc.).

2.1.1 Opinions

opin·ion
a judgment about a person or thing
Source: Merriam-Webster

Opinions are usually defined as an opposite to facts. Many researches distinguish *factual* information (or *objective*) from *subjective*. Thus, if a fact is a piece of information which is commonly believed to be true (e.g. "The Sun rises in the East"), an opinion (subjective information) is the belief of an individual person. This implies that every opinion is assigned to its *holder*, since a different person may have a different belief about the same topic. Also, a claim is another necessary element of a belief.

The most common representation of an opinion is a tuple containing its properties. Kim and Hovy (2004) uses a quadruple [Topic, Holder, Claim, Sentiment] which is defined as

[. . .] the Holder believes a Claim about the Topic, and in many cases associates a Sentiment, such as good or bad, with the belief.

Kobayashi et al. (2007) also use the quadruple notation. However the elements are slightly different:

- *Opinion holder* A person who makes the evaluation.

- *Subject* An entity which is the target of the evaluation.
- *Aspect* A specific part of the entity which is being evaluated.
- *Evaluation* The quality of the aspect which forms the evaluation.

In this model, *opinion holder* is the same element as in the model of Kim and Hovy (2004), *subject* and *aspect* together correspond to *Topic*, and *evaluation* corresponds to a combination of *Claim* and *Sentiment*. Here is an example from Kobayashi et al. (2007) paper:

I just bought a Powershot a few days ago. I took some pictures using the camera. Colors are so beautiful even when the flash is used.

According to the definition of opinion by Kobayashi et al. (2007):

- Opinion holder = writer
- Subject = Powershot
- Aspect = picture, colors
- Evaluation = beautiful

According to Kim and Hovy (2004):

- Holder = writer
- Topic = Powershot pictures
- Claim = "colors are so beautiful"
- Sentiment = positive

The goal of opinion mining is identifying these tuple elements: opinion holder, opinion subject, claim and/or sentiment.

2.1.2 Sentiments

sen·ti·ment
an attitude, thought, or judgment prompted by feeling
Source: Merriam-Webster

As we have seen in the previous example, sentiments are sometimes considered to be a part of an opinion such as in the model of Kim and Hovy (2004). However, many researchers consider sentiments separately from opinions and define them as a personal judgement towards an entity expressed within a text (Thompson and Hunston, 2000). The main characteristics of this judgement are *polarity* and *intensity*. We can think of a sentiment as a vector: the direction of the vector would be its polarity and the length would be its intensity (Figure 2.1). Polarity can be either positive or negative, and intensity can be represented with a discrete or a continuous scale. A positive polarity corresponds to a positive evaluation such as appraisal, satisfaction, or joy. A negative polarity defines opposite feelings: anger, dissatisfaction, disappointment. Some researchers also add a *neutral* polarity which would correspond to a null vector and others also consider a *mixed* polarity for the cases when a person expresses positive and negative feelings at the same time.

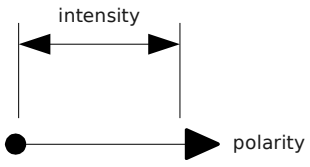


Figure 2.1: Sentiment as a vector

Sentiment analysis tasks include finding the sentiments expressed

in a text, determining their polarity (positive/negative/mixed/neutral) and intensity (week/strong).

2.1.3 *Emotions*

e·mo·tion
a conscious mental reaction (as anger or fear) subjectively experienced as strong feeling usually directed toward a specific object and typically accompanied by physiological and behavioral changes in the body
Source: Merriam-Webster

The concept of emotions can be seen as a refinement of sentiments with a finer granularity of information. While sentiments are usually viewed as positive or negative, emotions define more particular types of human behavior or a mental state. Researchers usually define a small set of basic emotions that are used to derive more complex ones. Here are some emotions that are usually included in the basic set: anger, sadness, happiness, love, fear, guilt, and others. Figure 2.2 shows Plutchik (2001)'s model of emotions represented in a form of a wheel. The wheel contains 8 basic emotions each having an opposite one.

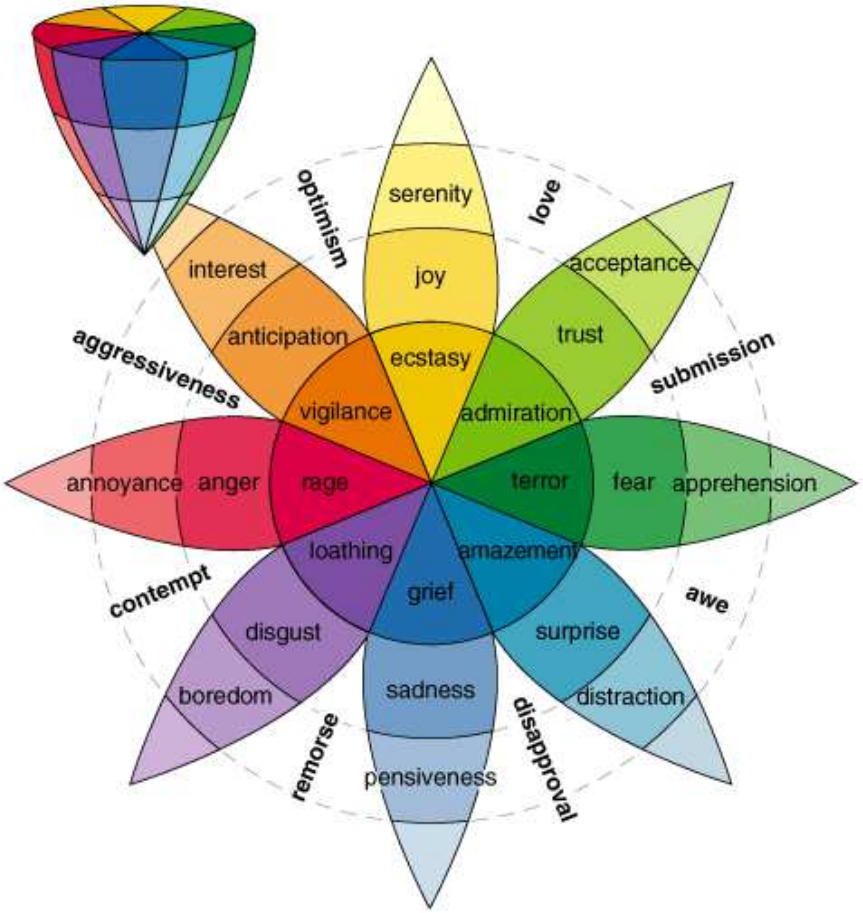


Figure 2.2: Plutchik's wheel of emotions. *Source: American Scientist*

Emotion classification is usually considered as a problem of identifying basic emotions in text.

2.1.4 *Affects*

af·fect
the conscious subjective aspect of an emotion considered apart from bodily changes; also: a set of observable manifestations of a subjectively experienced emotion
Source: Merriam-Webster

Affects refer to the actual expression of emotions, thus the field of affective computing studies recording of human behaviors, captured on video, audio, photographic images, and sensors. Written text does not capture human behavior explicitly, thus a term "affective computing" is used in text processing less often than *sentiment analysis*, *opinion mining*, or *emotion detection*. Some researchers however prefer to use this terminology in their work (Read, 2004; Neviarouskaya et al., 2010).

2.1.5 *Mood*

mood
a conscious state of mind or predominant emotion
Source: Merriam-Webster

Mood is very similar to emotions and can be described with the same categories, such as sadness or happiness. However, an emotion is usually caused by some event, while a mood has not explicit cause. Also, in text, a mood is usually been described in the beginning, such as a background for other events, whereas emotions are described as consequences of the happening. An example of emotion vs. mood taken from Twitter messages:

It makes me sad to see bamboo trees but no pandas.

In this message, the author expresses sadness emotion caused by a danger of extinction of giant pandas.

Music makes me feel so better when I'm sad.

In this message, the author describes a sad mood that causes an event of listening to music to feel better.

Since emotions are caused by events they are more relevant to information retrieval while mood suits better for research in medical domain. For example, marketing researchers are interested in emotions invoked by the products and psychologists are interested in mood of patients. Nevertheless, both terms can be easily confused and most of the time are used interchangeably in research (Jung et al., 2006; Généreux and Evans, 2006).

2.2 Summing up

As there is no single convention on the terminology and researchers use different terms to describe their work in the same research field, we want to define the concepts we will be working on in order to reduce a possible ambiguity in terminology.

We define **opinion** as the statement of an individual person about a specific object or topic. We call the person who makes a statement **opinion holder** and the subject of the statement **opinion target**. Thus, the term **opinion mining** refers to a field of computational linguistics

that studies opinions. We oppose opinions to **facts**, which is an information proved to be true such as common knowledge. Our definition of opinion requires it to have holder and claim, otherwise it cannot be considered as opinion. For example, a phrase "I am cold" is not an opinion because there is no claim and it is rather a state of a fact, but "I think it is cold in this room" is an opinion with room temperature being the opinion target. Similarly, "Economy is in recession" is not an opinion because the opinion holder is not explicitly mentioned, but "The minister believes that the economy is in recession" is an opinion with the minister being the opinion holder.

We define **sentiment** as a judgement of an individual person about a specific object or topic which is characterized by **polarity** and **intensity**. We refer to **sentiment analysis** as a field of computational linguistics that studies sentiments. We assume polarity to be positive, negative, or both (mixed). Intensity shows the degree of sentiment positiveness or negativeness and varies from week to strong. From our definition follows that **a sentiment is a particular type of an opinion which has a polarity**. Thus, we oppose sentiments to neutral opinions and facts.

We also distinguish **emotions** and **mood** which we define as a mental state affecting individual behavior. As compared to sentiments, emotions have a holder, but do not necessarily have a specific target. Mood is similar to emotions, but does not have a specific cause. We illustrate the difference between all the defined concepts on the following sample text:

It was a rainy day. I felt sad and lonely, so I went shopping. I have bought the last year album of my favorite band. Their music made me feel much better. I think it is their best album so far. I am sure they will release a new one this year.

- *It was a rainy day* – fact
- *I felt sad and lonely* – mood (there is no explicit cause)
- *I went shopping* – fact
- *I have bought the last year album of my favorite band* – fact
- *Their music made me feel much better* – emotion (caused by listening to music)
- *I think it is their best album so far* – positive sentiment (a positive judgement about the album)
- *I am sure they will release a new one this year* – neutral opinion (there is no polarity in the statement)

We do not claim our definitions to be complete and able to handle any possible case. We are also aware of the fact that many applications do not require such a specific distinction of the concepts. However, these are the settings we use in this work. When citing other research materials, we will use the terminology used by their authors.

In our work, we did not apply the appraisal theory (Martin and White, 2005), since we consider it to be cumbersome for the practical use with the statistical method that we use in our framework. However, one may find it more suitable for modeling sentiments and opinions. Note that like in our proposed model of sentiment which requires a target, appraisal theory (Bednarek, 2009) also requires a target. Charaudeau (1992) goes further by precising in his work that an opinion expression can take two forms of appraisal:

- epistemic opinions, which are qualify the target against a knowledge based criteria,
- axiologic opinions which corresond to a value appreciation proper.

OPINION MINING AND SENTIMENT ANALYSIS TASKS

3

In this chapter, we describe the common tasks of sentiment analysis and opinion mining. The plan is similar to the one proposed by Liu (2010). We provide a definition of each task, common issues and main approaches. In the next chapter, we shall focus on the task of polarity classification, the main topic of our research.

3.1 Subjectivity analysis and opinion detection

Subjectivity analysis is a problem of detecting the presence of opinions in a given text. Simply speaking, it determines what is opinion and what is not. Since different authors define sentiments and opinions in a different manner, subjectivity analysis can deal with sentiments (separating polar text from neutral text) as well as opinions (separating subjective statements from facts). Although, this problem is one of the basic ones, it is also one of the most difficult because it is extremely difficult to define even for humans what is an opinion, what is a sentiment and what is neither.

Generally, subjectivity analysis is considered a binary classification problem: for a given text the system should return *true* if it contains opinions/sentiments and *false* otherwise. A more advanced problem is to identify borders of opinions or sentiments in a text. The common approach is to use a machine learning classifier trained on two sets of texts representing positive and negative samples. It is relatively easy to collect a set of texts containing only facts and another containing opinions. However, it is difficult to compose a homogeneous corpus, which contains texts of the same genre, style, and topic. If that is not the case, there is a great chance that a classifier would learn a model that distinguishes between different text sources instead of identifying subjectivity.

Pang and Lee (2004) augmented the polarity classification framework (Pang et al., 2002) with a additional preprocessing step where sentences from an analyzed text are being classified as subjective or objective. The authors translated subjectivity classification into a graph-partitioning problem and used the min-cut max-flow theorem to solve it. The sentences labelled as "subjective" are extracted and passed to a general polarity classifier (bag-of-words model with SVM). They reported a statistically significant improvement of the

classification accuracy from 82.8% to 86.4%.

3.2 Polarity classification

In general, the task of polarity classification is to determine whether a text expresses a positive or a negative attitude of its author towards the topic of the text. A more advanced task is to define the degree of polarity (week or strong) and also be able to identify mixed polarity, which is positive and negative at the same time. Polarity classification is the main topic of our work, more on the issues and the state of the art can be found in Chapter 4 (p. 21).

3.3 Opinion holder and target identification

Another basic task of opinion mining is identification of opinion holder and target. In other words, we need to know who holds the opinion and what the opinion is about. The purpose of this task is to filter opinions that are relevant to the given topic, since there can be several opinions in a text on different subjects. For example, if it is a movie review and the author writes about the experience of going to a cinema, we want to extract only opinions related to the movie and not about the cinema. Knowing the opinion holder helps us to estimate the demographics or collect opinions of a specific person. The latter is useful for user personalization, i.e. selecting topics that a specific user prefers and avoid things the user does not like.

Coreference resolution
is the process of determining if two expressions in natural language refer to the same entity in the world (Soon et al., 2001)

The task of identifying the opinion holder and the opinion target is relatively difficult because of coreference chains, i.e. the same entity can be addressed in many ways. For example, a movie can be referred to by its title, by a pronoun (*it*), by a noun phrase (*movie*, *motion picture*). The title of the movie may also have its variations. A good coreference resolution system is needed to handle this problem (Stoyanov and Cardie, 2008).

In many cases, however, the task is obsolete. When working with the data from the Web, the opinion holder is often the author of the review or the blog post. The topic is usually also known. Movie reviews are usually located on a page dedicated to a specific movie (Figure 3.1), thus we can assume that all the reviews on this page are about the same movie. Nevertheless, for an advanced opinion mining system, we want to make sure that the opinions we extract are related to the movie itself and also belong to the review author, for example a review may contain a quote or a reference to other critic's opinion.

3.4 Opinion summarization

Automatic summarization is a well known problem in text analysis. The task of summarization is to provide a shorter version of a text

Figure 3.1: A movie page with user reviews. *Source: a screenshot from IMDb.com*

preserving its main idea. For example, search engines provide short descriptions of web pages in search results (Figure 3.2).

Opinion summarization is a similar task in opinion mining domain. Its main goal is to extract opinions from the text and present them in a shorter form. In general, the problem is seen as extracting opinionated phrases from the original text with the following reordering if necessary. Nishikawa et al. (2010) translated the opinion summarization problem into an integer linear programming[1] formulation by maximizing the scoring function which measures opinion weights of sentences constraint as subject to the coherence of the extracted sentences.

1. **Integer Linear Programming** problem of optimizing a linear function of several integer variables subject to linear constraints on these variables

3.5 Irony identification

Computational treatment of irony is another task to be solved in order to obtain an accurate sentiment analysis system. Rhetorical theory distinguishes several types of irony, among them **verbal irony** is the one that became a subject of sentiment analysis research. Abrams

iro·ny
the use of words to express something other than and especially the opposite of the literal meaning
Source: Merriam-Webster

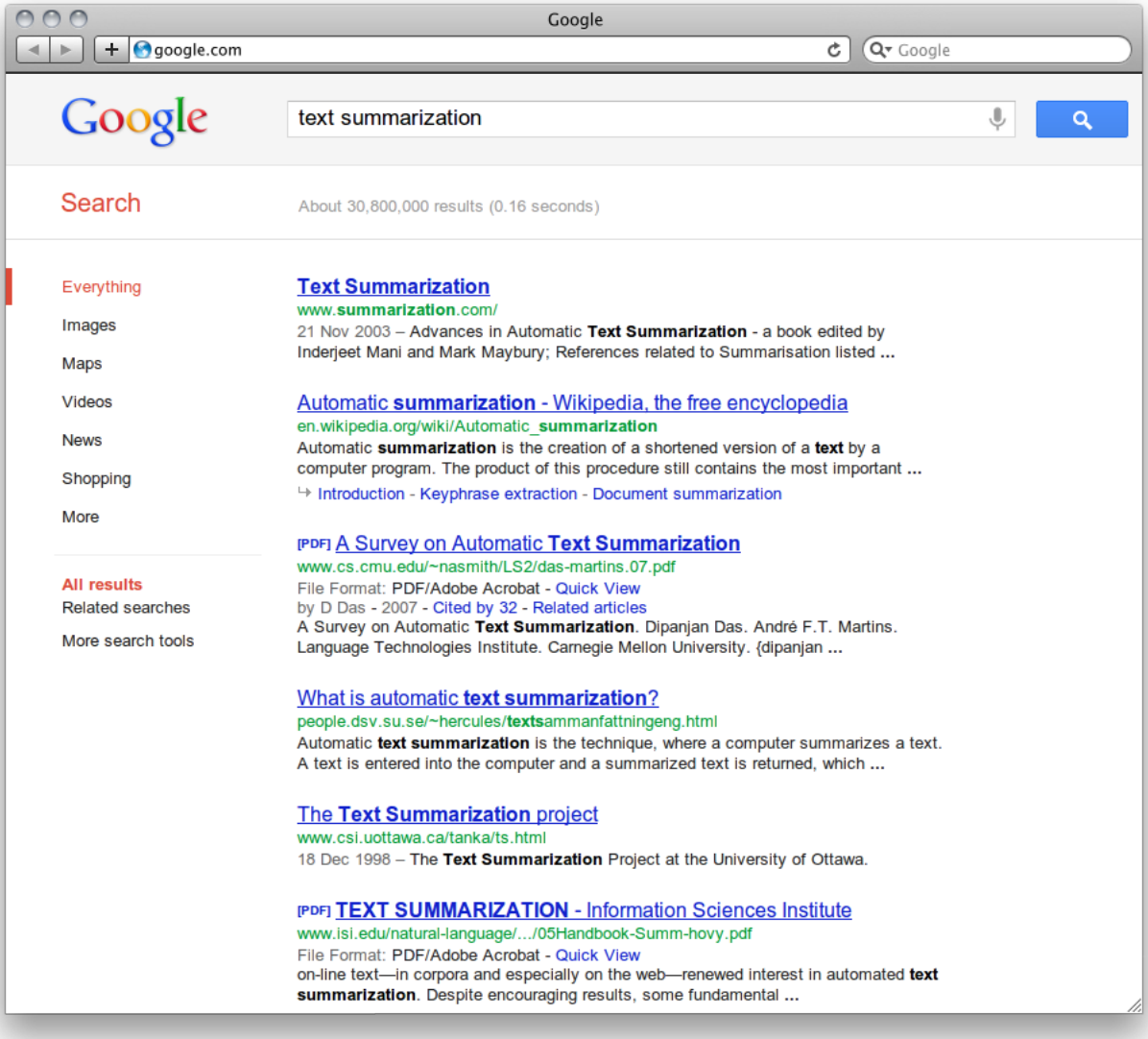


Figure 3.2: Google search results page includes short descriptions of found web pages (snippets)

and Harpham (2011) define verbal irony as follows:

Verbal irony is a statement in which the meaning that a speaker employs is sharply different from the meaning that is ostensibly expressed. The ironic statement usually involves the explicit expression of one attitude or evaluation, but with indications in the overall speech-situation that the speaker intends a very different, and often opposite, attitude or evaluation.

sar·casm
a sharp and often satirical or ironic utterance designed to cut or give pain
Source: Merriam-Webster

The term **sarcasm** is often used interchangeably with irony when describing the task of irony identification. Some dictionaries define sarcasm as a type of verbal irony, while other ones state that there may be no relationship between these two terms. Researchers in computational linguistics usually define irony as expressing negative emotions by using words with positive literal meaning (or vice versa).

Carvalho et al. (2009) have reported that 35% of polarity classification errors were due to verbal irony when applying their polarity detection rules to analyse political debates. Other researchers have also reported that irony causes misclassification of polarity since it is used to express negative sentiments with positive phrases. Existing approaches to irony identification in text are based on pattern matching and machine learning. Carvalho et al. (2009) presented a set of patterns to serve as clues for distinguishing ironic phrases from non-ironic ones. The most frequently observed patterns were those capturing interjections, punctuations, quotation marks, and laugh indicators (e.g. *LOL*). Reyes and Rosso (2011) applied a machine learning approach. The list of features used in their method consisted of n-grams, POS n-grams, and profiling features that were obtained by looking up words from the examined text in different dictionaries such as affective lexicons. A semi-supervised algorithm for sarcasm identification based on kNN[2] with syntactic and pattern matching features has been proposed by Tsur et al. (2010). The authors applied their approach for classification of Amazon reviews and later to sarcasm identification in Twitter messages.

One of the difficulties of irony study is to collect a corpus of ironic texts. Carvalho et al. (2009) manually annotated reviews, however this process is time and resource consuming thus poorly adaptive to other domains. Reyes and Rosso (2011) built a corpus of ironic comments automatically by collecting reviews with abnormally high rated comments to particular products at Amazon[3]. However, this approach is quite limited as it allows to retrieve comments only for these products. Davidov et al. (2010) used a more scalable approach using noisy labels discussed in § 4.3.2 (p. 30). They collected a corpus of Twitter messages labeled with a hashtag *#sarcasm*. The same approach has been used later by González-Ibáñez et al. (2011) to collect a corpus of sarcastic messages from Twitter. *Hashtags* are a special way to incorporate tags in messages in Twitter which is used by its users to label people, places, events, mood, etc. A hashtag *#sarcasm* is used by Twitter users to specify that the message is sarcastic such as in the following example:

The only thing better than a forecast calling for snow or rain is one calling for snow AND rain. Yippee. #Sarcasm

We describe Twitter data in more details in Chapter 6 (p. 51).

Irony identification remains an open problem for opinion mining and sentiment analysis. Existing approaches apply only commonly used methods such as supervised learning and pattern matching, while the achieved results are far from a human-like performance.

2. **k-nearest neighbor algorithm (k-NN)**
a method for classifying objects based on closest training examples in the feature space

Amazon.com
an online retailer
http://www.amazon.com

3. An example of an ironic comment for a product "The Mountain Three Wolf Moon Short Sleeve Tee":

Rating: 5 out of 5
I just purchased this t-shirt and I already feel a surge of magical energy. Anyone who does not own this shirt will never know the power of pure joy. Now everyone will know how much of a Boss I am when they gaze upon the trinity of wolves.

3.6 Opinion spam identification

Public opinions do not only serve as a feedback for product manufacturers (service providers, political parties, etc.), but they also influence other people's decision when choosing a product (using a service, voting for a political candidate). Hence, many parties are interested in publishing positive opinions about themselves and negative ones about their competitors. In such an environment, the question of opinion authenticity rises, as nowadays the Web provides many ways to express opinions publicly without any means of verification of their trustworthiness, creating a new research direction called **identification of opinion spam**.

Opinion spam (or untruthful opinions, deceptive opinions) are addressed in several forms:

1. self promotion or deliberately created reviews by hired authors to promote a product or a service
2. negative reviews which purpose is to damage the competitors reputation

In addition, researchers also define opinion spam as repetitive low quality reviews created for advertising purposes. Opinion spam can be harmful for automatic systems that process opinions, since they add noise to the training data.

A typical challenge of such studies, is to collect a sufficiently large annotated corpus. Jindal and Liu (2008) applied heuristics to extract untruthful reviews from Amazon. To collect such reviews, authors performed an analysis of 5.8 million reviews from 2.14 million reviewers to find authors posted similar repetitive comments about products which were considered as opinion spam. However, in a general case, an authorship information may not be available to researchers and all there is to analyse is just a text of the review.

To build a gold standard, Ott et al. (2011) with the help of crowdsourcing (see § 4.3.3, p. 30) composed a collection of fake reviews about hotels to be used in addition to a set of filtered reviews collected from TripAdvisor[4] which were considered as truthful. A part of the constructed gold standard was given to three volunteers to perform an analysis of human performance at identifying deceptive opinions. These results were compared to performance of machine learning based approach that used n-grams and features produced by Linguistic Inquiry and Word Count (LIWC)[5] software developed by Pennebaker et al. (2007). The comparison revealed that even a simple n-grams based classifier performs better than human experts at classifying reviews into deceptive and truthful. Moreover, human experts were biased towards the truthful decision (i.e. labeling most of the reviews as truthful), while a classifier based on a combination of n-grams and LIWC features yielded up to 89.9 of F-score estimated by 5-fold cross validation.

4. **TripAdvisor**
a travel website that gathers information and posts reviews of travel-related content.
http://tripadvisor.com

5. **Linguistic Inquiry and Word Count (LIWC)**
a text analysis software program. LIWC calculates the degree to which people use different categories of words in texts.
http://www.liwc.net/

As the opinions in the Web gain more importance, it will be more abused by malicious parties what makes the task of opinion spam identification an important problem for future opinion mining and sentiment analysis.

POLARITY CLASSIFICATION IN DETAIL

4

Polarity classification is probably the most studied subproblem of sentiment analysis. Yet, we cannot report a computational solution that performs nearly as well as human experts. In this section, we focus on polarity classification, the main topic of our research. We formulate the problem definition, describe the issues, the resources, and the evaluation methodology.

4.1 Problem definition

Most of the time, polarity classification is considered a binary classification problem. The goal of the task is to determine if a given text expresses a positive attitude of its author regarding a specific topic. This attitude is called opinion polarity and in case of binary classification it takes either a positive or a negative value. While it is intuitive for humans which opinion to consider positive and which one a negative, we need to formulate the problem in a more specific manner, in order to develop a computational solution. Inspired by the keynote speech of Hovy (2011), where the author made a strong connection between opinions and goals, we formulate the key concepts of the problem as follows.

Given our definition of opinions and sentiments in § 2.2 (p. 10), we further define **beneficial action** as an action that brings a benefit to the opinion target owner. For example, in case of movie reviews, the opinion target is a movie, the owner is a movie production studio, and the beneficial action would be buying a cinema ticket to watch the movie or DVD rental. In political domain, the opinion target is a political candidate and the beneficial action is voting for the candidate. Thus, **sentiment polarity is said to be positive if the opinion supports the beneficial action** and **it is said to be negative if it opposes the beneficial action**. Sentiment intensity in this case measures the degree of the support or the opposition. The support or the opposition do not have to be explicit. For example, giving a good review for a movie does not explicitly force the reader to buy the DVD, however it motivates to watch a movie and consequently leads to a purchase.

4.2 Issues

Below is a list of issues that make polarity classification a difficult task. This list is by far not a complete one, however these are issues that we consider in our research, because they are frequently encountered in the literature.

4.2.1 Discourse analysis

The main reason why polarity classification is such a difficult task is because a simple analysis of text is not sufficient. Though simple cases easy to analyse such as "The soundtrack was awful" are frequent, in general, opinions are expressed using complex language constructions that require discourse analysis and real-world knowledge to be classified correctly. Consider the example below[1]:

1. The whole text is available here: http://www.imdb.com/reviews/310/31018.html

[. . .] The actors are pretty good for the most part, although Wes Bentley just seemed to be playing the exact same character that he did in AMERICAN BEAUTY, only in a new neighborhood. But my biggest kudos go out to Sagemiller, who holds her own throughout the entire film, and actually has you feeling her character's unraveling.

Overall, the film doesn't stick because it doesn't entertain, it's confusing, it rarely excites and it feels pretty redundant for most of its runtime, despite a pretty cool ending and explanation to all of the craziness that came before it. [. . .]

This is a typical example of a movie review taken from the dataset collected by Pang et al. (2002) and later used in many sentiment analysis research. The author does not simply give his opinion about the movie, but makes a review of its good and bad sides, provides separate judgements to the plot, actors, and overall realization. For a sentiment analysis system, to make a correct analysis, it has to follow the discourse to find the conclusion.

A recent research by Zirn et al. (2011) proposes an approach for fine-grained sentiment analysis using discourse parsing. The authors use a Markov logic network to combine information about polarities of sentence segments and discourse relations[2] between them. The proposed system showed a better performance than a traditional machine learning based approach which does not use discourse information.

2. The authors use *contrast* relation and group all other relations as *nocontrast*

In general, it is very difficult to perform a discourse analysis. Therefore, in our work we have to simplify the tasks by making very naïve assumptions (somewhat similar to assumptions made by Yarowsky (1995) for word sense disambiguation[3])

3. Yarowsky (1995) made two assumptions for WSD: **one sense per discourse**: the sense of a target word is highly consistent within any given document; **one sense per collocation**: nearby words provide strong and consistent clues to the sense of a target word, conditional on relative distance, order and syntactic relationship.

- **One opinion target per document** We assume there is only one entity which is the opinion target of the whole text.
- **One sentiment polarity per document** We assume there is one

overall sentiment whose polarity is needed to be classified.

By making these assumptions, we are not obliged to perform discourse analysis and examine each phrase separately since we assume there is only one overall opinion and each part of the text contribute to it without a conflict.

4.2.2 Figurative language

Figurative language is a manner of expressing ideas by using unusual meaning of words (i.e. different from those specified in common dictionaries). It is often used when expressing emotions and opinions in text since it can be interpreted as a sign of creativity or informality. The following example, taken from the subjectivity dataset by Pang and Lee (2004) depicts usage of figurative language in movie reviews:

[. . .] there are enough moments of heartbreaking honesty ***to keep one glued to the screen****. [. . .]*

[. . .] it won't ***bust your gut*** *– and it's not intended to – it's merely a blandly cinematic surgical examination of what makes a joke a joke . [. . .]*

[. . .] you might not ***buy the ideas*** *. but you'll definitely want the t-shirt . [. . .]*

Because the words are used out of their literal meaning, lexicon based approaches normally fail to correctly analyse such texts. Machine learning methods in their turn also face many difficulties as in general there is not enough data to learn the phenomena since figures of speech vary greatly and every author is able to compose one never seen before.

Rentoumi et al. (2009) have presented a machine learning (ML) based approach to sentiment analysis of figurative language using word sense disambiguation. The authors noticed that ML approach have much difficulties in processing such data and thus proposed an improved framework (Rentoumi et al., 2010) which combines ML with pattern matching rule based system for polarity classification.

In general, treatment of figurative language is a difficult task and it is often omitted in sentiment analysis research which is not particularly dedicated to solving this problem. Our approach to polarity classification does not directly tackle this problem, however, in the base of our system we use machine learning, which we consider to be less sensible to the issues posed by figurative language.

4.2.3 Negations

Treatment of negations in text is often mentioned in sentiment analysis as it is considered by many researchers to be important. Negation by its definition inverts meaning of words and phrases (e.g. *good* vs

not good) what makes it indeed important for polarity classification. In general it is relatively easy to identify negations (e.g. by doing a simple pattern matching looking for a negation particle) there are still some minor issues, such as possible spelling errors (e.g. *dont* instead of *don't*) and language specific issues. For example, in French, the word *personne* can mean a negation pronoun *nobody*, an indefinite pronoun *anybody* as well as a noun *person*. Below are examples taken from Twitter (original text on the left, English translation on the right).

*Sarkozy attaque plus la personnalité que le programme d'Hollande... oui parce que **personne** ne croit que hollande va appliquer son programme*	*Sarkozy attacks more the personality than the program of Hollande... Yes, because **nobody** believes that Holland will apply his program*
*J'entends pas mal de déçus de Sarkozy qui revoteraient pour lui face à Hollande. Il a du mal à convaincre sur sa **personne**.*	*I hear a lot of disappointed with Sarkozy who would re-vote for him against Holland. It's hard to convince on his **person**.*

There are different ways to capture negations. Our approach is to treat it locally by attaching a negation particle (such as *no*, *not*) to nearby words:

$$\text{unigrams}(S) = \{(\text{The}), (\text{movie}), (\text{was}), (\textbf{not}), (\text{good})\}$$
$$\Downarrow$$
$$\text{unigrams}(S) = \{(\text{The}), (\text{movie}), (\text{was } \textbf{not}), (\textbf{not } \text{good})\}$$

We also use dependency parsing to find the negation relation (see Chapter 7 (p. 65)). Another way is to add a binary feature representing if a negation is present in a text (Pang et al., 2002). Klenner et al. (2009) use negations to inverse the polarity of the negated word. In practice, however, negation treatment brings almost no or little benefit, but since usually it does not degrade the performance, researchers prefer to keep it since it "brings no harm".

A good survey on negations role in sentiment analysis can be found in the work by Wiegand et al. (2010).

4.2.4 Domain adaptation

In most cases, a sentiment analysis system is created for a specific task which assumes a single topic, for example, "polarity classification in movie reviews". However, there are many applications when a system has to analyse documents in many topics, for example, classifying product reviews. Analyzing different topics poses additional problem to sentiment analysis. If test and training data are not the same, probability distribution of language models are different which degrades the performance. In other words, features from the test data

may not be present in training data or what is worse they may have different meanings.

Let us consider *electronics* and *kitchen appliances* as two topics to work with. While they are quite close in vocabulary, there are cases when the same words provide different sentiments. In the examples below, taken from the product review dataset by Blitzer et al. (2007) collected from Amazon, the same word *hot* is present in both reviews. The first review (on the left) from kitchen appliances domain is positive because it speaks about a coffeemaker. In this context *hot coffee* signifies a positive experience. However, the second review (on the right) which speaks about a radio receiver, uses the word *hot* to describe a negative experience, because it is not good when an electronic device gets hot (unless it is a heater).

[. . .] Overall, this device come highly recommended for anyone who might like to have a ***hot*** *drink when there's none available [. . .]*	*[. . .] I had no problems picking up a signal unless I was in a long tunnel or deep valley. I did notice the unit did get* ***hot*** *during the trip and I think that is why it is acting the way it is now [. . .]*
– from a review for "Zelco Brisk Brew Travel Coffeemaker"	*– from a review for "Delphi SA10085 Roady2 XM Satellite Radio Receiver with Built-in Wireless FM Modulator"*

This problem known as **domain adaptation** studies how to adapt a system trained on one type of data to be able to classify samples from a different data source. It is a well studies problem in general which has been also applied to polarity classification. Blitzer et al. (2007) proposed a method based on their previously reported method on domain adaptation (Blitzer et al., 2006) called **structural correspondence learning** (SCL) which finds a correspondence between features in source and target domains[4]. They illustrate the method with an example of domain adaptation between a *computers* domain and a *cell-phone* domain:

4. Source domain is the one on which the system has been trained, target domain is the one to which we want to adapt the system.

[. . .] While many of the features of a good cell phone review are the same as a computer review – the words "excellent" and "awful" for example – many words are totally new, like "reception". At the same time, many features which were useful for computers, such as "dual-core" are no longer useful for cell phones.

Our key intuition is that even when "good-quality reception" and "fast dual-core" are completely distinct for each domain, if they both have high correlation with "excellent" and low correlation with "awful" on unlabeled data, then we can tentatively align them. After learning a classifier for computer reviews, when we see a cell-phone feature like "good quality reception", we know it should behave in a roughly similar manner to "fast dual-core". [. . .]

In many cases though it is not a problem to train a model for a new domain. If we are to work with product reviews, it is easy to collect reviews in different topics (we cover collection of annotated data in § 4.3.2, p. 30). However, if the classification approach depends on resources that are domain specific (e.g. lexicons, ontologies) it makes it more difficult to adapt such a method to other domains. That is why, in the base of our work, we do not rely on any domain specific resources such as product ontologies, since we want to build an adaptive framework which can be applied to any domain.

4.2.5 Multilingualism

Nowadays, the demand for multilingual systems is increasing. International manufacturers want to know consumer feedback worldwide, governments need to monitor their multicultural communities, internet applications have to respond to requests in different languages. Many approaches to polarity classification are however dependent to lexical resources and NLP tools that only exist in some languages. For such a system to be able to process another language, one needs to localize lexical resources and adapt the software (e.g. lexical parsers, text analyzers, etc.). Localization of lexicons is a difficult and an expensive task because it requires competent human translators. Translating an affective lexicon can be a tricky problem due to cultural differences in languages. For example, a word *aimer* in French can be translated into English as two words with different semantics: *to love* or *to like*.

Figure 4.1 depicts a part of infographics from the The Unspeakableness project[5] by Pei-Ying Lin who investigates human emotions. The whole figure represents words in different languages describing emotions that do not have a direct translation in English. The words are presented as nodes in a graph with relations to Parrott (2001) classification of human emotions. These examples illustrate difficulties to be faced with when localizing an affective lexicon.

5. The Unspeakableness project: http://untranslatable.peiyinglin.net/

Adapting a software that was not initially created for a multilingual application may be also complicated. A relatively easy task of text tokenization can face difficulties such as character encoding or defining word boundaries. For example, in French, the apostrophe (') is used in place of a vowel letter to indicate elision[6] such as *j'écoute* (I listen), which we want to split into two words *je* (I) and *écoute* (listen). In English, however, apostrophe can indicate contractions (*can't* – cannot) or abbreviations (*gov't* – government) where it does not serve as a word separator. Moreover, many languages use whitespace as a word boundary, while others (Chinese, Japanese, Thai) do not have a special symbol for separating words in a sentence.

6. **Elision** orthographic convention in French by which the deletion of a vowel is reflected in writing, and indicated with an apostrophe.

Another approach to a multilingual sentiment analysis is by using machine translation[7] (MT). In this approach, MT can be used either for translating training data before we build a model or to translate the data to be analysed by an already trained model. Several studies

7. **Machine translation** a sub-field of computational linguistics that investigates the use of software to translate text or speech from one natural language to another.

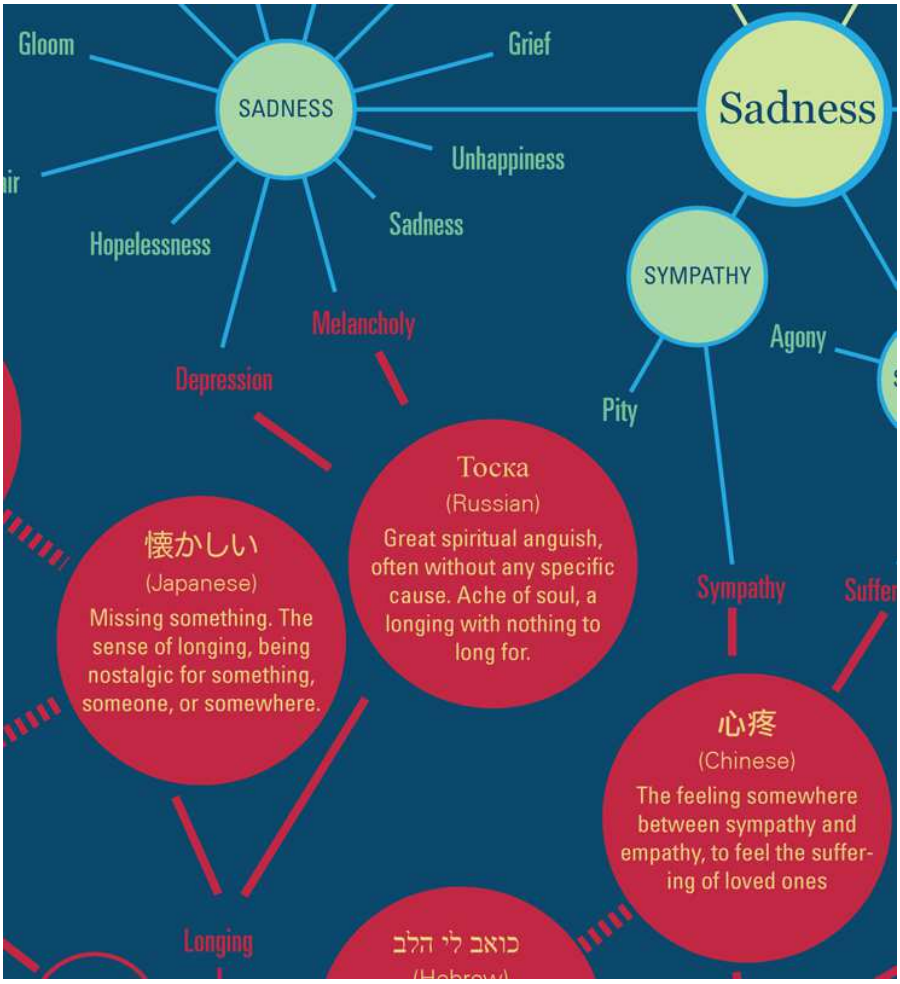


Figure 4.1: "Untranslatable words about emotions in languages other than English" by Pei-Ying Lin, Design Interacitons, Royal College of Art. Red discs represent emotion words that do not have exact translation in English, yellow discs represent emotions according to Parrot's model.
Source: http://untranslatable.peiyinglin.net/

*"No single word in English renders all the shades of **toska**. At its deepest and most painful, it is a sensation of great spiritual anguish, often without any specific cause. At less morbid levels it is a dull ache of the soul, a longing with nothing to long for, a sick pining, a vague restlessness, mental throes, yearning. In particular cases it may be the desire for somebody of something specific, nostalgia, love-sickness. At the lowest level it grades into ennui, boredom."*
– Vladimir Nabokov

showed promising results (Prettenhofer and Stein, 2010, Duh et al., 2011) even though lower accuracy is observed when applying MT as compare to monolingual settings. In Chapter 9 (p. 91), we describe how we apply MT to cover the lack of training data.

In our work, we aim at creating a framework for polarity classification which is easily adaptable to new languages. In the core of our framework, there is a machine learning classifier that simply requires training data in the target language. We consider that it is much easier to collect training data in different languages rather than localizing lexical resources (we cover approaches for automatic data collection in the next section § 4.3, p. 27).

4.3 Data

All the approaches to polarity classification need annotated data either for training the model or for evaluating system performance.

Below, we review common approaches to constructing corpora for sentiment analysis and particularly for polarity classification, as well as give an overview of existing datasets.

4.3.1 *Manual annotation of raw text (the DOXA project annotation)*

Manual data annotation is the easiest way to fill the lack of annotated data, although it is as well the most resource consuming. To construct an annotated corpus, one needs first to collect raw data, next to apply chosen in advance annotating scheme and finally to validate the produced annotations (Wiebe and Cardie, 2005; Toprak et al., 2010).

DOXA[8] is a project (DGE nº 08-2-93-0888) supported by the numeric competitiveness center CAP DIGITAL of Île-de-France region which aims among other things at defining and implementing an OSA semantic model for opinion mining in an industrial context. We have developed the annotation scheme for the manual annotation and performed the evaluation of participants' systems.

8. http://www.projet-doxa.fr

Origin of the model

The model that was selected to serve as an initial basis for opinion mining is the one proposed by Mathieu (2000), which relies on a study of the verbs used in French to express sentiments. The linguistic theory underpinning its model is the Lexique-Grammaire by Gross (1968). In Mathieu (2000) work, the verbs expressing sentiment were divided in homogeneous semantic classes (for instance in one class you could find all the verb expressing joy) further refined into classes based on common linguistic characteristics, mainly subcategorization frames.

The initial 38 verb classes of the model, divided into three main semantic categories according to the polarity expressed (pleasant, unpleasant, indifferent), were reduced to ten classes, following preliminary experimentation at manual annotation of the corpora used in DOXA. The resulting classes were then further organized into a hierarchical network using as structuring guideline the intensity and polarity of the sentiment expressed. New classes were then added to the model to account for opinion expressing an appreciation or a judgement.

Opinion annotation at macro and meso levels

Taking into account the need of the end-users of the DOXA project, three levels of analysis were defined in DOXA for opinion mining:

- macro, which corresponds to the document level,
- meso, for the section level, which in DOXA is defined as a text span corresponding roughly to a paragraph, but based entirely

Attribute	Value
semantic category	recommendation_suggestion, demand_query, . . . (*one or more of the DOXA categories*)
polarity	positive, negative, neutral, mixed
intensity	strong, average
topic	*the target of the opinion expression taken from the current domain taxonomy (a list of 1 to 5 concepts)*
justification	*reference to the paragraph that represents best the opinion expressed in the document*

Table 4.1: DOXA macro annotation of opinion.

Attribute	Value
semantic category	recommendation_suggestion, demand_query, . . . (*one or more of the DOXA categories*)
polarity	positive, negative, neutral, mixed
intensity	strong, average
topic	*the target of the opinion expression taken from the current domain taxonomy (a list of 1 to 5 concepts)*
justification	*reference to the text segment that represents best the opinion expressed in the paragraph*

Table 4.2: DOXA meso annotation of opinion.

on arbitrary size criteria since neither thematic or opinion based criteria can be used to define reliably a section,

- micro, for the sentence level annotations, with a classical definition of sentence based on syntactic and typographic analysis.

Since it was decided in DOXA that evaluation will address only the macro and meso levels, the annotations for evaluation concern only these two levels. The attributes and their values are given in tables 4.1 and 4.2. Note that both for the macro and meso levels, two categories are to be given only when the polarity has value *mixed*.

In the model, we found it important for evaluation to have a four value scale for polarity. This was done to distinguish expressions of neutrality or indifference of the source of opinion with respect to a particular target from expressions of opinion that mix positive and negative remarks, thus resulting in an intermediate polarity impression between positive and negative. When the polarity neutral is used, the semantic category and the intensity attribute are to be left unannotated.

As an annotation software, we use the Knowtator[9] plugin for Protégé by Ogren (2006) since the resulting software combination provides an annotation graphic interface coupled to an ontology browser for annotating opinion and sentiment in corpora. Knowtator facilitates the manual creation of training and evaluation corpora since it offers the possibility to annotate complex annotation schemes such as relationships between entity types. However in addition to the software, the key element of a successful annotation for creating reference data lies in the guidelines that should provide clear instructions to the annotator for choosing the appropriate markup.

9. **Knowtator**
http://knowtator.sourceforge.net/

4.3.2 Collecting annotated data

Annotation schemes depend on the task for which they are designed. Minimum annotation requirements for polarity classification is to know the polarity of a text, which can be simply specified as *positive* or *negative*. Thanks to such a simple scheme, it is quite easy to collect an annotated corpus automatically from the Web.

Many web resources such as e-commerce websites provide a functionality for their users to rate products or services (movies, hotels, restaurants, etc.) to facilitate the purchase of the reviewed entity. In many such websites, users can also leave a text comment describing their experience with the product or service. Movie fans write reviews about movies they have watched, travellers describe hotel service of where they were staying, restaurant goers give critics on restaurants. All this information can be easily collected using a simple web crawler, thus obtaining opinionated texts with their polarity value usually given as a discrete value on a fixed scale (star rating). In addition to this, it is also possible in most cases to capture the opinion target and the opinion holder. This method, however, has its own issues:

- *Rating interpretation* To separate reviews, one need to decide which reviews to consider as positive or negative given the user rating. In general, websites use star rating system with 1-5 (or 1-10) scale. In this case, researchers usually consider reviews with 1-2 stars as negative, and 4-5 as positive. It is always an issue how to interpret intermediate values (e.g. 3 stars), whether to consider them as neutral opinions, mixed or weakly positive/negative.
- *Content extraction* Collecting documents from the Web always involves the process of extracting content from HTML page as our final target is usually a raw text. It means, that we need to consider only the part with the review and disregard other elements of the page (e.g. navigation, advertising, irrelevant text, etc.). We often need to filter HTML tags, entities, fix broken character encoding.
- *Copyrights issues* The website content are often subject to copyright. Thus, before collecting the data, one need to make sure it does not violates the website's terms of use.

The main disadvantage of this approach is that the reviews are limited to the topics presented on the websites, thus we cannot collect a general purpose corpus. In our work, we use datasets automatically collected from different review websites. We cover their use in § 6.4 (p. 60) and Chapter 7 (p. 65), 8 (p. 75), and 10 (p. 97).

4.3.3 Noisy labels

Websites that provide reviews compose only a small portion of the Internet and as mentioned before they are limited by their topic. If

we need to construct a corpus of opinions in a domain that is not represented by those websites, we need to consider other methods for data collection. One of such approaches is **noisy labels**. The term **noisy labels** is used when creating annotations with crowdsourcing[10] Crowdsourcing provides annotations with a low cost, but the annotations need to be verified. In our case, we refer to noisy labels as indicators of document classes with following characteristics:

10. **crowd·sourc·ing**
the practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people and especially from the online community rather than from traditional employees or suppliers

1. *Retrieve large amount of data* Noisy labels are simple heuristic criteria that allow retrieval of a large amount of data.
2. *Not reliable* Noisy labels do not assure accurate classification, thus we need to filter out the noise and keep only informative samples.

Noisy labels are often used in bootstrapping to collect initial data. One of common approaches for constructing a sentiment analysis corpus is to start with seed words with a known polarity (e.g. *good*, *great*, *bad*, *terrible*) as noisy labels to collect a set of documents. The obtained documents are then used to obtain stronger indicators for polarity classification. Harb et al. (2008) proposed an automatic method for building a list of positive and negative adjectives for any domain. The proposed algorithm uses Google Blog search engine to collect a corpus of emotional texts. It starts with two types of seed adjectives: positive (such as *good*, *nice*, *excellent*, etc.) and negative (bad, *nasty*, *poor*, etc.) that are used as seed words to query Google Blog search engine and retrieve two sets of blog posts: positive and negative. From the retrieved sets, the authors extract adjectives and determine their polarity by looking at the occurrence of a word near seed words.

Original message posted by Scott E. Fahlman back in 1982 proposing the use of emoticons

I propose that the following character sequence for joke markers :-)

Read it sideways. Actually, it is probably more economical to mark things that are NOT jokes, given current trends. For this, use :-(

Another example of noisy labels applicable to sentiment analysis is **emoticons**. Emoticons are textual representation of the author's mood, which make a creative use of punctuations to depict facial expressions. A classical example of an emoticon is the so-called *smiley face* `:-)` which indicates a positive mood, happiness, or a joke. The use of emoticons as noisy labels was first proposed by Read (2005) for collecting data from blogs for emotion classification.

The advantage of noisy labels is that they are domain and language independent which allows collecting data for almost any application. However, to ensure the quality of the obtained corpora, one need to collect a large amount of data for filtering out the noise.

4.4 Evaluation

4.4.1 Binary classification

Measures used in evaluating polarity classifiers depend on task settings. In case of binary classification, when a system is asked to classify document polarity as positive or negative, researchers use common measures adopted in information retrieval, such as precision, recall, and accuracy.

Precision, recall, F-score, accuracy

Precision measures the ratio of correctly classified documents among the retrieved documents:

$$\mathrm{Pr} = \frac{|\text{correct documents}|}{|\text{retrieved documents}|} \tag{4.1}$$

Precision is calculated for each class, i.e. precision for positive polarity and precision for negative polarity. Thus number of retrieved documents here is equal to number of documents marked with the corresponding class:

$$\begin{aligned} \mathrm{Pr}_{pos} &= \frac{|\text{positive documents} \cap \text{marked as positive}|}{|\text{marked as positive}|} \\ \mathrm{Pr}_{neg} &= \frac{|\text{negative documents} \cap \text{marked as negative}|}{|\text{marked as negative}|} \end{aligned} \tag{4.2}$$

Here, one should not confuse terms *positive document* and *negative document* with *true positives* and *true negatives* which we cover below.

Recall measures the ratio of correctly classified documents to all documents of this class. Similarly to precision, recall is calculated for each class:

$$\begin{aligned} \mathrm{Rec}_{pos} &= \frac{|\text{positive documents} \cap \text{marked as positive}|}{|\text{positive documents}|} \\ \mathrm{Rec}_{neg} &= \frac{|\text{negative documents} \cap \text{marked as negative}|}{|\text{negative documents}|} \end{aligned} \tag{4.3}$$

These two measures are usually considered together when evaluating a classifier, because it is possible to increase one of them at a cost of reducing the other. For example, we can achieve a high precision by classifying only few samples for which we have a strong confidence. In this case, the recall drops as we are leaving out many relevant samples. Similarly, we can mark all samples as positive, thus achieving a high recall, but reducing the precision. In general, we want to obtain a balanced system with a relatively high precision and a high recall.

To ease the analysis of system performance, one may use **F-measure** which combines precision and recall in one value. One of the most used versions of F-measure is F1-score, the harmonic mean of precision and recall:

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{Pr} \cdot \mathrm{Rec}}{\mathrm{Pr} + \mathrm{Rec}} \tag{4.4}$$

Another measure that combines characteristics of precision and recall is **accuracy**. Accuracy measures the ratio of correctly classified documents to all documents:

$$\mathrm{Acc} = \frac{|\text{correct documents}|}{|\text{all documents}|} \tag{4.5}$$

Unlike precision that only penalizes a system for incorrectly classified documents, accuracy also penalizes for relevant documents that are not classified.

Contingency table

All these measures can be also interpreted using a contingency table. Table 4.3 and Table 4.4 show contingency tables for positive and negative polarities respectively. Columns represent an actual class and rows represent predicted class by a classifier.

		actual	
		pos.	neg.
predict	positive	tp	fp
	negative	fn	tn

Table 4.3: Contingency tables for positive polarity

		actual	
		pos.	neg.
predict	positive	tn	fn
	negative	fp	tp

Table 4.4: Contingency tables for negative polarity

There are four possible cases which are depicted in contingency tables: **true positives** (tp), **true negatives** (tn), **false positives** (fp), and **false negatives** (fn). These are not to be confused with polarity values. In this context, *positive* means relative documents, i.e. whose predicted class is the one we are examining at the moment. If we examine the ability of a classifier to recognize *positive polarity*, documents which polarity marked as *positive* are *positives*. Similarly, if we examine performance of a classifier to recognize *negative polarity*, in this case, documents which polarity marked as positive are *negatives* and those marked as *negative* are *positives*.

Next, we redefine evaluation measures using the contingency table.

$$\mathrm{Pr} = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp}} \tag{4.6}$$

$$\mathrm{Rec} = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fn}} \tag{4.7}$$

$$\mathrm{F1} = \frac{2\mathrm{tp}}{2\mathrm{tp} + \mathrm{fp} + \mathrm{fn}} \tag{4.8}$$

$$\mathrm{Acc} = \frac{\mathrm{tp} + \mathrm{tn}}{\mathrm{tp} + \mathrm{tn} + \mathrm{fp} + \mathrm{fn}} \tag{4.9}$$

Micro and macro averaging

Usually, we are interested in a system that identifies equally well positive and negative polarities. Thus, we need to evaluate how well it recognizes both classes. We can do it either by examining measures for each class separately or by averaging them to a single value which is usually more convenient for analysis. There are two ways to perform averaging. One way is first to compute measures for each class separately and then take the mean value. This process is called **macroaveraging**. The second way is to compute measures over a single contingency table which combines both classes. This process is called **microaveraging**. The big difference between the two averaging

processes is that macroaveraging gives equal weights to all classes, while microaveraging favors large classes. In other words, microaveraged values are closer to measure values computed for large classes.

Microaveraged measures precision, recall, and F-score are equal to accuracy:

$$\begin{aligned}
\mathrm{Acc} &= \frac{|\text{correct documents}|}{|\text{all documents}|} \text{ (by definition)} \\
&= \frac{\mathrm{TPP} + \mathrm{TNP}}{\mathrm{TPP} + \mathrm{TNP} + \mathrm{FPP} + \mathrm{FNP}}
\end{aligned}$$

$$\begin{aligned}
\mathrm{tp} &= \mathrm{TPP} + \mathrm{TNP} \\
\mathrm{tn} &= \mathrm{TPP} + \mathrm{TNP} \\
\mathrm{fp} &= \mathrm{FPP} + \mathrm{FNP} \\
\mathrm{fn} &= \mathrm{FPP} + \mathrm{FNP}
\end{aligned}$$

$$\begin{aligned}
\mathrm{Pr}_{\mathrm{mic}} &= \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp}} \\
&= \frac{\mathrm{TPP} + \mathrm{TNP}}{\mathrm{TPP} + \mathrm{TNP} + \mathrm{FPP} + \mathrm{FNP}} \\
&= \mathrm{Acc} \\
\mathrm{Rec}_{\mathrm{mic}} &= \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fn}} \\
&= \frac{\mathrm{TPP} + \mathrm{TNP}}{\mathrm{TPP} + \mathrm{TNP} + \mathrm{FPP} + \mathrm{FNP}} \\
&= \mathrm{Acc} \\
\mathrm{F1}_{\mathrm{mic}} &= 2 \cdot \frac{\mathrm{Pr}_{\mathrm{mic}} \cdot \mathrm{Rec}_{\mathrm{mic}}}{\mathrm{Pr}_{\mathrm{mic}} + \mathrm{Rec}_{\mathrm{mic}}} \\
&= 2 \cdot \frac{\mathrm{Acc} \cdot \mathrm{Acc}}{\mathrm{Acc} + \mathrm{Acc}} \\
&= \mathrm{Acc}
\end{aligned} \tag{4.10}$$

where:

TPP	true positive polarity
TNP	true negative polarity
FPP	false positive polarity
FNP	false negative polarity

If distributions of positive and negative polarities in training and test data are not balanced, e.g. positive documents prevail, a classifier will mark more documents as positive. Since, the test data also contain more positive documents, precision for the positive class will be much higher than that of the negative class. In this case, we should use macroaveraged measures to distinguish systems that separate better both classes rather than simply assign the major class. In general, macroaveraged precision is a sufficient measure because

positive and negative classes are complementary.

$$\mathrm{Pr}_{\mathrm{mac}} = \frac{\mathrm{Pr}_{\mathrm{pos}} + \mathrm{Pr}_{\mathrm{neg}}}{2} \tag{4.11}$$

4.4.2 *Beyond binary classification*

Although most of research on polarity classification is focused on binary setting, there is a need to move beyond to be able to distinguish more polarity values. This can be done by adding more classes (e.g. *neutral*, *mixed*) or by representing polarity as a continuous value on a fixed scale[11]. In case of discrete values, we can use similar measures as for binary classification: accuracy, micro/macroaveraged precision, recall, F-score. As for the continuous case, a common choice is **mean squared error** (MSE):

11. For example, we can represent polarity as a value between 1 and 10, where 1 indicates strongly negative sentiments and 10 indicates strongly positive sentiments. This is similar to star rating system popular on review websites described in § 4.3.2 (p. 30).

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (X_i^{\mathrm{true}} - X_i^{\mathrm{predict}})^2 \tag{4.12}$$

where:

n	total number of documents
X_i^{true}	true polarity value of i-th document
X_i^{predict}	predicted polarity value by the classifier

APPROACHES TO POLARITY CLASSIFICATION

5

The existing approaches to polarity classification fall into two large categories:

1. lexicon based methods
2. statistical based methods

A lexicon based approach uses some sort of an affective lexicon to derive the polarity of the examined text. A statistical based approach uses annotated texts with a given polarity to learn a statistical model. The first approach is limited by the dictionary size and requires human expertise to build the lexicons, the other approach usually produces less accurate results, but as we add more training data the accuracy improves. Of course, the two approaches can be combined to achieve even a better performance.

5.1 Lexicon based approaches

Lexicon based approaches range from simple methods that look up words in lists of positive and negative terms (Messina et al., 1989, Syssau and Font, 2005, Buvet et al., 2005) to much more complex solutions that use semantic networks of concepts describing human emotions, real world objects, and possible relations between them. In this section, we will review lexical resources for sentiment analysis starting from the most simple ones.

5.1.1 Affective lexicons

Affective lexicons contain lists of words either divided by certain sentiment classes (e.g. positive/negative) or providing a single list of words each associated with a numerical value representing its polarity. Below are descriptions of most used affective lexicons in sentiment analysis research.

ANEW

Affective Norms of English Words (ANEW) is a set of normative emotional ratings for 1034 English words developed by Bradley and Lang (1999) from the NIMH Center for Emotion and Attention

The **Center for the Study of Emotion & Attention** is devoted to studying the behavior and physiology of human emotion, highlighting emotion's motivational significance for both attention and response mobilization.
http://csea.phhp.ufl.edu/

(CSEA) at the University of Florida. Localized versions of ANEW exist in other languages, such as Spanish by Redondo et al. (2007) and German by Vō et al. (2009). For each word in the dataset, there are scores for three dimensions of emotional assessment: valence (ranging from pleasant to unpleasant), arousal (ranging from calm to excited) and dominance (ranging from in-control to dominated). This dataset is a useful tool for emotion studies as well as for opinion mining and sentiment analysis. In our research, we use ANEW to validate a constructed affective lexicon (Chapter 6, p. 51) and to generate additional features to be used for emotion detection (Chapter 11, p. 107).

General Inquirer

General Inquirer (GI) originally a text analysis system created at IBM, but later decomposed into the software part and the dictionary containing 11789 senses of 8641 English words (i.e. certain words have several senses), each mapped to one or more of 182 categories, such as *positive*, *negative*, *self*, *family*, etc. One of the earliest applications of GI was automatic analysis of suicide notes proposed by Stone and Hunt (1963). Concerning the application to sentiment analysis, GI contains a *positive* category with 1915 assigned to it words and a *negative* category with 2291 words that can be used for polarity classification or subjectivity analysis. We use General Inquirer to generate features for emotion detection (Chapter 11, p. 107).

5.1.2 WordNet and graph based methods

WordNet is one of the largest lexical resource for English language which is extensively used in scientific research. According to the description from the project homepage[1],

1. WordNet project homepage: http://wordnet.princeton.edu/

WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and concepts can be navigated with the browser. WordNet is also freely and publicly available for download. WordNet's structure makes it a useful tool for computational linguistics and natural language processing.

Figure 5.1 shows a graph representation of WordNet synsets and relations between them. Many researchers use WordNet for sentiment analysis as well as other resources based on it, such as WordNet Affect and SentiWordNet. It has been also localized in other languages (Vossen, 1998, Navigli and Ponzetto, 2010, Vetulani et al., 2009, Fišer and Sagot, 2008).

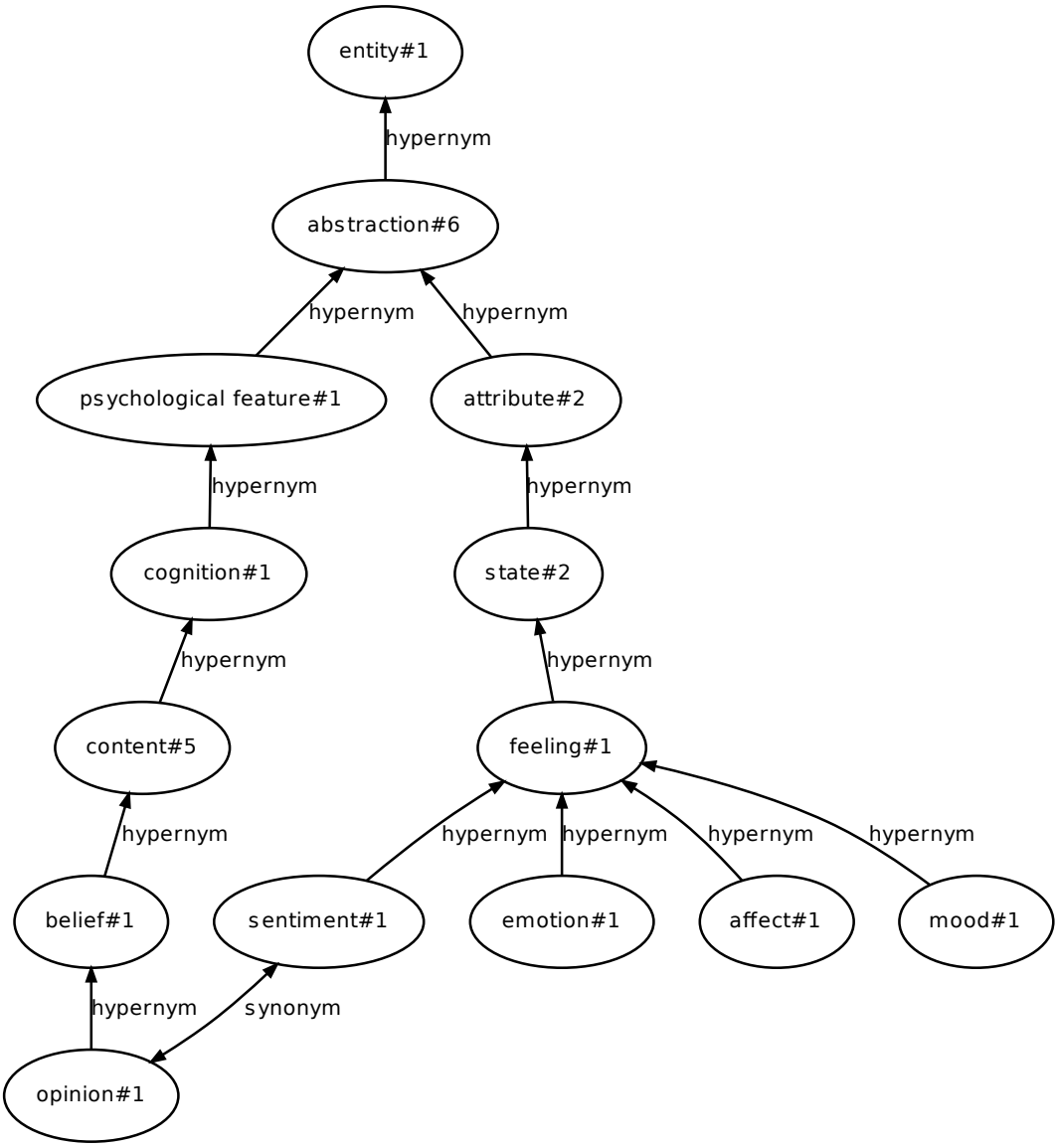


Figure 5.1: An example of a graph visualization of WordNet. Nodes represent synsets, edges represent relations between synsets.

Category	Example label
emotion	anger
cognitive state	doubt
trait	competitive
behavior	cry
attitude	skeptisicm
feeling	pleasure

Table 5.1: WordNet Affect categories of labels with examples

WordNet Affect

WordNet Affect is a manually created extension of the WordNet by Strapparava and Valitutti (2004). It is part of the WordNet Domains project[2] which assigns topical labels to WordNet synsets such as *sports*, *politics*, *medicine*. WordNet Affect assigns affective labels which in their turn are organized into categories presented in Table 5.1.

2. WordNet Domains project: http://wndomains.fbk.eu/

WordNet Affect has been used for different tasks in sentiment analysis, such as irony identification (Reyes and Rosso, 2011, González-Ibáñez et al., 2011) and polarity classification (Chaumartin, 2007). The usefulness of the resource leaves no doubt. However, since it has been constructed manually, with the update of WordNet the affective labels become outdated.

SentiWordNet

SentiWordNet is a freely available[3] lexical resource for sentiment analysis developed by Baccianella et al. (2010). It was constructed by automatic annotation of WordNet synsets with three numeric scores representing positiveness (*Pos*), negativeness (*Neg*), and objectivity (*Obj*) by taking the advantage of graph-based model of the WordNet. The annotation has been performed in two steps:

3. SentiWordNet is available at http://sentiwordnet.isti.cnr.it/

1. In the first step, all the synsets from the WordNet were classified as positive, negative, or objective (neutral) using a supervised classifier (Rocchio and SVM). To train a classifier, the authors used boosting to collect a training set. Boosting starts by defining a small set of seed words: paradigmatically positive and paradigmatically negative ones. Next, seed words neighbours are considered as having a similar polarity and thus are added to the initial sets of positive/negative words. The training set is composed of glosses (word definition and optional sentences of examples of use) of expanded sets of positive/negative words and a set of words considered as objective.
2. In the second step, random walk method is used to estimate *Pos*, *Neg*, and *Obj* values for each WordNet synset. Random walk is performed over the WordNet graph, where nodes represent

synsets and edges represent whether given two synsets are used in either one's definition. Thus, a synset's polarity would be estimated by the majority of other synsets that define it.

SentiWordNet as well as WordNet Affect has been used in a number of research (Chaumartin, 2007), however, its advantage is automatic construction which makes it potentially adaptive to other languages (if that language has its own version of WordNet).

5.1.3 Automatic construction of lexical resources

There are two ways to cover the lack of sentiment analysis resources. The first way is to localize a lexicon. Redondo et al. (2007) have adapted ANEW into Spanish, Vō et al. (2009) localized it into German. This approach requires human translators to ensure the quality of the localized resource and therefore is cost expensive and not scalable.

The second approach is automatic construction of a lexicon (Vernier and Monceau, 2010, Jackiewicz, 2010). The most common method is bootstrapping. This method starts with seed words with a known polarity (e.g. *good*, *happy*, *wonderful* for a positive class, *bad*, *sad*, *terrible* for a negative class). Next, the seed words are used to find related words and assign them the same class or estimate their polarity. Turney and Littman (2003) used pointwise mutual information and latent semantic analysis on the TOEFL[4] dataset. Takamura et al. (2005) used spin model and mean field estimation with WordNet on the Brown corpus. Lu et al. (2011) used travel recommendations and product reviews extracted from different sources along with WordNet to construct a context-aware sentiment lexicon.

4. **TOEFL** stands for test of English as a foreign language

Mathieu (2006) constructed a computational semantic lexicon of French verbs of feeling, emotion, and psychological states. The lexicon contains 600 verbs which are divided into 33 semantic classes. All classes are grouped into 3 categories: positive, negative, and neutral. The lexicon also contains links between the classes. Those links are meaning, intensity, and antonymy. The paper also presents the FEELING software that uses the built lexicon for interpretation of emotions within given phrases. Although the presented lexicon seems to be a valuable tool for sentiment analysis, it would require a lot of human resources to extend it. Such as for annotating adjectives or other verbs.

5.2 Statistical based approaches

Machine learning
a branch of artificial intelligence, is a scientific discipline concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data, such as from sensor data or databases.

A common approach is to train a supervised machine learning classifier over a set of given samples. Statistical based approaches to polarity classification originate from the time when researchers first applied techniques borrowed from traditional information retrieval problems such as topic detection or genre classification. Thus, positive and negative polarity was seen as two different topics or genres into which a text should be classified.

An early work by Pang et al. (2002) on polarity classification using bag-of-words model and machine learning reported 82.7% accuracy. This approach was tested on a movie review dataset collected from IMDb which was later used by other researchers thus producing comparable results (Whitelaw et al., 2005, Matsumoto et al., 2005). The authors reported that a simple setup using unigram features with binary weights yielded the highest accuracy.

5.2.1 Classifiers

A choice of a classification algorithm in general is not so important. Some machine learning algorithms perform better on a particular dataset but worse on others. No evidence has been provided so far that a particular classifier performs significantly better in opinion mining than others. A usual approach is to test different classifiers when conducting experiments. Otherwise, researchers use the one which they are most familiar with. What follows is an overview of some of most used classifiers for sentiment analysis and particularly for polarity classification.

Naïve Bayesean classifier

Naïve Baysean (NB) classifiers are considered to be one of the simplest among other supervised learners. They are rather easy to implement and computationally inexpensive. Yet, they provide quite good results (Rish, 2001) and often outperform more complex learners. Another advantage of NB is its ability to classify into more than two classes, while other learners require more tuning when applied to a multiclass problem. NB classifiers are based on the Bayes' theorem:

Bayes theorem

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)} \quad (5.1)$$

given that A, B – events, $P(B) \neq 0$

$$P(C_i|F_1 \dots F_n) = \frac{P(C_i)P(F_i \dots F_n|C_i)}{P(F_i \dots F_n)} \quad (5.2)$$

where:

C_i	class
$\{F_n\}$	set of features

This equation is usually simplified by assuming conditional independence of features and using a log-likelihood form:

$$L(C_i|F_1 \dots F_n) = L(C_i) + \sum_{j=1}^{n} L(F_j|C_i) \quad (5.3)$$

where $L(x) = log(P(x))$ The classifier thus selects a class with the maximum log-likelihood by estimating the probabilities $P(C_i)$ and $P(F_j|C_i)$:

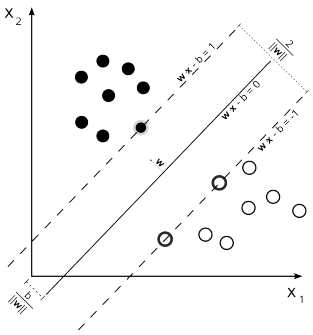
$$\begin{aligned} C &= \underset{i}{\operatorname{argmax}}\, L(C_i|F_1 \dots F_n) \\ &= \underset{i}{\operatorname{argmax}}\, L(C_i) + \sum_{j=1}^{n} L(F_j|C_i) \end{aligned} \tag{5.4}$$

We use NB classifier for polarity classification in Chapter 7 (p. 65), 9 (p. 91).

Support vector machines

Support vector machines (SVM) use an optimization technique called **quadratic programming** to divide samples of two classes as best as possible. SVM classifier represents samples as points in a multidimensional space and then looks for a hyperplane that separates points of one class from the points of another class. While there exists an infinite number of such hyperplanes, an SVM chooses the one with the largest margins between the hyperplane and the points representing a class. These points are called **support vectors**. SVM can be applied also to a multiclass problem as well as a regression task. We actively use SVM for polarity classification in Chapter 7 (p. 65), 8 (p. 75), 10 (p. 97), 11 (p. 107).

Quadratic programming is the problem of optimizing (minimizing or maximizing) a quadratic function of several variables subject to linear constraints on these variables.



5.2.2 *Feature construction*

The choice of features directly affects the quality of the trained model and thus the system performance. There are at least two important choices to make:

- which features to use for document representation
- how to weight the features

Bag-of-words

The most commonly used model in text analysis is the **bag-of-words** (BOW) model. The bag-of-words model represents a text as an unordered set of words composing the text. When using this model, we assume that words are independent from each other and also disregard the order of words. While these assumptions are very naïve, BOW produces good results in many tasks of computational linguistics and information retrieval.

An extension of this model is the **n-gram** model which uses subsequences of words of a size n instead of single words. Thus, a **unigram** (or 1-gram) model corresponds to a bag of words, a **bigram** (or 2-gram) model takes two consecutive words to form an n-gram, and a **trigram** (or 3-gram) model takes three consecutive words. Higher

order n-grams are rarely used due to lack of data for probability estimation. In practice, opinion mining researchers use either unigrams or bigrams (Pang et al., 2002).

Below is an example of a text (S = "The soundtrack was awful") being represented using a unigram and a bigram models.

$$\begin{aligned}\text{unigrams}(S) = \{&(\text{The}),\\ &(\text{soundtrack}),\\ &(\text{was}),\\ &(\text{awful})\}\end{aligned}$$

$$\begin{aligned}\text{bigrams}(S) = \{&(\text{The}, \text{soundtrack}),\\ &(\text{soundtrack}, \text{was}),\\ &(\text{was}, \text{awful})\}\end{aligned}$$

Alternative models are rarely used. In this work, we propose an alternative model, which is based on dependency parse trees. A detailed description of the model is presented in Chapter 7 (p. 65).

The bag-of-words model is often used as a base for a sentiment analysis system (we actively use it throughout this work: Chapter 6, p. 51; 7, p. 65; 8, p. 75; 10, p. 97; 11, p. 107) with other features to be added on top of it to achieve a better accuracy. These features include lexical (part-of-speech tags, negations, punctuations), semantic (word semantic categories, emotion categories, word polarity scores), discourse (discourse relations), positional and others. We report the usefulness of POS tags for sentiment analysis in § 6.2.2 (p. 55). Dependency parsing has been also widely used for extracting additional features (Arora et al., 2010, Nakagawa et al., 2010). A recent work by Zirn et al. (2011), used discourse parsing to take into account relation between phrases for fine-grained polarity classification.

Weights

Another important aspect of feature construction is how to weight the features. If we use bag-of-words model or n-grams, clearly some terms are more important for polarity classification than others. Traditional information retrieval make an extensive use of the **tf-idf** (term frequency–inversed document frequency) weighting scheme. However, as it has been shown by Pang et al. (2002), tf-idf scheme is not efficient for sentiment analysis and more particularly **binary** scheme outperforms term frequency. The authors explain this as:

Binary scheme
captures if a word is present in a text by assigning value 1 if a words is in the text and 0 otherwise.

We speculate that this indicates a difference between sentiment and topic categorization – perhaps due to topic being conveyed mostly by particular content words that tend to be repeated – but this remains to be verified.

Martineau and Finin (2009) proposed **delta tf-idf** scheme which computes the difference of a word's tf-idf score in a positive and

a negative training sets. They claimed that the proposed technique boosts the importance of words unevenly distributed between the positive and the negative classes, thus these words should contribute more in the classification. Evaluation experiments on three different datasets showed statistically significant improvement of classification accuracy over binary weighting. The efficiency of delta tf-idf has been confirmed in experiments by Paltoglou and Thelwall (2010) in which the authors performed a thorough study on different weighting schemes and their impact on polarity classification accuracy.

In Chapter 8 (p. 75), we perform a study on the impact of feature weighting on polarity classification. We emphasize the problem of entity-specific features and propose three weighting schemes to improve classification accuracy of reviews with polarities different from the majority.

PART II

AUTOMATION AND ADAPTIVITY

Researchers usually focus on a certain domain, such as movie or product reviews, and often use specific language resources, such as affective lexicons or specific taxonomies. Once built for a certain task, sentiment analysis systems are difficult to adapt to other domains or languages without a significant performance loss. Building a universal sentiment analysis system which can handle any domain in any language is comparable to solving natural language understanding and well out of reach of current technology.

The goal of our work is adaptive sentiment analysis. We do not aim at building a universal system, but rather develop a set of adaptive methods which produce resources required for creating a sentiment analysis system for a given task in any domain and any language. This means that we cannot rely on lexical resources and tools that are available only for certain languages, but we have to use common tools and text properties that are constant across languages and domains.

Particularly, we use opinionated texts from Twitter as an example of a multilingual cross-domain lexical resource. We propose our method for text indexing based on dependency parse trees, which performs better than n-grams models according to our experiments. Finally, we use statistical properties of corpora to improve features weights.

To show how easily our approach can be adapted, we tested it on several distinct domains: movies, video games, products, microblogging, medical reports. In our experiments, we worked with English, French, Russian, Spanish, and Chinese. This part presents the main contribution of our work which lies in three fields:

1. Automatic construction of affective resources
2. Feature construction for machine learning classifiers
3. Improving feature weighing schemes

AUTOMATIC LEXICON CONSTRUCTION FROM MICROBLOGS

6

As we have mentioned in § 4.3 (p. 27), both lexical based and machine based approaches need annotated data for training and evaluating the model. In § 5.1.3 (p. 41), we have reviewed some existing methods for automatic construction of affective lexicons. Most of those methods are limited in building resources for certain domains and in certain languages. In this chapter, we present our method for building multilingual resources for sentiment analysis.

In order to construct such a resource, we use the noisy labels approach (§ 4.3.3, p. 30) which is domain and language independent. To collect data, we use Twitter, a popular microblogging platform that meets our basic requirements: it contains a vast amount of text comments many of which are opinionated. Twitter audience consists of users from different countries who speak different languages and have different interests what makes it a multilingual and multidomain data source.

6.1 Microblogging

As opposed to traditional blogging platforms, **microblogging**, a recent trend in the Internet, limits the length of blog posts allowing to submit only short texts. Such a restriction that seems to constrain self expression, in reality turned out to be a motivating factor for writing messages. The authors were not bound anymore by a formal style of writing and as a result began to make more posts though much shorter ones than usually.

blog
a web site that contains an online personal journal with reflections, comments, and often hyperlinks provided by the writer
Source: Merriam-Webster

One of the reasons of the message size limit was due to the ability to post messages by sending it via SMS[1]. Because of the simplicity of message posting and informal writing style, many users tend to use microblogging as the main electronic communication tool instead of email and instant messaging. Nowadays, microblogs along with personal use became a communication channel between companies and consumers, celebrities and fans, politicians and regular citizens.

1. SMS stands for Short Message Service

As the audience of microblogging platforms rapidly grows everyday, data from these sources can be used for opinion mining and sentiment analysis. Jansen et al. (2009) called microblogging "*an online word-of-mouth branding*" highlighting the fact that microblogging is used to exchange opinions about product brands. Since the length

word of mouth
the passing of information from person to person by oral communication

funkeybrewster: @redeyechicago **I think Obama's visit might've sealed the victory for Chicago**. Hopefully the **games mean good things** for the city.
vcurve: **I like how Google celebrates** little things like this: Google.co.jp honors Confucius' Birthday | Japan Probe
mattfellows: Hai world. **I hate faulty hardware** on remote systems where politics prevents you from moving software to less faulty systems.
brrooklyn: **I love the sound my iPod** makes when I shake to shuffle it. Boo bee boo

Table 6.1: Examples of Twitter opinionated posts. Sentiment expressions highlighted in bold for reference.

of a message is limited, users are forced to express opinions more explicitly what makes it easier to analyse. For example, product manufacturing companies can retrieve the following information:

- What people think about their product or brand
- How positive (or negative) are people about the product
- What people prefer the product to be like

Other organizations may as well profit from microblogging analysis. Political parties may be interested to know whether people support their program. Social organizations may ask people's opinion on current debates.

Figure 6.1: Twitter logo

In our research, we address **Twitter**, the most popular microblogging platform nowadays. In Twitter, users exchange short text messages called *tweets* which cannot exceed 140 characters. Table 6.1 gives examples of opinionated tweets. A typical tweet consists of a text and meta data. Meta data contains author information (name, location, bio, language, etc.) and tweet information (created time, number of replies, etc.). A text often includes URLs[2], usernames of other authors, hashtags (a special way to label entities, see § 3.5, p. 15), emoticons, a *retweet* sign.

2. **Universal Resource Locator** a specific character string that constitutes a reference to an Internet resource

Figure 6.2 shows an example of a *retweet*. Retweets are replies to other tweets. They can be distinguished by the retweet sign (*RT*) in the text which indicates that a following text is from another message. Hashtags and usernames can also be distinguished: hashtags follow a *sharp sign* (#) and usernames follow *commercial at* (@) sign.

While there are many debates whether the content of Twitter is valuable or it is just a huge collection of daily rambling and spam, we believe that in spite of much noise, it is a useful data source for sentiment analysis and opinion mining for the following reasons:

- Twitter is used by a broad audience to express their point of view about different topics, thus it is a valuable source of people's opin-

Figure 6.2: Anatomy of a tweet

ions and sentiments.

- It contains an enormous number of text posts and it grows every day. The collected corpus can be arbitrarily large.
- The audience varies from regular users to celebrities, company representatives, politicians, and even country presidents. Therefore, it is possible to collect text posts of users from different social and interest groups.
- Twitter audience is represented by users from many countries. Although users from U.S. are prevailing (see Table 6.2), it is possible to collect data in different languages. In our research, we collected data in English, Spanish, French, and Chinese.

Rank	Country	Users
1	U.S.	107.7 M
2	Brazil	33.3 M
3	Japan	29.9 M
4	U.K.	23.8 M
5	Indonesia	19.5 M
	...	
16	France	5.2 M

Table 6.2: Top countries by registered users in Twitter as of Jan 2012. *Source:* http://semiocast.com/

Despite a recent change in Twitter policy on data usage and API rate limit, Twitter is still widely used in academic research.

[...] Twitter's recent announcement that it was no longer granting whitelisting requests and that it would no longer allow redistribution of content will have huge consequences on scholars' ability to conduct their research, as they will no longer have the ability to collect or export datasets for analysis.
– Source: http://www.readwriteweb.com/

We use Twitter as a multilingual resource to collect a dataset for sentiment analysis. In this chapter, we show how to obtain a labeled dataset with sentiment labels in an automatic way using emoticons as noisy sentiment labels. We use the obtained dataset for three kinds of tasks:

1. construction of affective lexicons for different languages
2. polarity classification of video game reviews in French
3. disambiguation of sentiment ambiguous adjectives in Chinese (Chapter 9, p. 91)

Language	Negative	Positive	Neutral	Total
English	100.0	100.0	100.0	300.0
English2	2,616.9	2,616.9	-	5,233.9
Spanish	336.4	336.4	-	672.8
French	143.7	143.7	-	287.4
Chinese	3.9	3.9	-	7.8

Table 6.4: Number of collected Twitter messages for different language versions of the sentiment corpus (in thousands)

6.2 Corpus collection and analysis

6.2.1 Corpus collection

Data collection from the Web usually involves crawling and parsing of HTML pages which is a solvable but at the same time a consuming task. In our case, collecting data from Twitter is much easier since it provides an easy and well-documented API[3] to access its content. We use the API to retrieve positive, negative, and neutral messages in the following languages: English, French, Spanish, and Chinese. Datasets in different languages were collected separately and they were used for different tasks. English dataset was used in most of our analysis because it is the major language of the Twitter audience.

3. Twitter API
https://dev.twitter.com/docs

To collect sets of positive and negative messages, we use emoticons as noisy sentiment labels. Because each message cannot exceed 140 characters it is usually composed of a single sentence, therefore we assume that an emoticon within a message represents an emotion for the whole message and all the words of the message are related to this emotion. We use emoticons expressing happy emotions, such as `:-)`, to collect the positive set and sad emoticons, such as `:-(`, for the negative set. We ignored messages that contain both types of emoticons since their interpretation is more ambiguous.

For English, we additionally collected a set of neutral messages, for which we had composed a list of newspapers that post news titles in their Twitter accounts (such as "New York Times", "Washington Posts" etc.) as we assumed that newspapers try to post objective information. The final list contains 44 newspapers that we used to collect tweets for the neutral dataset.

Characters per message	70.14
Characters per word	6.84
Words per message	10.26

Table 6.3: Characteristics of collected English tweets

Table 6.4 summarizes the size of the collected dataset across the languages. For English, we have collected two different sets, the first one containing 3 classes (positive, negative, neutral) of 100,000 messages each and was used to perform a linguistic analysis of tweets which is presented in the next section, the second one composed of two classes (positive, negative) with a total of more than 5 million tweets was used to create an affective lexicon.

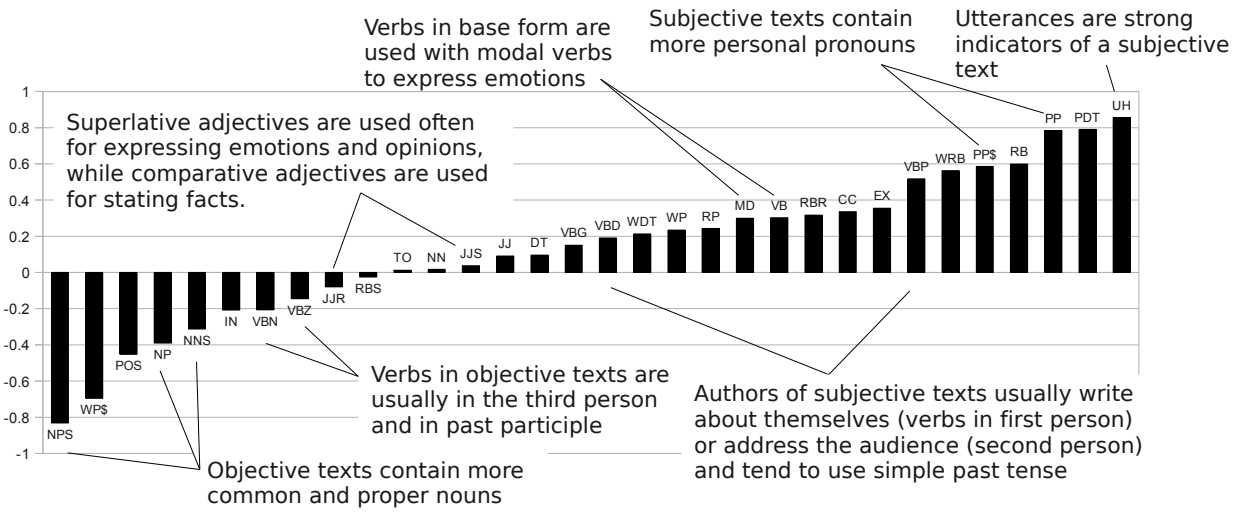


Figure 6.3: $p(t)$ values for objective vs. subjective

6.2.2 *Corpus analysis*

An average tweet in English is about 70 characters long, contains 10 words each of 6–7 letters (see Table 6.3). To perform a linguistic analysis, we tag all the messages in the corpus with the TreeTagger by Schmid (1994). We are interested in difference of part-of-speech distributions between subjective (positive and negative) and objective (neutral) texts, and between positive and negative texts. To perform a pairwise comparison of tags distributions, we calculate the following value for each tag and two given sets:

$$p(t) = \frac{N(t, M_1) - N(t, M_2)}{N(t, M_1) + N(t, M_2)} \tag{6.1}$$

where $N(t, M_1)$ and $N(t, M_2)$ are numbers of tag t occurrences in the first and second sets M_1 and M_2 respectively. The value of $p(t)$ characterizes the following cases:

- if $p(t)$ is positive and close to 1, then the set 1 contains much more occurrences of the tag than the set 2
- if $p(t)$ is negative and close to -1, then the set 1 contains much less occurrences of the tag than the set 2
- if $p(t)$ is close to 0, then two sets contain approximately the same number of occurrences of the tag

Subjective vs. objective

Figure 6.3 shows the values of $p(t)$ across all the tags where set 1 is a subjective set (mixture of the positive and the negative sets) and set 2 is an objective set (the neutral set). From the graph we can observe that POS tags are not distributed evenly in two sets, and therefore can be used as indicators of a set. For example, utterances (UH) can be a strong indicator of a subjective text.

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
IN	Preposition/subord. conj.
JJ	Adjective
JJR	“, comparative
JJS	“, superlative
MD	Modal
NN	Noun, singular or mass
NNS	“, plural
NP	Proper noun, singular
NPS	“, plural
PDT	Predeterminer
POS	Possessive ending
PP	Personal pronoun
PP$	Possessive pronoun
RB	Adverb
RBR	“, comparative
RBS	“, superlative
RP	Particle
TO	to
UH	Interjection
VB	Verb, base form
VBD	“, past tense
VBG	“, gerund/pres. particip.
VBN	“, past participle
VBP	“, non-3rd p. sing. present
VBZ	“, 3rd p. singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possessive wh-pronoun
WRB	Wh-adverb

Table 6.5: TreeTagger tagset

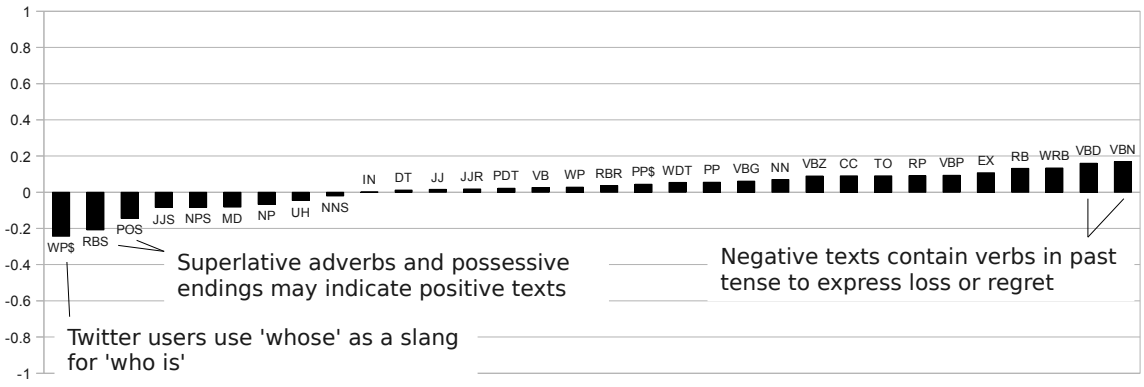


Figure 6.4: $p(t)$ values for positive vs. negative

We observe that objective texts tend to contain more common and proper nouns (NPS, NP, NNS), while authors of subjective texts use more often personal pronouns (PP, PP$). Authors of subjective texts usually describe themselves (first person) or address the audience (second person) (VBP), while verbs in objective texts are usually in the third person (VBZ). As for the tense, subjective texts tend to use simple past tense (VBD) instead of the past participle (VBN). Also a base form of verbs (VB) is used often in subjective texts, which is explained by the frequent use of modal verbs (MD). In the graph, we see that superlative adjectives (JJS) are used more often for expressing emotions and opinions, and comparative adjectives (JJR) are used for stating facts and providing information. Adverbs (RB) are mostly used in subjective texts to give an emotional color to a verb.

Positive vs. negative

Figure 6.4 shows values of $p(t)$ for negative and positive sets. As we see from the graph, a positive set has a prevailing number of possessive wh-pronoun *whose* (WH$), which is unexpected. However, if we look in the corpus, we discover that Twitter users tend to use *whose* as a slang version of *who is*:

dinner & jack o'lantern spectacular tonight! :) whose ready for some pumpkins??

Another indicator of a positive text is superlative adverbs (RBS), such as *most* and *best*. Positive texts are also characterized by the use of possessive ending (POS). As opposite to the positive set, the negative set contains more often verbs in the past tense (VBN, VBD), because many authors express their negative sentiments about their loss or disappointment. Here is an example of the most frequent

verbs: *missed*, *bored*, *gone*, *lost*, *stuck*, *taken*. We have compared distributions of POS-tags in two parts of the same sets (i.e. a half of the positive set with another half of the positive set). The proximity of the obtained distributions allows us to conclude on the homogeneity of the corpus.

6.3 Lexicon construction from Twitter

As we have described in § 5.1.1 (p. 37), affective lexicons are lexical resources that are used in sentiment analysis. Once created for a specific language, it is a non trivial task to adapt it to another language. We aim at automatic construction of affective lexicons using Twitter since it can be seen as a multilingual data source. Below, we describe our approach to estimate polarity score for words in Twitter corpus. To evaluate the constructed lexicon, we test the correlation with ANEW and its adapted versions in Spanish and French[4].

4. French adaptation has been done by ourselves by applying MT with manual correction

Our basic assumption is that a word should have a high valence score if it appears frequently in the positive set and at the same time rarely in the negative set. Contrary, a word that appears more often in the negative set rather than in the positive one, should have a low valence score. For each word w in the corpus, given the counts of its occurrences in the positive set $N(w, M_{pos})$ and in the negative set $N(w, M_{neg})$, we estimate the value of polarity of a word on a scale[5] of 1–9 as follows:

5. The 9-scale was chosen to compare the estimated values with ANEW scores

$$\begin{aligned} \text{polarity}^*(w) &= \frac{9 \cdot N(w, M_{pos}) + N(w, M_{neg})}{N(w, M_{pos}) + N(w, M_{neg})} \\ \text{polarity}^*(w) &= 8 \cdot \frac{N(w, M_{pos})}{N(w, M_{pos}) + N(w, M_{neg})} + 1 \\ &= 8 \cdot P(M_{pos}|w) + 1 \end{aligned} \tag{6.2}$$

where

$P(M_{pos}\|w)$	probability of a word w to be positive
8	scale factor
1	offset

To validate our estimation, we calculate the mean squared error:

$$MSE = \frac{1}{|W|} \sum_{w \in W} (polarity(w) - polarity^*(w))^2 \tag{6.3}$$

where

W	list of all the words
$\text{polarity}(w)$	polarity score from ANEW

Another way to validate the estimated values is to check the correlation of two words rankings: the ranking according to the values of polarity from the ANEW and the ranking according to the estimated values of polarity. Two scores were calculated to measure the

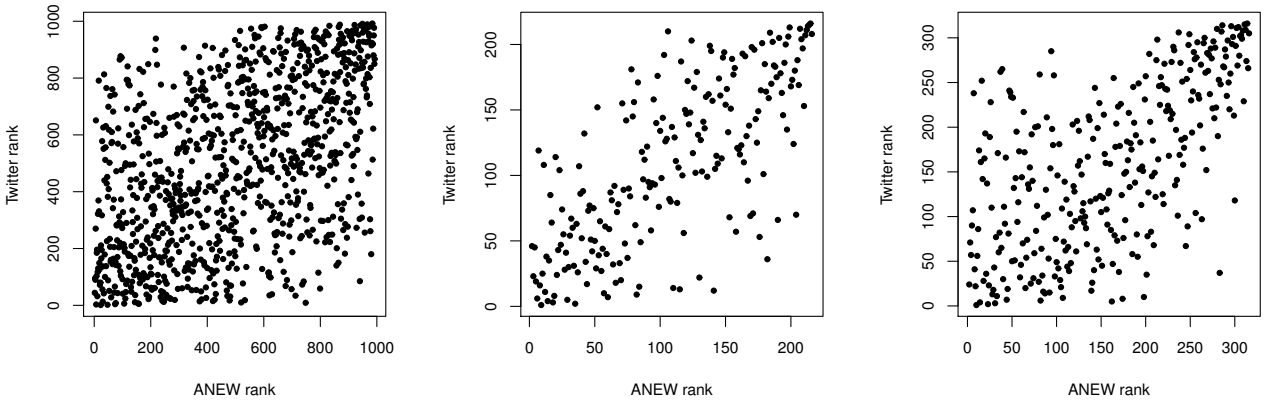


Figure 6.5: Correlation between ANEW and the constructed affective lexicon: all words (left), only adjectives (center), filtered words (right).

Kendall's tau:

$$\tau = \frac{n_c - n_d}{\frac{1}{n}(n-1)} \tag{6.4}$$

where

n_c	number of concordant pairs
n_d	number of discordant pairs
n	total number of elements

Spearman's coefficient:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{6.5}$$

where

d_i	difference between positions of an element in two rankings
n	total number of elements

correlation Kendall's tau coefficient (Kendall, 1938) and Spearman's rank coefficient (Maritz, 1981).

Both coefficients take a value between -1 and 1, where:

- -1 indicates the perfect disagreement between two rankings
- 1 indicates the perfect agreement between two rankings

We experimented with three different lists:

- All words – all the words from the ANEW
- Adjectives – only adjectives from the ANEW list
- Frequent words – words from the ANEW list that occur at least 100 times in our collected dataset

We have visualized the rankings for English in Figure 6.5, where each dot represents a word, X-axis corresponds to a ranking according to ANEW polarity score, Y-axis corresponds to a ranking according to estimated polarity score. Numerical values are presented in Table 6.6 for English, Spanish, and French.

Kendall's tau and Spearman's coefficient show a good agreement of the two rankings. We can observe quite a large value of the mean squared error. The latter can be diminished by using machine learning techniques or regression analysis. We consider that correlation coefficients are more appropriate evaluation measures for our purpose. Kendall's tau and Spearman's coefficient show a good correlation for adjectives and frequent words in English, Spanish, and French. Hence, we can construct ANEW-like lexicon by selecting unigrams with frequent appearance in our corpus. Table 6.7 shows examples from the English lexicon.

	Score	Kendall's tau	Spearman's coef.	Mean Squared Err.
English	All words	0.359	0.510	3.055
	Adjectives	0.550	0.736	2.046
	Frequent words	0.454	0.626	2.861
Spanish	All words	0.327	0.468	5.068
	Adjectives	0.434	0.610	4.251
	Frequent words	0.441	0.626	3.287
French	All words	0.287	0.376	6.184
	Adjectives	0.384	0.489	4.836
	Frequent words	0.448	0.543	3.174

Table 6.6: Correlation coefficients and mean squared error for English, Spanish, and French

Word	Estimated polarity	Word	Estimated polarity
congratulations	8.460	depressed	1.480
welcome	8.388	ache	1.493
blessed	8.333	sadly	1.551
pleasure	8.015	hurts	1.558
thank	8.013	coughing	1.558
smile	7.976	depressing	1.602
appreciated	7.974	poorly	1.640
congrats	7.973	lonely	1.654
sharing	7.957	poor	1.655
smiling	7.937	cancelled	1.667
thanks	7.909	upset	1.674
appreciate	7.880	hates	1.680
proud	7.863	headache	1.708
cheers	7.767	disappointed	1.714
fabulous	7.684	fever	1.718
hilarious	7.631	earthquake	1.727
adorable	7.626	infection	1.744
excellent	7.603	ruined	1.744
rocks	7.561	miserable	1.752
wonderful	7.522	died	1.764

Table 6.7: Example of the obtained word list with high (left) and low (right) estimated polarity values.

6.4 Polarity classification

To test whether the constructed affective lexicon is useful for a real application, we applied it to polarity classification of French video game reviews. We could not find any existing affective lexicons in French[6], hence it is a good proof of usefulness of the constructed lexicon in the case when other resources for sentiment analysis are not available.

6. There exist several affective lexicons constructed for French by Messina et al. (1989) and Syssau and Font (2005), though not publicly available.

6.4.1 Data

We use the video game review dataset from the DOXA project, which aims at building a domain independent industrial platform for opinion mining.

In the DOXA annotation, the sentiment polarity is expressed by means of a six-value scale: *neutral*, *strong-negative*, *weak-negative*, *mixed*, *weak-positive*, *strong-positive*. We take all the documents with positive polarity (*strong positive* and *weak positive*) all the documents with negative polarity (*strong negative* and *weak negative*) and assign them to a positive and a negative classes respectively. We do not use documents marked as neutral (no sentiment expressed) or mixed (both positive and negative sentiment expressed together). This way, we obtain 387 positive documents and 250 negative documents from the annotated set. We further divide the positive set into a training set and a test set by taking all the documents that have been annotated by two annotators and assigning them to the test set. The remaining documents constitute the training set. The same procedure is performed for the negative documents. The training set is used for the baseline system and the test set is for validating both the baseline system and our approach. Table 6.8 summarizes the dataset composition.

Class	**Train**	**Test**
Positive	334	53
Negative	197	35
Total	531	88

Table 6.8: Training and test sets from the DOXA project

6.4.2 Classification

Given a text T as a collection of terms (words) $\{w_1, w_2, w_3, \ldots, w_n\}$ we define its polarity score as a mean of valence scores of all the terms:

$$\text{polarity}(T) = \frac{\sum_{i=1}^{n} \text{polarity}(w_i)}{n} \tag{6.6}$$

where valence score of a term w_i is calculated using the constructed affective lexicon and equal to smoothed delta idf[7]:

7. inversed document frequency

$$\text{polarity}(w_i) = \log \frac{N(w_i, M_{pos}) + 1}{N(w_i, M_{neg}) + 1} \tag{6.7}$$

The classification decision is based on the obtained $polarity(T)$. If $polarity(T) > avg.polarity$, where $avg.polarity$ is a mean polarity score of all the terms in the lexicon, then the text is considered to have a positive polarity. Otherwise the text has a negative polarity.

Because the data from Twitter contains a lot of noisy text, we need to filter out those noisy terms from the obtained lexicon. We apply two rules to remove noisy terms:

1. Filter non frequent terms (words that appear less than K times)
2. Filter short terms (words with less than L characters)

Thus, we introduce two parameters that bring an impact to the final accuracy: minimum term frequency and minimum term length. By changing these parameters we try to maximize the classification accuracy.

6.4.3 Evaluation setup and results

To evaluate our system, we build a baseline based on SVM with n-grams features. We use an open source implementation of SVM from the LIBLINEAR package by Fan et al. (2008), with default parameters and a linear kernel. To construct a document feature vector, we apply delta tf-idf weighting scheme (Martineau and Finin, 2009). Negations were handled by attaching a negation particle to the preceding and the following word (see § 4.2.3, p. 23), when generating n-grams. We compare the performance of our system with three baselines: unigram, bigram, and trigram based classifiers. Average accuracy and average precision were chosen as evaluation measures.

Figure 6.6 shows the results of the parameters setting for minimum frequency and minimum length filtering. The 3D plot shows the obtained accuracy when changing both parameters. We change the minimum frequency from 0 to 100 and minimum length from 0 to 10. We obtained a maximum accuracy of 71.6% by filtering words that are shorter than 4 characters and those that appear less than 10 times in the Twitter corpus.

The results of the classification evaluation are presented in Table 6.9. We observe that our proposed method (referenced in the table as *Twitter*) provides comparable results to the baselines. The yielded accuracy is 71.59% which is better than the accuracy yielded by a trigram based classifier and almost as good as bigram and unigram based ones. Notice that the baseline classifiers use annotated data to train a model, while our method does not require any training data.

Model	**Acc**	$\mathbf{Pr}_{macro}$	$\mathbf{Pr}_{pos}$	$\mathbf{Pr}_{neg}$
Unigram	73.86	69.57	90.57	48.57
Bigram	72.73	69.11	86.79	51.43
Trigram	64.77	60.08	83.02	37.14
Twitter	71.59	68.16	84.90	51.40

Table 6.9: Accuracy and precision of polarity classification with unigram, bigram, trigram models and our proposed model

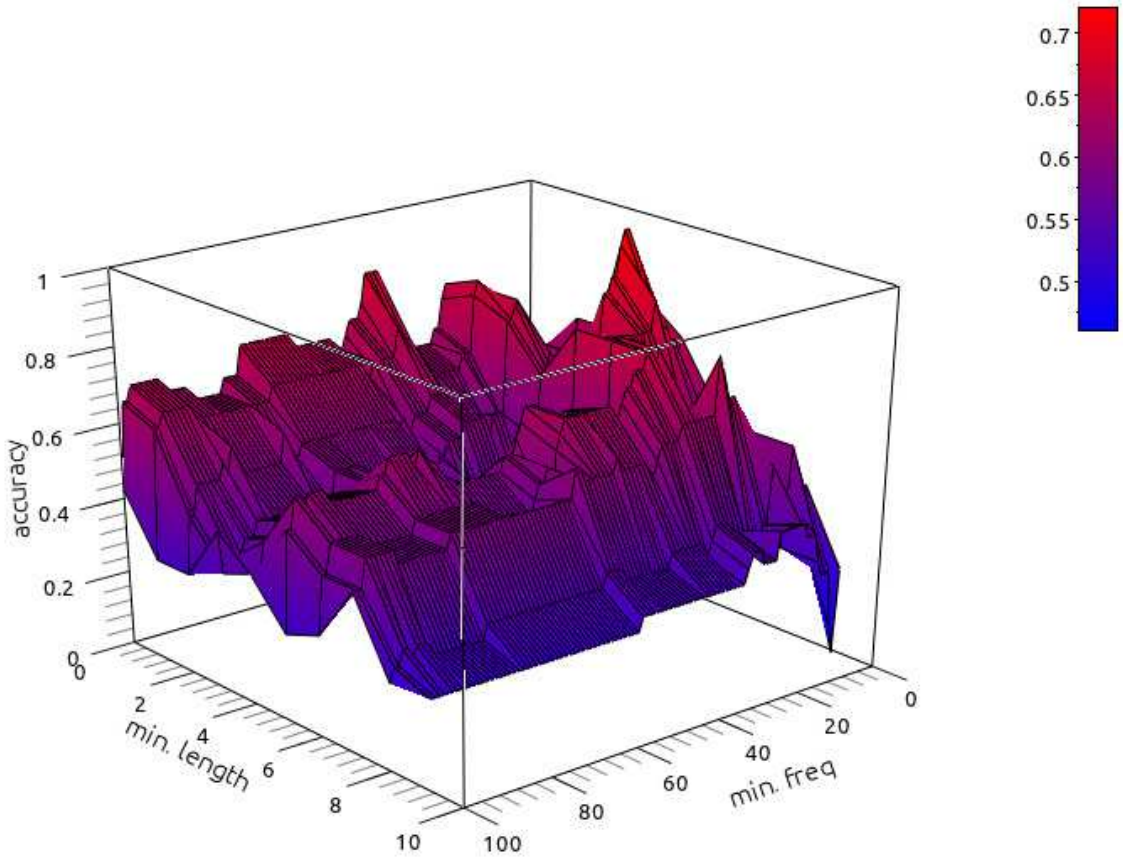


Figure 6.6: The impact of parameter settings (min frequency and min length) to the classification accuracy

6.5 Conclusions

Microblogging nowadays became one of the major types of the communication. The large amount of information contained in microblogging web-sites makes them an attractive source of data for opinion mining and sentiment analysis.

In our research, we have presented a method for an automatic collection of corpora that can be used to train a sentiment classifier. We addressed several tasks that our approach may be applied to. Our intention was not to use sophisticated linguistic tools. We want to keep our approach language independent or at least easy to port to other languages. The presented approach is based on the multinomial Naïve Bayes classifier that uses n-grams and POS features extracted from a set of training data. Our method is fully automated, we do not use any additional hand-built language resources.

We have conducted evaluation experiments using hand annotated corpus, hand-built lexicon and we participated in SemEval evaluation campaign (Chapter 9, p. 91). Our system showed good performance in the addressed tasks.

BEYOND THE BAG-OF-WORDS MODEL: USING DEPENDENCY GRAPHS

7

N-gram model is a traditional text representation used in many applications of information retrieval and natural language processing, including sentiment analysis. The major flaw of the n-gram model is information loss due to the assumption of word independence. We believe that it crippling for sentiment analysis task as user opinions are expressed using complex language constructions. Our intention is to compensate the information loss of the n-gram model but at the same time keep the model simple enough to avoid overfitting. We propose to use dependency parsing output to construct n-gram like features. We call the proposed model **d-grams**. To prove the efficiency of the new model, we perform experimental evaluations in English and French using three different dependency parsers over a multidomain dataset of product reviews. The reported results show the improvement of polarity classification accuracy when using different machine learning classifiers.

7.1 Motivation

A common approach for solving polarity classification problem is by training a supervised machine learning classifier such as support vector machines (SVM) or a Naïve Bayes (NB) classifier (see § 5.2.1, p. 42). To produce a classifier's input, texts from the dataset are being represented as bag of words or n-grams (see § 5.2.2, p. 43). Thus by doing this, we make an assumption of words independence, i.e. we consider a text as a unordered list of words disregarding relationships between them. However, we believe that these relationships carry information which is important for sentiment analysis and therefore propose another way to construct n-gram like terms for text representation.

Let's consider an example: "I did not like this video for several reasons". Notice that an n-gram representation of this sentence would contain terms that can confuse a classifier and cause misclassification of opinion polarity. For example, a unigram representation contains a term "like", and bigrams contain a term "like this" (which can be considered as positive terms such as in "I like this video"). One way to deal with this problem is to use higher order n-grams. However, using higher order n-grams would cause model overfitting and thus reduce the performance of our classifier. To deal with this

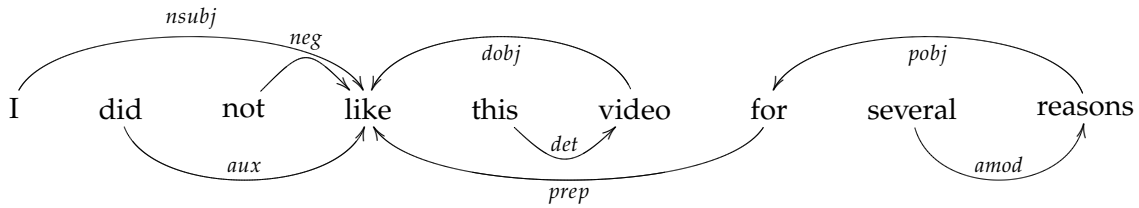


Figure 7.1: Dependency graph of a sentence "I did not like this video for several reasons"

problem, we propose to use dependency graphs produced by dependency parsers.

A dependency graph (Nivre, 2005) is a graphical representation of a sentence where nodes correspond to words of the sentence and edges represent syntactic relations between them such as 'object', 'subject', 'modifier' etc. Figure 7.1 displays dependency graph for our example sentence. Such sentence representation suits well for our needs, it captures long dependencies and provide correct word attachments. Our idea is to generate features from the dependency graph by splitting it into triples consisting of a source word, a dependency type, and a target word. The dependency graph depicted above, generates the following triples:

$$\{(\text{I}, \text{nsubj}, \text{like}),\\ (\text{did}, \text{aux}, \text{like}),\\ (\text{not}, \text{neg}, \text{like}),\\ (\text{this}, \text{det}, \text{video}),\\ (\text{video}, \text{dobj}, \text{like}),\\ (\text{for}, \text{prep}, \text{like}),\\ (\text{several}, \text{amod}, \text{reasons}),\\ (\text{reasons}, \text{pobj}, \text{for})\}$$

7.2 Related work

Sentence dependency tree has been widely used in the sentiment analysis domain. A recent research by Arora et al. (2010) noted the problems of the standard bag-of-words text representation. The authors suggested their algorithm to extract subgraph features using genetic programming. However, the obtained features were not used to replace the standard n-gram model, but rather as a complementary set of features. Another recent research by Nakagawa et al. (2010) used dependency tree to obtain features that were used to train a CRF classifier for sentiment polarity detection. In Zhuang et al. (2006), authors used dependency tree to extract feature-opinion pairs, where the first member of the pair is a feature term (such as movie") and the second is an opinionated term (such as "masterpiece"). The dependency tree is used to establish relations between feature words

and opinion keywords. In Chaumartin (2007), dependency tree was used to normalize headlines to grammatically correct form for further sentiment tagging. In Meena and Prabhakar (2007), authors used dependency tree to analyze the sentence construction along with WordNet to perform sentence level sentiment classification. Several studies have also used dependency parsing for language modeling and machine translation (Wu and Khudanpur, 1999; Habash, 2004; Guo et al., 2008; Popel and Mareček, 2010).

7.3 D-grams

Given a document D as a list of sentences:

$$D = [S_1, S_2 \dots S_n] \tag{7.1}$$

where each sentence is a list or tokens (words and punctuations):

$$S_i = [t_1, t_2 \dots t_m] \tag{7.2}$$

For each sentence, we obtain its tagged version:

$$\text{tag}(S_i) = [(t_1^{\text{form}}, t_1^{\text{pos}}, t_1^{\text{lem}}) \dots (t_m^{\text{form}}, t_m^{\text{pos}}, t_m^{\text{lem}})] \tag{7.3}$$

where t_i^{form} is a word's form as it is in the original sentence, t_i^{pos} is a part-of-speech tag, t_i^{lem} is a word's lemma. We use $\text{tag}(S_i)$ as input for a dependency parser to obtain sentence dependency graph G_i:

$$G_i = (S_i, E_i) \tag{7.4}$$

E_i is a set of edges between words in a sentence:

$$E_i = \{e_1, e_2 \dots e_k\} \tag{7.5}$$

where e_i is a triple $(e_j^{\text{s}}, e_j^{\text{d}}, e_j^{\text{t}})$, e_j^{s} is a source word, e_j^{t} is a target word, e_j^{d} is a dependency type (e.g. *nsubj* subject, *dobj* direct object, etc.).

In a traditional n-gram model, we obtain a set of n-grams from a sentence by splitting it into subsequences of fixed size:

$$\begin{aligned} \text{unigrams}(S_i) &= \{(t_1), (t_2) \dots (t_m)\} \\ \text{bigrams}(S_i) &= \{(t_1, t_2), (t_2, t_3) \dots (t_{m-1}, t_m)\} \end{aligned} \tag{7.6}$$

We propose to construct n-grams from the dependency graph of a sentence. We refer to the constructed n-grams as d-grams:

$$\text{dgrams}(S_i) = \{(e_1^{\text{s}}, e_1^{\text{d}}, e_1^{\text{t}}) \dots (e_k^{\text{s}}, e_k^{\text{d}}, e_k^{\text{t}})\} \tag{7.7}$$

7.3.1 Example

Let's consider an example:

$S =$ "The soundtrack was awful"

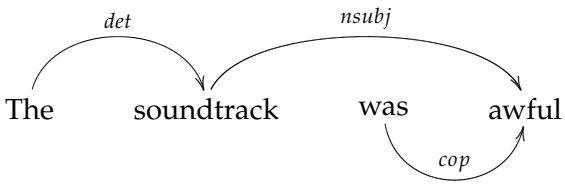


Figure 7.2: Dependency graph of a sentence "The soundtrack was awful"

The dependency graph of the sentence is shown in Figure 7.2. From the obtained dependency graph, we construct following d-grams:

$$\begin{aligned}\text{dgrams}(S) = \{&(\text{The}, \text{det}, \text{soundtrack}),\\ &(\text{soundtrack}, \text{nsubj}, \text{awful}),\\ &(\text{was}, \text{cop}, \text{awful})\}\end{aligned}$$

For comparison, here are unigram and bigram representations of the same sentence:

$$\begin{aligned}\text{unigrams}(S) = \{&(\text{The}),\\ &(\text{soundtrack}),\\ &(\text{was}),\\ &(\text{awful})\}\end{aligned}$$

$$\begin{aligned}\text{bigrams}(S) = \{&(\text{The}, \text{soundtrack}),\\ &(\text{soundtrack}, \text{was}),\\ &(\text{was}, \text{awful})\}\end{aligned}$$

7.3.2 Wildcards

D-grams obtained from the previous step better preserve relation information between words in a sentence. However, they became more specific features and can cause overfitting. To keep our features general enough, we add *wildcards*. A wildcard (denoted as asterisk '*') is a placeholder that replaces an element (source, dependency type, or target) in a d-gram and matches any word or a dependency type. Given a d-gram $(\text{soundtrack}, \text{nsubj}, \text{awful})$, we produce a new d-gram with a wildcard: $(*, \text{nsubj}, \text{awful})$ that matches a phrase "Acting is awful" even if it does not present in the training data since the wildcard substitutes a source word and matches "acting".

We can obtain 3 new triples by inserting 1 wildcard:

1. $(\text{source}, *, \text{target})$
2. $(\text{source}, \text{type}, *)$
3. $(*, \text{type}, \text{target})$

There are also 3 possible triples we can obtain by inserting 2 wildcards. However, replacing both the source and the target does not make sense, which leaves us with 2 possible triples:

1. $(\text{source}, *, *)$
2. $(*, *, \text{target})$

Notice, the former triple corresponds to a unigram feature, and the latter one corresponds to a target word being a head of a dependency.

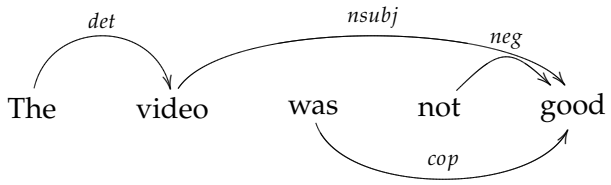
In our experiments we use a set of d-grams with 1 wildcard and refer to it as simply *d-grams*. An example sentence "The soundtrack was awful" is represented by the following set of d-grams:

$$\begin{aligned}\text{dgrams}(S) = \{&(\text{The}, *, \text{soundtrack}),\\ &(\text{The}, \text{det}, *),\\ &(*, \text{det}, \text{soundtrack}),\\ &(\text{soundtrack}, *, \text{awful}),\\ &(\text{soundtrack}, \text{nsubj}, *),\\ &(*, \text{nsubj}, \text{awful}),\\ &(\text{was}, *, \text{awful}),\\ &(\text{was}, \text{cop}, \text{awful}),\\ &(*, \text{cop}, \text{awful})\}\end{aligned}$$

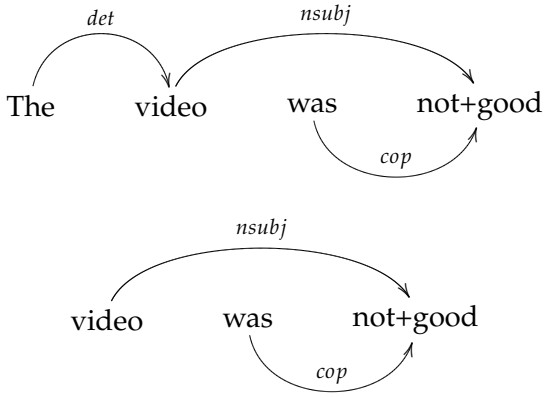
The extended set of d-grams – **xd-grams** (1 triple without wildcards, 3 triples with one wildcard and 2 triples with two wildcards) yields better accuracy but at the same time requires much more computational time to train a model.

7.3.3 Fusing and pruning triples

To profit from the additional information that we obtain when parsing a text, we look at the dependency types. We notice that some types of dependencies convey more relevant information than others. For example, negations are believed to be important for sentiment analysis (§ 4.2.3, p. 23). In a dependency graph, negations can be recognized by a link *neg* connecting a negation particle with negated word. To treat negations, we fuse the source with the target of the negation triple, and use the obtained node instead of the target word. For example, given two graph:



We fuse elements in triple 1 and make necessary replacements in triple 2: On the other hand relations, such as determiners, possessives, and noun modifiers do not add much to our model. Therefore, we prune triples containing these relations:
These step are performed before construction of d-grams.



7.4 Experiments

7.4.1 Data and setup

For experimental evaluations, we use the Cross-Lingual Sentiment[1] dataset constructed by Prettenhofer and Stein (2010). The dataset is composed by product reviews from Amazon in four languages. In this research, we use reviews in English and French. Reviews are separated in three domains (product types): books, music, and DVD. Each domain contains 2000 positive and 2000 negative reviews which makes total of 24000 documents that were used in this work.

Data preprocessing includes part-of-speech tagging and parsing We use TreeTagger for tokenization and tagging. For English, we use MaltParser[2] by Nivre et al. (2006) and Stanford Lexical parser[3] by de Marneffe et al. (2006). For French, we use Bonsai[4] package by Candito et al. (2010) which includes MaltParser and Berkeley parser. Our goal was not to make a benchmarking of different dependency parsers, but rather to show that the proposed method works with publicly available tools and the quality of the obtained results is not much influenced by the choice of software.

To evaluate our model, we use an implementation of linear SVM from the LIBLINEAR package by Fan et al. (2008) and our own implementation of Naïve Bayes classifier. For SVM, we set default parameters and a binary classification mode. To ensure that the difference in performance is not caused by tokenization, we use the tokenized output from TreeTagger to produce n-grams.

Since our dataset is balanced (i.e. contains positive and negative sets of equal sizes), we use accuracy (Equation 4.5, p. 32) to measure the performance. We perform 10-fold cross validation to estimate average accuracy.

7.4.2 Results

Figure 7.3 displays obtained averaged accuracy across all the domains. For all the models we used binary weighting scheme, i.e. capturing presence of a feature in the document when constructing a feature vector. The top chart corresponds experiments using the

1. Cross-Lingual Sentiment dataset is available at: http://www.uni-weimar.de/cms/medien/webis/research/corpora/webis-cls-10.html

2. MaltParser: http://maltparser.org/

3. Stanford Lexical parser: http://nlp.stanford.edu/software/stanford-dependencies.shtml

4. Bonsai package: http://alpage.inria.fr/statgram/frdep/fr_stat_dep_parsing.html

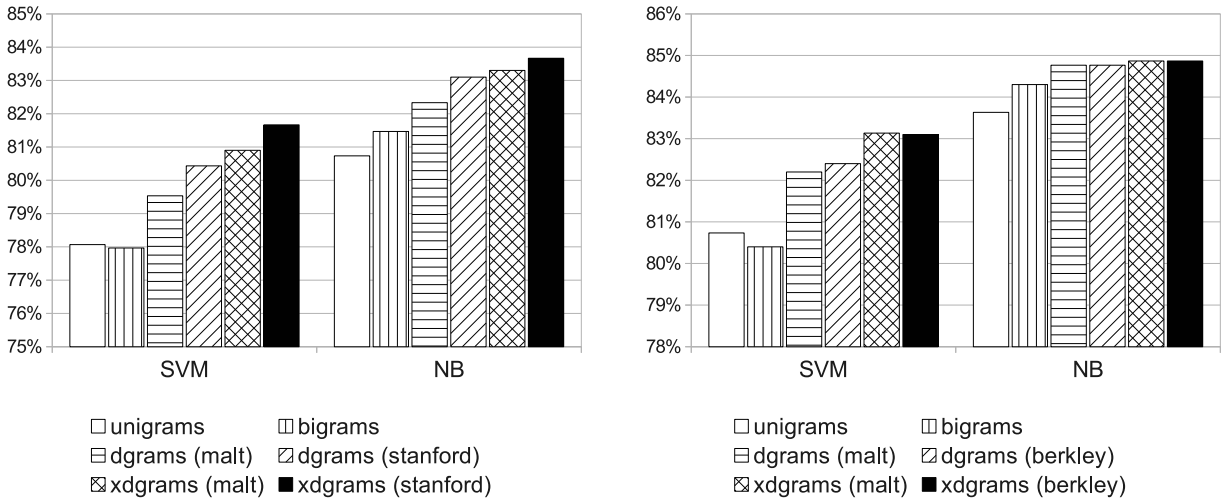


Figure 7.3: Classification accuracy averaged across different domains in English (on top) and French (on bottom) using traditional n-grams model and proposed d-grams.

		N-grams		D-grams		Extended d-grams	
		unigrams	**bigrams**	**malt**	**stanford**	**malt**	**stanford**
SVM	dvd	77.9	77.8	79.9	81.1	81.9	**82.3**
	books	77.7	78.3	79.2	80.2	80.2	**81.2**
	music	78.6	77.8	79.5	80.0	80.6	**81.5**
NB	dvd	81.0	81.0	82.8	83.4	83.6	**83.9**
	books	81.4	80.7	82.4	83.6	83.5	**84.1**
	music	79.8	82.7	81.8	82.3	82.8	**83.0**

		N-grams		D-grams		Extended d-grams	
		unigrams	**bigrams**	**malt**	**berkeley**	**malt**	**berkeley**
SVM	dvd	79.3	79.5	80.9	81.5	81.6	**81.9**
	books	79.0	79.3	80.6	81.0	**82.3**	82.2
	music	83.9	82.4	85.1	84.7	**85.5**	85.2
NB	dvd	82.9	83.5	84.3	83.9	**84.1**	**84.1**
	books	82.5	82.4	82.6	83.3	83.4	**83.5**
	music	85.5	87.0	**87.4**	87.1	87.1	87.0

Table 7.1: Classification accuracy across different domains in English (on top) and French (on bottom) using traditional n-grams model and proposed d-grams.

	English	French	Fr-En Translation
Positive d-grams	really advmod enjoyed	ne mod lasse	doesn't tire
	the det team	pas mod lasse	doesn't tire
	highly nsubj recommended	top dep au	at thetop
	out prt stands	manquer obj à	don't miss
	can't aux wait	sans mod revoir	watch again without
	right pobj in	absolument mod posséder	must have
	own amod right	modération obj sans	without moderation
	too advmod good	décors mod est	decorations are
	to aux stand	ne mod manquer	not to miss
	well advmod worth	le det quotidien	the daily
Negative d-grams	terrible root root	ce det navet	the turnip (rubbish)
	two num stars	une det honte	a shame
	awful root root	suis aux_pass déçu	am disappointed
	avoid root root	mauvais mod film	bad movie
	money dobj save	très mod déçu	very disappointed
	is cop waste	déçue root root	disappointed
	wasted root root	ennui obj d'	bored of
	money conj time	ne mod suffit	not enough
	don't aux recommend	ne mod vaut	not worth
	was cop boring	pas mod suffit	not enough

Table 7.2: Top positive and negative d-grams from English and French DVD reviews. We have not observed significant cultural differences.

English part of the dataset and the bottom chart corresponds to the French one. More detailed numbers with separation by domains are presented in Table 7.1. In English data, d-grams obtained using Stanford Parser output yielded higher accuracy than d-grams obtained with MaltParser. In French data, the results were similar between MaltParser and Berkeley.

Results over both languages show similar trends. All the models showed slightly better results on French data. That can be caused by the quality of parsing as we have not observed significant cultural differences of expressing opinions. Table 7.2 shows top-10 positive and negative d-grams from DVD reviews in English and in French.

Suprisingly, Naïve Bayes yielded better accuracy than SVM. On both languages and across all the domains, the results obtained with NB are higher than those using SVM up to 3.9%. The performance of both SVM and NB improves when using our proposed d-gram representation. While unigrams and bigrams perform similarly in general, d-grams perform better on each dataset. We also observe that extended d-grams yield higher accuracy, however they require more time to train the model as the number of features is doubled.

7.5 Conclusion

The n-gram model is a traditional text representation which is often used in sentiment analysis. However, we believe that difficulty of this task requires new models that are better suited for capturing user opinions. In this work, we refer to dependency parsing as a possible source for such a representation. From our observations, we concluded that dependency graphs convey more information that is important for sentiment analysis rather then simple bag-of-words.

We have proposed a method of constructing n-gram like features from triples of a given dependency graph which we call d-grams. Evaluation experiments using different dependency parsers and machine learning classifiers over a multidomain dataset in English and French have shown the efficiency of d-grams as compared to traditional unigrams and bigrams. Therefore, we conclude that our method is generic and can be applied to texts in different languages and domains.

In this work, we have not fully employed all the advantages of dependency graphs. In the future, we plan to improve our model by using dependency type information along with considering combinations of triples rather then treating them separately. We show how we use d-grams for polarity classification in Chapter 10 (p. 97) and for emotion detection in Chapter 11 (p. 107).

IMPROVING WEIGHTING SCHEMES FOR POLARITY CLASSIFICATION

8

As we have described in § 4.3.2 (p. 30), nowadays it is quite easy to collect vast datasets of opinionated texts from social networks, product review web-sites, forums etc. Therefore it is possible to build a system that would classify opinion polarities of texts of the same nature as the training corpus with an acceptable level of accuracy. However such a system would be biased towards the opinions included in its training data. If we would use it to analyse polarities of reviews on a certain product that already has many positive reviews, it is highly probable that all the analysed reviews would get assigned a positive class because they contain the same features (such as a product name and a model) as the reviews for the same product in our training set. The same tendency is hold for other types of reviews, such as movies: reviews for the same movie usually contain its title and names of the director, producers, and the cast. We refer to these features as *entity-specific* and opinion target (such as a movie or a product) as *entity*.

Paradoxically this bias problem improves the overall accuracy of a sentiment analysis system because the distribution of positive vs. negative reviews for a product in the test set is usually the same as in the training set. If a product has already received many positive reviews it will receive more positive than negative reviews in the test set and vice versa. However, we might want to have a system that not only has a good overall classification accuracy, but can also determine correctly the polarities of minor reviews: reviews that criticize a good product in spite of other people's praise or on the contrary find good points about a bad product. Such a system can be compared to an objective expert that takes decision based on the information contained in the review and does not take into consideration prior opinions about the reviewed product.

To prove our argument, we take two standard datasets: movie reviews and product reviews. Both datasets were previously used in related sentiment analysis research by Maas et al. (2011), Blitzer et al. (2007), Duh et al. (2011). For each dataset, we compose its *biased* version, by first grouping reviews by their target entities (a movie or a product) and then selecting groups with uneven distribution of positive and negative reviews. We show that traditional settings, i.e. SVM with n-gram features perform much worse at minor reviews

	Movies	Kitchen	Books	DVD	Electronics
Average document size:					
Words	284	75	92	95	74
Characters	1309	340	447	447	338
Initial number of reviews:					
Positive	25000	13793	5192	30600	10328
Negative	25000	2991	807	4140	2824
Training and test sets sizes:					
Training	3680	2000	480	2568	2160
Test	1580	664	160	856	720

Table 8.1: Characteristics of preprocessed movie and product review datasets. The size of each dataset was reduced after dividing them into training and test sets.

classification as compare to classification of major reviews in these biased settings. We propose three schemes for normalizing feature weights to reduce the importance of the entity-specific terms causing misclassification of the minor reviews. The first scheme is based on average term frequency. It lowers the importance of n-grams that are frequently used within a document. The second scheme is based on term's occurrence across different entities as compared to its occurrence across the reviews. It lowers the importance of n-grams that appear rarely in different entities relatively to its occurrence in different reviews. Finally, the combined scheme uses both normalizations.

8.1 Data

To investigate how prior reviews of a product influence the classification decision, we use standard datasets for sentiment analysis research, but separate then into a training and a test set in a special way. Movie reviews is one of the frequently used data source for sentiment analysis. We use Large Movie Review Dataset[1] collected by Maas et al. (2011) which contains 50,000 texts with equal proportion of negative and positive opinion classes. Another popular type of data is product reviews. We additionally use a raw version of Multi-Domain Sentiment Dataset[2] collected by Blitzer et al., 2007 which includes reviews in 4 different domains (product types): kitchen, books, DVDs, and electronics. Characteristics of both datasets are presented in Table 8.1. We separate product reviews into 4 domains and treat them as separate datasets.

To balance the dataset and to have equal numbers of positive and negative reviews in the training and testing sets, we fix the number of reviews used for training and testing for each movie or product. From the movie reviews, we took 3 documents of each movie for test and 7 for training. From the product reviews, we took 1 document of

1. Large Movie Review Dataset is available at: http://ai.stanford.edu/~amaas/data/sentiment/

2. Multi-Domain Sentiment Dataset is available at: http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

each product for test and 3 for training. These numbers were chosen heuristically with a criteria to maximize the total number of reviews to be used both for training and during testing. We illustrate the process of dataset composition in Figure 8.1.

To separate a dataset into training and test sets, we proceed as follows. First, we group all the reviews by their entity (movie or product) identified by a unique ID in the dataset. Next, we select groups that have enough numbers of positive and negative reviews. From the selected groups, from each entity we select all the reviews of a dominant polarity in this group and move them to the training set. The remaining reviews from each group are moved to the test set.

For example, if a product has more positive reviews, we use them for training while its negative reviews are used for testing. Otherwise, we use negative reviews for training and positive reviews for testing. After this procedure, we obtain the training and the test set composed of reviews for the same number of entities and the same number of positive and negative classes. However, for each entity, its reviews in the training set have different polarity than in the test set. This way we simulate a case when the reviews are biased towards a certain opinion and the test samples have different polarity. We call this dataset *minor biased* as the test set contains reviews with minor polarities.

We expect traditional settings for polarity classifiers to yield worse results on this dataset due to the bias in reviews for each product. To prove that the drop of the performance is caused by the biased features, we construct a dataset composed of the same reviews but reorganized such that the reviews in the test set for each entity have the same polarity as the dominant polarity in the training set for each entity. We call this dataset *major biased* as the test set contains reviews with major polarities. For each biased variant, we additionally create an *excluded setting*, in which where we perform an evaluation for each of entity separately and during the training stage we exclude the samples belonging to this entity. Finally, we compose the *unbiased* dataset, by separating reviews of different entities into training and test sets, such that entities in the test set have no reviews in the training set.

8.2 Our method

Given a document d as a set of terms:

$$\mathrm{d} = \{g_1, g_2, \ldots, g_k\} \tag{8.1}$$

we define a feature vector of d as

$$\vec{\mathrm{d}} = \{w(g_1), w(g_2), \ldots, w(g_k)\} \tag{8.2}$$

where $w(g_i)$ is a weight function of a term g_i. We consider two weighting schemes which are used in sentiment analysis.

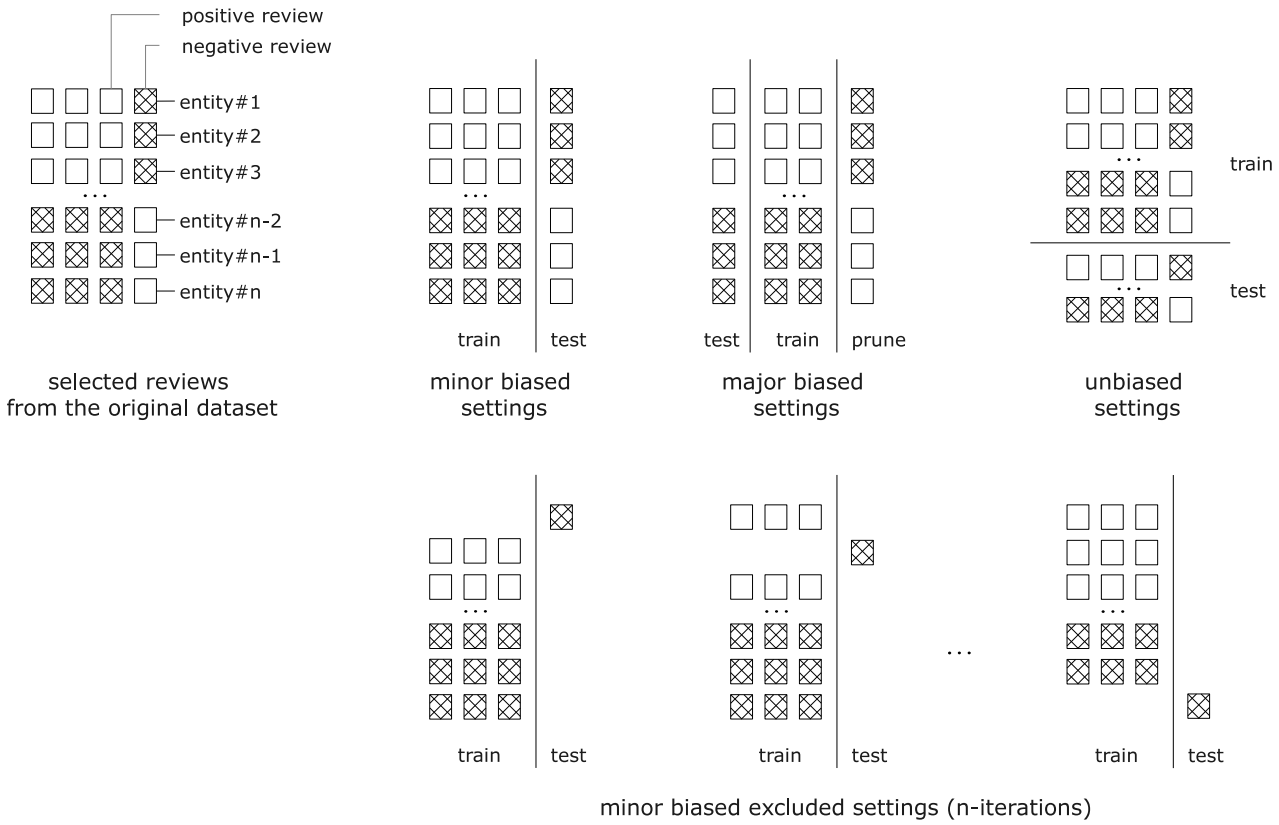


Figure 8.1: Dataset composition process

- **Binary** weights were used in first experiments by Pang et al. (2002) and proven to yield better results than traditional information retrieval weighting such as TF-IDF. Assigns equal importance to all the terms presented in a document:

$$w(g_i) = 1, \text{if} g_i \in d, \text{otherwise} = 0 \tag{8.3}$$

  where g_i is a term (n-gram), d is a document.

- **Delta tf-idf** was proposed by Martineau and Finin (2009) and proven to be efficient by Paltoglou and Thelwall (2010), assigns more importance to terms that appear primarily in one set (positive or negative):

$$w(g_i) = \text{tf}(g_i) \cdot \log \frac{\text{df}_\text{p}(g_i) + 0.5}{\text{df}_\text{n}(g_i) + 0.5} \tag{8.4}$$

  where $\text{tf}(g_i)$ is term-frequency of a term (number of times g_i appears in document D), $\text{df}_\text{p}(g_i)$ is positive document frequency (number of times g_i appears in documents with positive polarity), $\text{df}_\text{p}(g_i)$ is negative document frequency.

In order to improve the classification of minor reviews, we want to reduce the importance of terms that may bias the classification decision. For that, we propose two measures: average term frequency and entity proportion.

Unigram	Avg TF	Unigram	Avg TF	Unigram	EP	Unigram	EP
the	12.17	absolute	1.0	not+nearly	7.0	and	1.04
a	6.19	travel	1.0	subplots	7.0	a	1.04
and	6.16	disjointed	1.0	disturbed	7.0	of	1.06
of	5.49	altogether	1.0	olds	7.0	to	1.07
to	5.27	doubts	1.0	positively	7.0	this	1.12
i	4.47	split	1.0	affairs	7.0	it	1.14
fulci	4.31	beloved	1.0	altogether	7.0	is	1.14
is	4.21	hat	1.0	doubts	7.0	in	1.16
it	4.17	shadow	1.0	beloved	7.0	i	1.24
in	3.84	suffice	1.0	hat	7.0	that	1.26
helen	3.83	whoever	1.0	crowd	7.0	**reiser**	1.27
hitch	3.64	unintentionally	1.0	beings	7.0	**belushi**	1.27
that	3.4	accomplished	1.0	tame	7.0	**ringwald**	1.27
this	3.26	opposed	1.0	greatness	7.0	**rickman**	1.31
carrie	3.25	pulled	1.0	chair	7.0	s	1.44
hartley	3.09	suspension	1.0	stays	7.0	but	1.45

Table 8.2: List of unigrams with highest and lowest values of average term frequency (on the left) and entity proportion (on the right). Entity specific features are highlighted in bold.

8.2.1 *Average term frequency*

Average term frequency (TF) is an average number of times a term occurs in documents

$$\text{avg.tf}(g_i) = \frac{\sum_{\{d|g_i \in d\}} \text{tf}(g_i)}{\|\{d|g_i \in d\}\|} \tag{8.5}$$

where $\{d|g_i \in d\}$ is a set of documents containing g_i

Average term frequency normalization is based on the observation that review authors tend to use a rich vocabulary when expressing their attitude towards a movie or a product. Thus, terms related to a sentiment expression (such as *outstanding* or *lovingly*) have average frequency close or equal to 1 while other non-subjective terms have a higher average term frequency. This includes movie names, actors, brands and product parts as they are mentioned several times within documents. To normalize a document vector, we divide each term's weight by its average TF:

$$w(g_i)^* = \frac{w(g_i)}{\text{avg.tf}(g_i)} \tag{8.6}$$

8.2.2 *Entity proportion*

Entity proportion (EP) is a proportion of term's occurrences across different entities as compared to document frequency

$$\text{ep}(g_i) = \log\left(\frac{\|\{e|g_i \in e\}\|}{\|\{d|g_i \in d\}\|} \cdot \frac{\|D\|}{\|E\|}\right) \tag{8.7}$$

where $\{e|g_i \in e\}$ is a set of entities that contain g_i in their reviews, $\|D\|$ is a total number of documents, $\|E\|$ is a total number of entities

Entity proportion normalization favors terms that appear in many entities but not in many documents. We observe three types of terms:

1. Terms specific to an entity, such as movie and product names, would appear in few entities and thus in few documents. The EP value should be close to the normalization constant $\frac{\|D\|}{\|E\|}$ (average number of documents per product).
2. Subjective terms (such as "outstanding" or "lovingly"), would appear in many products and in a relatively small number of documents (because authors tend to use a rich vocabulary). The EP value will be greater than the normalization constant.
3. Stopwords, such as determinants and prepositions, would appear in almost all products and almost all documents. The EP value will be close to the normalization constant.

To normalize a document vector, we multiply each term's weight by its product proportion:

$$w(g_i)^* = w(g_i) \cdot \text{ep}(g_i) \tag{8.8}$$

Table 8.2 shows examples of unigrams with high and low values of avg.tf and ep. Finally, we also consider a combination of both normalization schemes:

$$w(g_i)^* = w(g_i) \cdot \frac{\text{ep}(g_i)}{\text{avg.tf}(g_i)} \tag{8.9}$$

8.3 Experiments and results

8.3.1 Setup

In each of our experiments, we use an implementation of linear SVM from LIBLINEAR package by Fan et al. (2008). We set default parameters and a binary classification mode. Texts from the reviews were preprocessed minimally. We considered any sequence of non-alphabetic characters as word boundaries to tokenize a text into words. Negation particles (*no*, *not*) were attached to its preceding and following words (§ 4.2.3, p. 23). For example, the text "I do not like this movie" would produce a sequence of unigrams {I, do+not, not+like, this, movie}. As our datasets are balanced (contain negative and positive sets of equal sizes), we use accuracy (Equation 4.5, p. 32) as the evaluation measure.

8.3.2 Unbiased vs. biased vs. biased excluded

First, we prove the negative effect of entity specific features on classification accuracy of minor reviews. We run experiments on 5 variants of the datasets: unbiased (unb), minor biased (minb), minor

		unb	minb	minx	majb	majx
Movies	uni + bin	80.7	69.4	75.3	83.4	80.8
	uni + Δ	83.4	63.5	75.1	89.2	84.2
	bi + bin	79.6	71.9	74.1	83.5	81.1
	bi + Δ	83.0	69.9	77.8	87.6	84.0
Books	uni + bin	65.6	46.9	53.1	70.6	70.0
	uni + Δ	64.4	46.3	56.9	76.9	68.8
	bi + bin	61.3	50.6	53.1	67.5	65.6
	bi + Δ	65.0	51.3	56.3	66.3	62.5
DVD	uni + bin	72.6	64.1	65.9	72.9	70.9
	uni + Δ	75.8	63.7	65.7	76.1	73.8
	bi + bin	72.3	65.1	65.4	70.4	69.6
	bi + Δ	76.3	64.0	64.7	74.0	73.0
Kitchen	uni + bin	79.4	72.6	73.9	78.1	76.3
	uni + Δ	76.8	70.5	71.8	78.9	77.2
	bi + bin	78.3	74.6	75.9	77.8	76.3
	bi + Δ	81.2	73.6	74.8	78.9	78.3
Electronics	uni + bin	72.9	66.7	67.4	71.9	71.1
	uni + Δ	76.1	68.1	69.0	71.7	71.7
	bi + bin	74.9	68.9	69.9	73.6	72.9
	bi + Δ	76.1	69.0	69.4	75.4	74.4

Table 8.3: Classification accuracy across different datasets. Notice the difference between biased and biased excluded variants (minb vs minx, majb vs majx).

biased excluded (minx), major biased (majb), major biased excluded (majx). We use unigrams (uni) and bigrams (bi) with binary (bin) and Delta TF-IDF (Δ) weights. Results on classification accuracy across the datasets and features are presented in Table 8.3. Notice that we cannot directly compare accuracy values across different variants of datasets, as they are composed of different test data, except for pairs: minb vs. minx, and maxb vs. majx. However, we assume that our datasets are homogeneous and results obtained with different dataset variants reflect the complexity of the classification task.

Entity-specific features cause performance drop on the minor biased set as compare to the unbiased set (unb vs. minb). Accuracy increases when we remove reviews of the same entity thus removing entity-specific features (minb vs. minx). We also observe a boost in performance on the major biased dataset in spite of a smaller training size (unb vs. majb). When removing reviews of the same entity, accuracy decreases on major biased dataset (majb vs. majx). This shows that our classifier learns the mapping between entity specific features to entity major polarity, instead of learning affective language model. All the results are similar across different datasets, variants of datasets, and features.

Delta tf-idf while improves overall accuracy, causes misclassification of minor review as it gives more importance to entity-specific

features. We can observe that by comparing the results of using delta tf-idf (uni + Δ and bi + Δ) on the minor biased set with the unbiased and major biased datasets.

8.3.3 Normalization schemes

Next, we evaluate the effect of the proposed normalization schemes on classification accuracy. The proposed normalization measures should lower importance of entity-specific features, thus we expect boost in performance on minor biased dataset. Our goal is to improve classification of minor reviews. As we observer from the previous experiments, excluding reviews of the same entity increases the performance. However, in real settings this approach is not feasible as it requires more computational resources, since we need to train the classifier for each entity. We test the normalization schemes on 3 datasets: unbiased, minor biased and major biased. The results of the evaluations are presented in Table 8.4 for movie reviews and Table 8.6 – 8.9 for product reviews. Δ column shows a gain in accuracy when applying a normalization scheme as compare to no normalization been applied.

From the results, we observe increase of accuracy on minor biased set up to 6% on movie reviews and 12.5% on book reviews when using proposed normalization. Both on unbiased and major biased datasets, the results are quite interesting: the normalization slightly decreases performance of delta tf-idf weighting while improving binary scheme. This can also be served as a proof that delta tf-idf favors entity-specific features and our normalization lowers this effect. The combined normalization scheme yields better accuracy in general and can be used if a dataset allows to compute entity proportion values (i.e. reviews contain products IDs). Otherwise, average term frequency should be applied to normalize feature vectors.

8.4 Should a sentiment analysis system be objective?

As we observed in our experiments, when a classifier has been trained using reviews about the same entities as those contained in the test set, there is a high probability of biased classification decision towards the majority of opinions. While it actually improves the overall classification accuracy as we observed on the major biased dataset, the bias makes it harder to classify minor reviews as has been showed using the minor biased dataset. One way to deal with this is to remove training samples that contain opinions about an entity whose reviews we are to test. Results obtained on the minor biased excluded dataset are getting closer to the results obtained on the unbiased dataset. However, doing so requires more time and space resources to train and store separate models for each entity. The second problem,

	unb		**min**	Δ	**maj**	Δ
Unigrams + binary						
no	80.7		69.4		83.4	
avg.tf	81.5	+0.8	72.3	+2.9	84.8	+1.4
ep	80.1	-0.6	71.3	+1.9	83.5	+0.1
comb	80.7	+0.0	73.0	+3.6	84.4	+1.0
Unigrams + delta tf-idf						
no	83.3		63.5		89.2	
avg.tf	81.1	-2.2	69.4	+5.9	87.6	-1.6
ep	82.3	-1.0	67.2	+3.7	87.8	-1.4
comb	81.7	-1.6	69.0	+5.5	87.5	-1.7
Bigrams + binary						
no	79.6		71.9		83.5	
avg.tf	79.7	+0.1	72.8	+0.9	84.0	+0.5
ep	80.3	+0.7	74.0	+2.1	84.2	+0.7
comb	80.8	+1.2	74.9	+3.0	84.6	+1.1
Bigrams + delta tf-idf						
no	83.0		69.9		87.6	
avg.tf	82.9	-0.1	76.0	+6.0	86.1	-1.5
ep	83.2	+0.2	74.4	+4.5	86.2	-1.4
comb	83.3	+0.3	75.1	+5.2	85.8	-1.8

Table 8.4: Classification accuracy obtained using different normalization schemes on movie reviews. Accuracy improves when using proposed normalization.

which explains the gap between the unbiased and excluded settings, is that we cannot remove entity specific features by simply removing reviews related to an entity. Some entities may share the same features, for example actors that play in several movies, or product specific features.

The method we propose lowers the importance of entity specific features by normalizing their weights in a feature vector. Our method does not require additional source of information. It is automatic and can be considered multilingual as we do not use any language specific features. Evaluation experiments performed on especially organized versions of standard datasets showed improvement in classification accuracy of minor reviews. However, we also observed a slight drop in overall accuracy, because it is more benificial to follow the majority of opinions. Thus, it is for the developers of a sentiment analysis system to decide whether they prefer to have a biased system with a better overall performance or a system that handles better minor reviews. Possible applications of our approach include customer feedback analysis, rumor detection, security, i.e. systems that aim at fine grain detection of events.

In future work, we plan to continue our research on polarity classification of minor reviews. We believe that our proposed normal-

Table 8.5: Classification accuracy obtained using proposed normalization schemes on domains of product reviews.

	unb		**min**	Δ	**maj**	Δ
Unigrams + binary						
no	65.6		46.9		70.6	
avg.tf	66.9	+1.3	51.9	+5.0	70.6	+0.0
ep	65.0	-0.6	52.5	+5.6	73.1	+2.5
comb	66.2	+0.7	59.4	12.5	72.5	+1.9
Unigrams + delta tf-idf						
no	64.4		46.2		76.9	
avg.tf	61.9	-2.5	52.5	+6.3	75.0	-1.9
ep	64.4	-0.0	53.1	+6.9	75.6	-1.3
comb	63.8	-0.7	52.5	+6.3	76.2	-0.7
Bigrams + binary						
no	61.2		50.6		67.5	
avg.tf	61.2	+0.0	53.1	+2.5	65.0	-2.5
ep	61.9	+0.7	52.5	+1.9	65.6	-1.9
comb	63.1	+1.9	53.1	+2.5	63.1	-4.4
Bigrams + delta tf-idf						
no	65.0		51.2		66.2	
avg.tf	63.8	-1.2	52.5	+1.3	64.4	-1.8
ep	66.9	+1.9	54.4	+3.2	62.5	-3.7
comb	65.6	+0.6	52.5	+1.3	65.6	-0.6

Table 8.6: Books

	unb		**min**	Δ	**maj**	Δ
Unigrams + binary						
no	72.9		64.1		72.5	
avg.tf	73.5	+0.6	65.4	+1.3	72.3	-0.2
ep	72.7	-0.2	65.7	+1.6	74.4	+1.9
comb	73.4	+0.5	66.2	+2.1	74.9	+2.4
Unigrams + delta tf-idf						
no	75.8		63.7		76.0	
avg.tf	73.4	-2.4	65.2	+1.5	72.9	-3.1
ep	75.0	-0.8	65.3	+1.6	75.2	-0.8
comb	74.2	-1.6	65.9	+2.2	74.9	-1.1
Bigrams + binary						
no	72.5		65.1		70.4	
avg.tf	72.8	+0.3	65.3	+0.2	70.4	+0.0
ep	72.2	-0.3	65.5	+0.4	71.5	+1.1
comb	73.0	+0.5	65.3	+0.2	72.0	+1.6
Bigrams + delta tf-idf						
no	76.3		64.0		74.0	
avg.tf	75.9	-0.4	65.2	+1.2	74.0	-0.0
ep	76.3	-0.0	65.1	+1.1	73.7	-0.3
comb	76.6	+0.3	66.1	+2.1	74.4	+0.4

Table 8.7: DVD

	unb		**min**	Δ	**maj**	Δ
Unigrams + binary						
no	79.4		72.6		78.1	
avg.tf	79.1	-0.3	73.2	+0.6	78.1	+0.0
ep	77.9	-1.5	73.5	+0.9	78.0	-0.1
comb	78.8	-0.6	73.2	+0.6	77.8	-0.3
Unigrams + delta tf-idf						
no	76.8		70.5		78.9	
avg.tf	74.8	-2.0	71.1	+0.6	77.4	-1.5
ep	75.6	-1.2	72.4	+1.9	80.8	+1.9
comb	75.8	-1.0	72.9	+2.4	81.3	+2.4
Bigrams + binary						
no	78.3		74.5		77.8	
avg.tf	78.3	+0.0	74.7	+0.2	77.5	-0.3
ep	79.1	+0.8	74.5	+0.0	78.4	+0.6
comb	79.2	+0.9	75.2	+0.7	78.1	+0.3
Bigrams + delta tf-idf						
no	81.2		73.6		78.9	
avg.tf	80.3	-0.9	75.0	+1.4	79.0	+0.1
ep	80.9	-0.3	74.0	+0.4	79.0	+0.1
comb	80.9	-0.3	74.8	41.2	79.5	+0.6

Table 8.8: Kitchen

	unb		**min**	Δ	**maj**	Δ
Unigrams + binary						
no	72.9		66.7		71.9	
avg.tf	74.0	+1.1	67.8	+1.1	72.1	+0.2
ep	72.6	-0.3	66.8	+0.1	72.9	+1.0
comb	72.8	-0.1	68.3	+1.6	73.3	+1.4
Unigrams + delta tf-idf						
no	76.1		68.1		71.7	
avg.tf	74.3	-1.8	67.6	-0.5	71.1	-0.6
ep	76.0	-0.1	68.8	+0.7	74.0	+2.3
comb	75.7	-0.4	69.4	+1.3	73.9	+2.2
Bigrams + binary						
no	74.9		68.9		73.6	
avg.tf	75.4	+0.5	69.2	+0.3	73.3	-0.3
ep	73.8	-1.2	68.9	-0.0	73.9	+0.3
comb	74.3	-0.6	68.8	-0.2	73.9	+0.3
Bigrams + delta tf-idf						
no	76.1		69.0		75.4	
avg.tf	75.6	-0.5	68.1	-0.9	74.7	-0.7
ep	76.1	+0.0	68.9	-0.1	75.4	+0.0
comb	76.0	-0.1	69.3	+0.3	75.1	-0.3

Table 8.9: Electronics

ization measures can also be used for feature selection in other tasks of sentiment analysis. We use average term frequency normalization for polarity classification (Chapter 10, p. 97) and emotion detection (Chapter 11, p. 107).

PART III

APPLICATIONS

One of the important aspects of our work is its applicability to real world problems. In this part, we describe our participation in different evaluation campaigns in which we could test our proposed framework. Particularly, we have participated in the following tasks:

- SemEval'10: disambiguation of sentiment ambiguous adjectives in Chinese
- ROMIP'11: polarity classification in Russian product reviews
- I2B2'11: emotion detection in suicide notes

DISAMBIGUATING SENTIMENT AMBIGUOUS ADJECTIVES IN CHINESE

9

9.1 SemEval 2010 task description

Disambiguating sentiment ambiguous adjectives is a challenging task for NLP. Previous studies were mostly focused on word sense disambiguation rather than sentiment disambiguation. Although both problems look similar, the latter is more challenging in our opinion because impregnated with more subjectivity. In order to solve the task, one has to deal not only with the semantics of the context, but also with the psychological aspects of human perception of emotions from the written text.

The dataset of the SemEval 2010 task organized by Wu et al. (2010) consists of short texts in Chinese containing target adjectives whose sentiments need to be disambiguated in the given contexts. Table 9.1 lists the target adjectives. Table 9.2 lists an excerpt from the dataset translated into English using MT.

大	big
小	small
多	many
少	few
高	high
低	low
厚	thick
薄	thin
深	deep
淺	shallow
重	heavy
輕	light
巨大	huge
重大	grave

Table 9.1: List of target adjectives for the SemEval task

In our approach, we use Twitter microblogging platform to retrieve emotional messages and form two sets of texts: messages with

Text	**Polarity**
Yang Weize said that labor costs is **Low** in Suzhou, talent quality is very high, with a better business conditions	Positive
It is reported that the accident occurred, water fog big, visibility is very **low**. Two ships collided, the tonnage of the smaller ferry sank immediately, while others took advantage of a ferry boat fled the scene in fog	Negative
Hong Kong officials said that the Mainland ”individual visit“ travelers crime is very **low** Hong Kong SAR Government deficit 81.1 billion	Positive
(International) Iraqi Foreign Minister said that the morale of the Iraqi people is **high**	Positive
TV quality of series ”New Silk Road“ is **high**	Positive
the click rate of youtube video is **high**	Positive
hpv3917tu: why the cpu temperature is **high**?	Negative

Table 9.2: Automatically translated samples from the SemEval sample dataset. Target adjectives are highlighed in bold.

positive emotions and those with negative ones as we described in Chapter 6 (p. 51). After the dataset of emotional texts has been obtained, we build a classifier based on n-grams Naïve Bayes approach. We tested two approaches to build a sentiment classifier:

1. In the first one, we collected Chinese texts from Twitter and used them to train a classifier to annotate the test dataset.
2. In the second one, we used machine translator to translate the dataset from Chinese to English and annotated it using collected English texts from Twitter as the training data.

We used the second approach because we could collect more of English texts from Twitter than Chinese ones and we also wanted to test the impact of machine translation on performance of our polarity classifier. We have experimented with Google Translate[1] and Yahoo Babelfish[2].

1. **Google Translate** http://translate.google.com/

2. **Yahoo Babelfish** http://babelfish.yahoo.com/

9.2 Our approach to sentiment disambiguation

9.2.1 Corpus collection

Using Twitter API we collected a corpus of text posts and formed a dataset of two classes: positive sentiments and negative sentiments. We queried Twitter for two types of emoticons considering eastern and western types of emoticons[3]:

- Happy emoticons: `:-)`, `:)`, `^_^`, `^o^`, etc.
- Sad emoticons: :-(, :(, T_T, ;_;, etc.

We were able to obtain 7,800 Twitter posts in Chinese, and 200,000 posts in English evenly split between negative and positive classes.

The collected texts were processed as follows to obtain a set of n-grams:

1. Filtering – we remove URL links (e.g. http://example.com), Twitter user names (e.g. @alex – with symbol @ indicating a user name), Twitter special words (such as "RT"[4]), and emoticons.
2. Tokenization – we segment text by splitting it by spaces and punctuation marks, and form a bag of words. For English, we kept short forms as a single word: "don't", "I'll", "she'd".
3. Stopwords removal – in English, texts we removed articles ("a", "an", "the") from the bag of words.
4. N-grams construction – we make a set of n-grams out of consecutive words.

A negation particle is attached to a word which precedes it and follows it. For example, a sentence "I do not like fish" will form three bigrams: "I do+not", "do+not like", "not+like fish". In English, we considered negative particles 'no' and 'not'. In Chinese, we consid-

3. **Eastern emoticons** unlike western ones, represent facial expressions with a different style. While a western emoticon depicts a face rotated 90° (e.g. `:-P`), an eastern one depicts it horizontally with outer elements representing eyes and the middle elements representing the mouth. For example, the symbol `^` is used to show happy emotions (e.g. `^_^`), the `T` is used to represent tears, sad feelings (e.g. T_T).

4. An abbreviation for retweet, which means citation or re-posting of a message (see Chapter 6, p. 51)

ered the following particles:

1. 不 – is not + noun
2. 未 – does not + verb, will not + verb
3. 莫 (别) – do not (imperative)
4. 無 (沒有) – does not have

9.2.2 Classifier

We build a sentiment classifier using the multinomial Naïve Bayesean classifier which is based on Bayes' theorem.

$$P(s|M) = \frac{P(s) \cdot P(M|s)}{P(M)} \tag{9.1}$$

where s is a sentiment, M is a text. We assume that a target adjective has the same sentiment polarity as the whole text, because in general the lengths of the given texts are small.

Since we have sets of equal number of positive and negative messages, we simplify the equation:

$$P(s|M) \propto \frac{P(M|s)}{P(M)} \tag{9.2}$$

$$P(s|M) \propto P(M|s) \tag{9.3}$$

We train Bayesean classifiers which use a presence of an n-grams as a binary feature. We have experimented with unigrams, bigrams, and trigrams. Pang et al. (2002) reported that unigrams outperform bigrams when doing sentiment classification of movie reviews, but Dave et al. (2003) have obtained contrary results: bigrams and trigrams worked better for the product-review polarity classification. We tried to determine the best settings for our microblogging data. On the one hand high-order n-grams, such as trigrams, should capture patterns of sentiments expressions better. On the other hand, unigrams should provide a good coverage of the data. Therefore we combine three classifiers that are based on different n-gram orders (unigrams, bigrams and trigrams). We make an assumption of conditional independence of n-gram for the calculation simplicity:

$$P(s|M) \propto P(G1|s) \cdot P(G2|s) \cdot P(G3|s) \tag{9.4}$$

where $G1$ is a set of unigrams representing the message, $G2$ is a set of bigrams, and $G3$ is a set of trigrams. We assume that n-grams are conditionally independent:

$$P(Gn|s) = \prod_{g \in Gn} P(g|s) \tag{9.5}$$

Where Gn is a set of n-grams of an order n.

$$P(s|M) \propto \prod_{g \in G1} P(g|s) \cdot \prod_{g \in G2} P(g|s) \cdot \prod_{g \in G3} P(g|s) \tag{9.6}$$

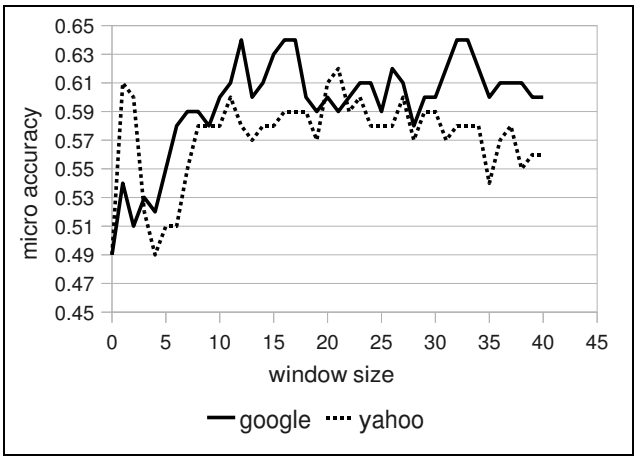


Figure 9.1: Micro accuracy when using Google Translate and Yahoo Babelfish

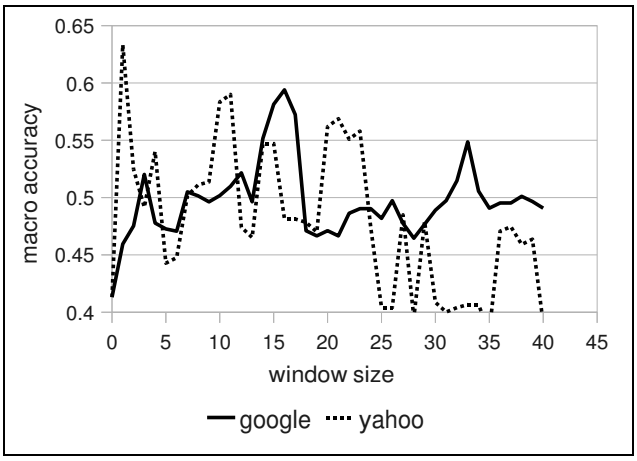


Figure 9.2: Macro accuracy when using Google Translate and Yahoo Babelfish

Finally, we calculate a log-likelihood of each sentiment:

$$L(s|M) = \sum_{g \in G1} log(P(g|s)) + \sum_{g \in G2} log(P(g|s)) + \sum_{g \in G3} log(P(g|s)) \tag{9.7}$$

In order to improve the accuracy, we changed the size of the context window, i.e. the number of words before and after the target adjective used for classification.

9.3 Experiments and results

In our experiments, we used two datasets: a trial dataset containing 100 sentences in Chinese and a test dataset with 2917 sentences. Both datasets were provided by the task organizers. Micro and macro accuracy were chosen as the evaluation measure.

First, we compare the performance of our method when using Google Translate and Yahoo Babelfish for translating the trial dataset. The results for micro and macro accuracy are shown in Figure 9.1 and 9.2 respectively. The X-axis represents the size of a context window equal to a number of words on both sides of the target adjective. The Y-axis shows accuracy values. From the obtained results, we observed that Google Translate provided better results, thus it was chosen when annotating the test dataset.

Next, we studied the impact of the context window size on micro and macro accuracy. The impact of the size of the context window on the accuracy of the classifier trained on Chinese texts is presented in Figure 9.3 and for the classifier trained on English texts with translated test dataset in Figure 9.4. The second approach achieves slightly better results: 64% of macro and 61% of micro accuracy vs. 63% of macro and 61% of micro accuracy when training on English texts.

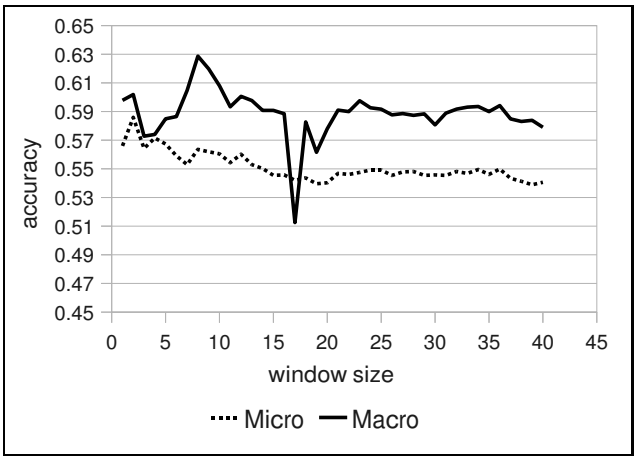


Figure 9.3: Micro and macro accuracy for the first approach (training on Chinese texts)

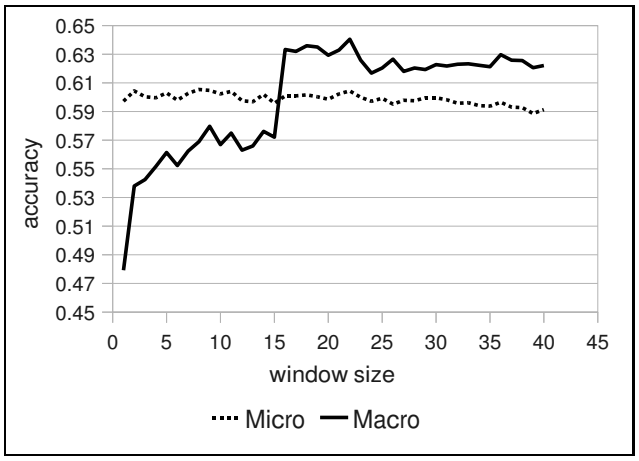


Figure 9.4: Micro and macro accuracy for the second approach (training on English texts which have been machine translated)

Chinese requires a smaller size of a context window to obtain the best performance. When training on Chinese texts, a window size of 8 words yields the best macro accuracy. For the second approach, training on English texts, we obtained the highest accuracy with a window size of 22 words.

9.4 Conclusion

In this chapter, we have described our system for disambiguating sentiments of adjectives in Chinese texts based on Naïve Bayesean classifier trained on English and Chinese datasets of opinionated messages extracted from Twitter. The techniques used in our approach can be applied to any other language provided that there is sufficient Twitter data. We were able to achieve up to 64% of macro and 61% of micro accuracy at the SemEval 2010 task which is lower than other participants' results (our rank is 6 out of 7), but our system is automatic and does not require any manually built lexicon.

POLARITY CLASSIFICATION OF RUSSIAN PRODUCTS REVIEWS

10

10.1 ROMIP 2011 task description

ROMIP is an annual evaluation campaign in information retrieval launched in 2002 by Dobrov et al. (2004). In ROMIP 2011, the organizers added the sentiment analysis track which aimed at classification of opinions in user generated content. A dataset composed of product reviews collected from a recommendation service Imhonet[1] and product aggregator service Yandex.Market[2] was provided to participants for training their systems. The dataset contained reviews about three topics: digital cameras, books, and movies. Table 10.1 shows the characteristics of the dataset.

Each review consists of a text of the review and meta information. Meta information contains a rating score assigned to a product, a product ID, a reviewer ID, and a review ID. Reviews from Yandex.Market also contain review creation time, usefulness of the review (assigned by other users), pros and cons of the product given by the review author. In our work, we used only a review text, a rating score, and pros/cons if available. The score is given on 1–5 scale for Imhonet reviews, and 1–10 scale for Yandex.Market reviews, where a higher value represents more positive opinion. Figure 10.2 shows an example of a digital camera review.

The evaluation dataset was not provided until the evaluation phase at the end of the campaign. The organizers have collected 16,861 posts from LiveJournal[3] blogging platform that mention books, movies, or cameras out of which 874 posts were annotated by two human experts. This track is somewhat different from other evaluation campaigns because the evaluation dataset was not of the same nature as the training data. First, the texts had different genres (product reviews vs. blog posts), and secondly the annotations were produced differently: the training data was composed automatically, while the test data was annotated manually. Figure 10.3 shows an example of a test document.

The track was divided into three subtracks:

- Polarity classification into two classes: negative/positive
- Polarity classification into three classes: negative/mixed/positive
- Polarity classification into five classes: a score on the scale 1–5,

1. **Imhonet**
http://imhonet.ru

2. **Yandex.Market**
http://market.yandex.ru

Topic	Source	# docs
Books	Imhonet	24,159
Movies	Imhonet	15,718
Cameras	Yandex. Market	10,370

Table 10.1: Characteristics of the training dataset

3. **LiveJournal**
http://livejournal.com

```
<row rowNumber="0">
  <value columnNumber="0">1328131</value>     <!-- review ID     -->
  <value columnNumber="1">926707</value>      <!-- product ID    -->
  <value columnNumber="2">48983640</value>    <!-- author ID     -->
  <value columnNumber="3">2009-05-03</value> <!-- creation time -->
  <value columnNumber="4">4</value>           <!-- rating        -->
  <value columnNumber="5">                    <!-- text          -->
    Хороший выбор для опытного фотолюбителя.
    <!-- A good choice for an experienced amateur photographer. -->
  </value>
  <value columnNumber="6">                    <!-- pros          -->
    Большой выбор режимов съемки,12-кратный оптический зум,
    естественная цветопередача,большой ЖК-экран.
    <!-- Large selection of shooting modes,12-times optical zoom,
    natural color, large LCD screen. -->
  </value>
  <value columnNumber="7">                    <!-- cons          -->
    Невысокая скорость подзарядки фотовспышки.
    <!-- The low speed of flash recharge. -->
  </value>
  <value columnNumber="8">0.59375</value>     <!-- usefulness    -->
</row>
```

Table 10.2: An example of a review from the training dataset. Russian text has been translated into English only for this example

where 1 represents an exclusively negative opinion, and 5 represents an exclusively positive opinion

In its turn, each subtrack had 3 runs by the number of topics: classification in each topic was evaluated separately, resulting in total 9 separate evaluations.

10.1.1 Task challenge

Sentiment analysis is a difficult task even for resource-rich languages (read, English). Along with simple language processing, such as part-of-speech (POS) tagging, more sophisticated NLP tools such as discourse parsers and lexical resources may be required by existing approaches. Thus, it is quite difficult to adapt methods that were developed in other languages (read, English) to Russian.

The ROMIP track poses additional challenges other than the difficulty of analysing sentiments in general. As mentioned before, the evaluation set was not constructed the same way as the training data. That makes it more difficult for statistical based approaches as the language model differs in two datasets. Moreover, the distribution of classes is also different. The training set contained more posi-

```
<?xml version="1.0" encoding="windows-1251"?>
<document>
  <ID>11347</ID>
  <link>http://vikilt.livejournal.com/12619.html</link>
  <date>2011-02-06T20:59:15Z</date>
  <object>
    Плохая училка
    <!-- Bad teacher -->
  </object>
  <text>
    Недавно посмотрел фильм "Очень плохая училка" и наконец,
    увидел этого самого Джастина Тимберлейка о котором так много
    было звона и сильно удивился. В фильме персонаж Кэмерон Диос
    как только видит этого Джастина начинает млеть и интенсивно
    намокать, хотя сам персонаж никаких эротический эмоций кроме смеха
    и недоумения не вызывает. Дальше он там, в фильме поёт песенку,
    которая тоже оставляет желать лучшего. Девушки, неужели вам
    действительно нравятся такие чахлые додики сомнительной наружности?
    <!-- Recently, I have watched a movie "Bad teacher" and finally,
    I've seen this Justin Timberlake about whom there have been
    so much buzz and I was surprised a lot. In the movie,
    the character of Cameron Diaz becomes excited as soon as
    she sees this Justin, although his character does not invoke
    any feelings except laughing. Next, he there, in the movie,
    sings a song, which is poor also. Girls, do you really
    like such doubtful looking nerds? -->
  </text>
</document>
```

Table 10.3: An example of a document from the evaluation set. Russian text has been translated into English only for this example

tive reviews, however the way the reviews were picked for annotation was unknown. Finally, the interpretation of rating also varies, as there were different conventions when assigning scoring products and when annotating the test set. In other words, a user of Yandex.Market may have a different interpretation of 3 stars assigned to a camera from a human annotator who rates a review. Multiclass classification was another challenge, since most of research on polarity classification consider it a binary problem, i.e. classifying a document into positive/negative classes.

One of few works on sentiment analysis in Russian by Pazelskaya and Solovyev (2011) used a manually constructed affective lexicon along with POS-tagging and lexical parsing information for a rule based polarity classifier. However, to our knowledge no publicly available affective resource exists in Russian, therefore a lexicon based approach would require to create a lexicon from scratch which is a

costly process. In order to tackle the problem, we have decided to use a language independent approach that does not require sophisticated NLP tools or lexical resources that are not available in Russian. We used an SVM based system with features based on n-grams, part-of-speech tags, and dependency parsing. For that we have trained a dependency parser on the Russian National Corpus [4]. Additionally, a study on terms weighting and corpus composition has been performed in order to optimize the performance of our system.

4. **Russian National Corpus**: http://www.ruscorpora.ru/en/

10.2 Our approach to polarity classification

We use the LIBLINEAR package developed by Fan et al. (2008). For the 2-class track we trained SVM in binary classification mode, for the 3 and 5-class tracks, we used a multiclass and regression modes.

10.2.1 Training dataset composition

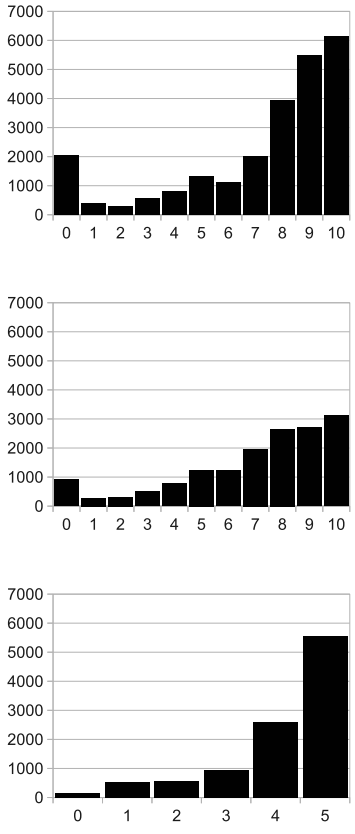


Figure 10.1: Score distribution in books (top), movies (middle), and cameras (bottom) datasets.

The distribution of opinion scores in the training dataset was highly unbalanced, which caused difficulties for training the model. Figure 10.1 shows distributions of reviews by scores in different topics. In general, positive reviews are prevailing in the training dataset which creates a bias towards a positive class. For the 2-class problem, we have decided to balance the training dataset by using an equal number of reviews of negative and positive opinions. Thus we considered books and movies reviews with scores 1–4 as negative and 9–10 as positive, and in the cameras collection, we considered reviews with scores 1–2 as negative and 5 as positive. The rest of the reviews were not included in the training. For 3-class and 5-class problems we left the dataset as is, because there would not be enough data to represent each class.

Another decision which had to be made, was whether to train three separate models for each topic or to combine all the data and to train one general model to classify reviews from each topic. We have experimented with both settings, and report the results in Section 10.3.

Reviews from Yandex.Market on cameras contain product pros and cons. To benefit from this additional information, we decided to include it in the text of the review. Thus, if a review is considered to be positive (using the criteria as mentioned above) then we add pros as the last phrase of the text. Otherwise, if a review is negative, we use cons. We have discovered that by doing this, we improved the accuracy of binary polarity classification up to 13.7%.

10.2.2 Feature vector construction

We have experimented with two types of features to build the model: traditional n-grams and our proposed d-grams.

N-grams

In the n-gram model, text is represented as a bag of words subsequences of a fixed size. We have experimented with unigrams and bigrams. Any non alphanumeric character was considered as a word boundary. Negations has been handled by attaching a negation particle (не – *no*, ни – *neither*, нет – *not*) to a preceding and a following word when constructing n-grams (§ 4.2.3, p. 23).

D-grams

D-grams are similar to n-grams, however, while n-grams are constructed by splitting a text into subsequences of consecutive words, d-grams are constructed from a dependency parse tree, where words are linked by syntactic relations. See Chapter 7 (p. 65) for a detailed description of d-grams.

To obtain dependency parse trees, we first applied TreeTagger adapted for Russian by Sharoff et al. (2008) for tokenization and POS-tagging. Next, we fed the tagged output to the MaltParser by Nivre et al. (2006) that we had trained on the Russian National Corpora.

Weighting scheme

We consider two weighting schemes which are used in sentiment analysis: binary and delta tf-idf. We augmented delta tf-idf formula with our proposed average term-frequency normalization that lowers importance of words that are frequently used in a document (Chapter 8, p. 75):

$$w(g_i) = \frac{\mathrm{tf}(g_i)}{\mathrm{avg.tf}(g_i)} \cdot \log \frac{\mathrm{df_p}(g_i) + 0.5}{\mathrm{df_n}(g_i) + 0.5} \tag{10.1}$$

10.3 Experiments and results

In this section, we report results obtained during the system development phase and the official results provided by the organizers of ROMIP. All the development results were obtained after performing 10-fold cross validation.

10.3.1 Development results

For the development phase, we present results only on binary classification as all the system parameters were tuned according to the results of these experiments. Table 10.4 shows results of n-gram based model with binary weights across different topics. According to previous research on domain-adaptation for sentiment analysis a model trained on the same topics as the test set performs better than one trained on another topic. However, we were interested whether combining all the training data thus increasing the size of the available

		Test data			
		books	movies	cameras	combined
Train data	books	76.0	74.0	65.5	73.4
	movies	77.3	76.4	66.4	74.5
	cameras	63.2	62.0	76.0	65.5
	combined	**78.4**	**78.9**	**77.1**	**78.6**

Table 10.4: Macro-averaged accuracy over different training and test data. Rows correspond to a dataset on which the model has been trained, columns correspond to test data. *Combined* is a combination of all three topics.

	Books				**Movies**				**Cameras**			
	div		com		div		com		div		com	
default	76.0		78.4		76.4		78.9		76.0		77.1	
+ balanced	78.1	+1.9	79.5	+0.9	76.3	−0.1	78.2	−0.7	77.4	+1.4	77.5	+0.4
+ pros/cons	78.1		79.6	+0.1	76.3		78.6	+0.4	91.8	+13.7	87.9	+10.4

Table 10.5: Performance gain when adding class balancing and including pros/cons.

	Books		**Movies**		**Cameras**	
	div	com	div	com	div	com
ngrams + binary	78.1	79.6	**76.3**	**78.6**	91.8	87.9
ngrams + Δtfidf	77.4	78.8	76.2	76.5	93.1	90.4
dgrams + binary	78.0	79.8	74.9	77.8	91.3	88.2
dgrams + Δtfidf	**78.4**	**80.2**	76.1	77.3	**93.6**	**91.3**

Table 10.6: Classification accuracy across different topics. For each topic, we evaluated a model trained on the same topic (div) and a model trained on all the reviews (com).

training data set improves the model. As we can see from the results, the model trained on the combined data performs better than a model trained only on one topic and the model trained on the same topic as the test set performs better than a model trained on another topic (§ 4.2.4, p. 24). However, we will see that it would change once we add additional information.

Table 10.5 shows how the performance changes after balancing the training data, and after adding pros and cons. Balancing the training set improves accuracy when classifying books and cameras and slightly degrades the performance on the movies collection. Adding pros and cons drastically improves the performance over the cameras test set (up to 13.7% of gain). Notice, also that the model trained only on the cameras collection performs much better than the one trained on combined data (91.8% vs. 87.9%). Thus, for the following experiments we keep these settings: balancing training set and including pros and cons.

	System ID	Mode	Features	Weights	Training set
2-class	2-dgram-delta-div	binary	d-grams	Δtfidf	divided
	2-dgram-delta-com	binary	d-grams	Δtfidf	combined
	2-ngram-delta-div	binary	n-grams	Δtfidf	divided
	2-ngram-delta-com	binary	n-grams	Δtfidf	combined
	2-ngram-bin-div	binary	n-grams	binary	divided
	2-ngram-bin-com	binary	n-grams	binary	combined
3-class	3-ngram-bin-div	multiclass	n-grams	binary	divided
	3-ngram-bin-com	multiclass	n-grams	binary	combined
	3-regr-ngram-bin-div	regression	n-grams	binary	divided
	3-regr-ngram-bin-com	regression	n-grams	binary	combined
5-class	5-ngram-bin-div	multiclass	n-grams	binary	divided
	5-ngram-bin-com	multiclass	n-grams	binary	combined
	5-regr-ngram-bin-div	regression	n-grams	binary	divided
	5-regr-ngram-bin-com	regression	n-grams	binary	combined

Table 10.7: Summary of the submitted systems

Table 10.6 shows the comparison of the model using different features and weighting schemes. Here we have compared the traditional n-grams model with our proposed d-grams features using the same weighting schemes (binary and delta tf-idf). As we observe from the results, d-grams with delta tf-idf yields better accuracy on books and cameras test sets, while n-grams with binary weights perform better on the movies collection. However, the difference is not very big.

10.3.2 Official results

According to the results we have obtained during the development phase, we have submitted the official runs on the unseen data. For 2-class track we have submitted 6 systems. For 3-class and 5-class tracks, we trained only systems based on n-grams due to time and resource constrains. For each of these tracks, we have submitted 4 systems. The summary of the submitted systems is presented in Table 10.8. The overall standings are depicted in Figures 10.2 – 10.4.

10.4 Conclusions

Sentiment analysis is a challenging task for computational linguistics. It becomes especially difficult for resource-poor languages. In this paper, we have described our participation in Russian sentiment analysis evaluation campaign ROMIP 2011. We have tested our language independent framework for polarity classification that is based on SVM with the traditional n-grams model and our proposed features based on dependency parse trees. The developed system was ranked 1^{st} in the 5-class track in all topics, 3^{rd} in the 3-class track in

	System ID	Books		Movies		Cameras	
		score	rank	score	rank	score	rank
2-class	2-dgram-delta-div	65.1	24/53	70.3	5/27	**81.7**	**11/25**
	2-dgram-delta-com	**66.1**	**23/53**	**70.9**	**3/27**	76.6	17/25
	2-ngram-delta-div	61.8	31/53	70.0	7/27	77.8	15/25
	2-ngram-delta-com	63.0	27/53	67.7	8/27	80.6	12/25
	2-ngram-bin-div	57.9	36/53	63.7	10/27	79.2	13/25
	2-ngram-bin-com	58.8	35/53	65.3	9/27	78.8	14/25
3-class	3-ngram-bin-div	**48.4**	**12/52**	47.7	9/21	55.7	8/15
	3-ngram-bin-com	49.9	18/52	**50.4**	**5/21**	**62.6**	**4/15**
	3-regr-ngram-bin-div	47.6	21/52	48.4	8/21	50.0	9/15
	3-regr-ngram-bin-com	48.8	16/52	49.8	6/21	57.4	7/15
5-class	5-ngram-bin-div	27.0	4/10	24.6	5/10	**34.2**	**1/10**
	5-ngram-bin-com	**29.1**	**1/10**	**28.6**	**1/10**	28.3	7/10
	5-regr-ngram-bin-div	28.5	3/10	26.6	3/10	31.1	4/10
	5-regr-ngram-bin-com	**29.1**	**1/10**	**28.6**	**1/10**	28.3	7/10

Table 10.8: Official ranking of the submitted systems

movies domain, and 4^{th} in the binary classification track in cameras domain according to the official evaluation measures.

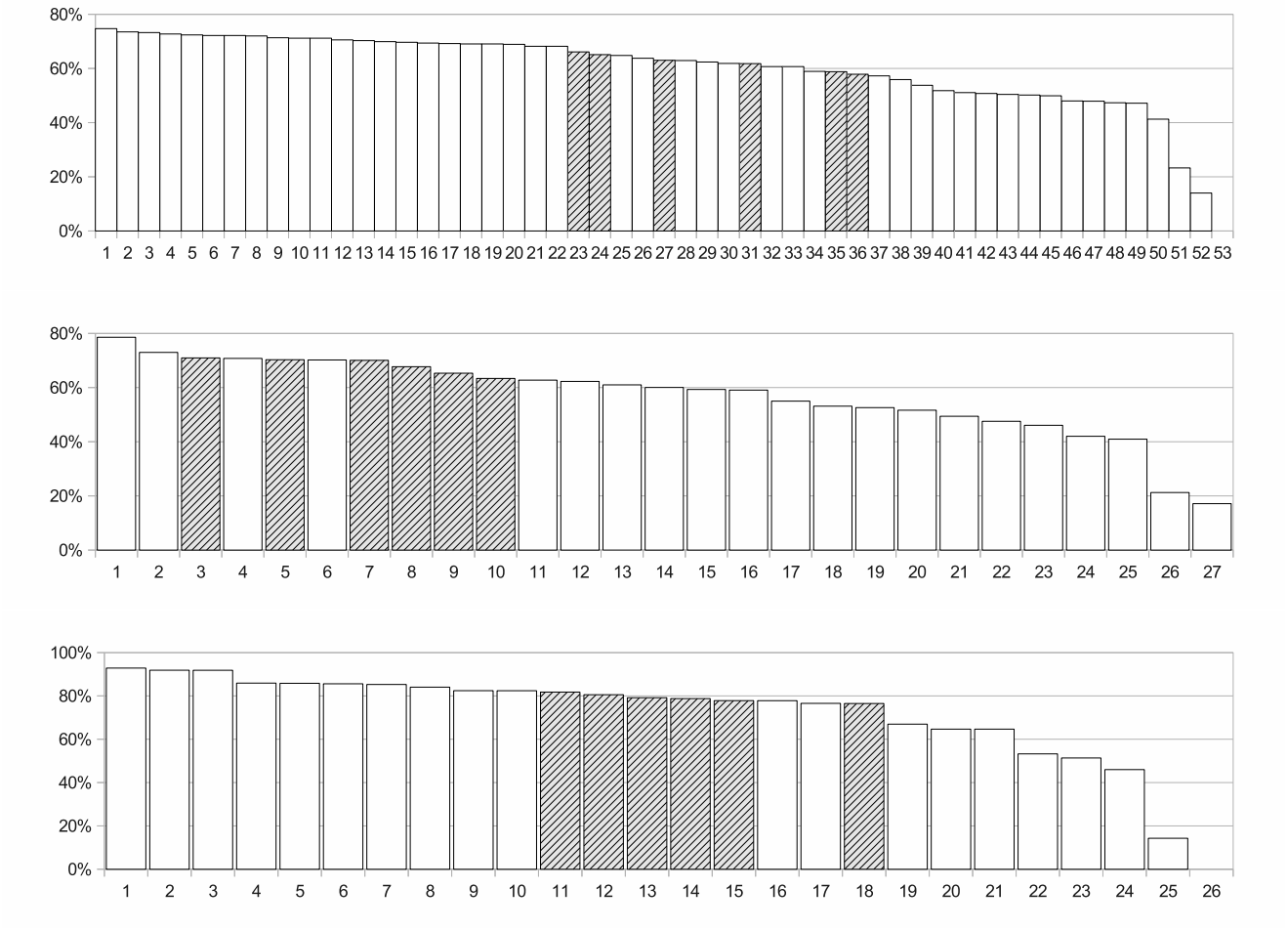


Figure 10.2: Systems performance and ranking on the 2-class track on books (top), movies (middle), and cameras (bottom) collections. Our systems are highlighted

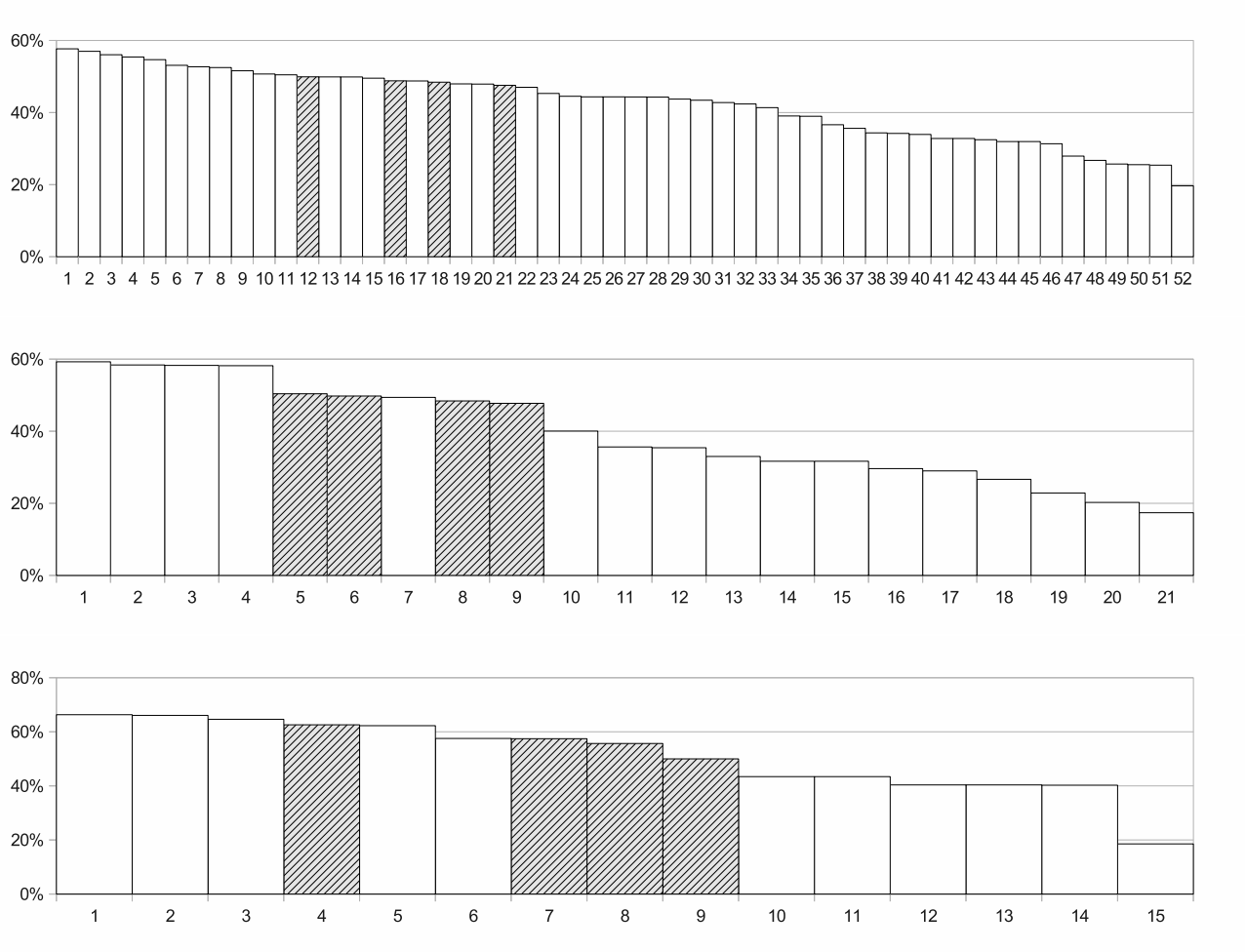


Figure 10.3: Systems performance and ranking on the 3-class track on books (top), movies (middle), and cameras (bottom) collections. Our systems are highlighted

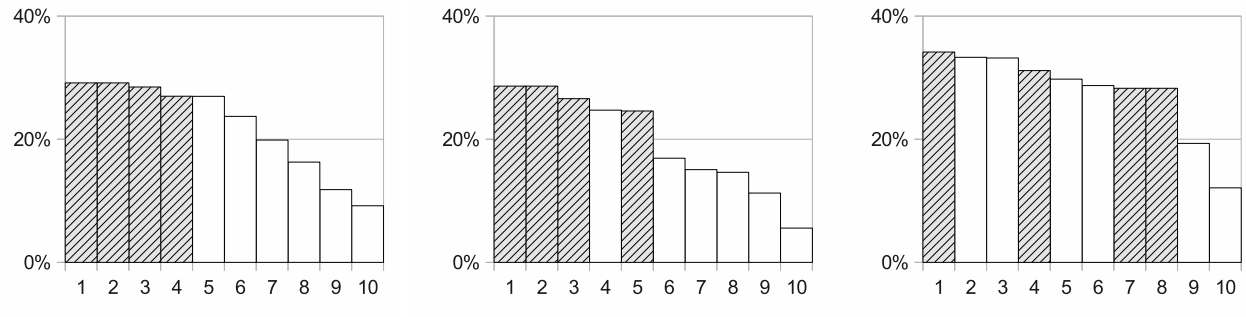


Figure 10.4: Systems performance and ranking on the 5-class track on books (top), movies (middle), and cameras (bottom) collections. Our systems are highlighted

EMOTION DETECTION IN SUICIDE NOTES

11

In this chapter, we present our participation in the second track of the i2b2/VA 2011 challenge, whose aim was the detection of emotions expressed in a corpus of suicide notes, provided by the organizers. After a short reminder of the challenge requirements and a description of the corpus, we present our natural language processing pipelines. We then report on the evaluation of the different approaches we have tried and discuss our results on the task.

11.1 I2B2 2011 task description

The second track of the i2b2 2011/VA challenge aims at identifying the opinion expressed in suicide notes by tagging sentences with one or several of the following fifteen categories: *instructions*, *information*, *hopelessness*, *guilt*, *blame*, *anger*, *sorrow*, *fear*, *abuse*, *love*, *thankfulness*, *hopefulness*, *happiness-peacefulness*, *pride*, *forgiveness*. In Table 11.1, we give the distribution of the annotation among the different categories. Note that the first two categories do not describe emotions but objective material. Sentences which do not fall into one of these categories have to be left untagged. The unique source of information provided to the participants is a training corpus, which has been hand-tagged.

Category	Train	Test
abuse	9	5
anger	69	26
blame	107	45
fear	25	13
forgiveness	6	8
guilt	208	117
happiness-peacefulness	25	16
hopefulness	47	38
hopelessness	455	229
information	295	104
instructions	820	382
love	296	201
pride	15	9
sorrow	51	34
thankfulness	94	45

Table 11.1: Number of annotations for each category in both training and test corpora

11.1.1 Corpus description

The training corpus consists of 600 suicide notes hand-annotated, while the test corpus is composed of 300 suicide notes. Those documents are of several kinds, mainly last will and testament. The corpus has been fully de-identified[1] (names, dates, address) and tokenized. Each document from the training corpus is very brief, on average: 7 sentences and 132.5 tokens (mainly words but also punctuation marks) per document. Proportions are similar for the test corpus. Documents include spelling errors (e.g. *conctract*, *poicies*). There are a few residual processing errors, more particularly the apostrophe in genitives and abbreviations, where spaces have been introduced (e.g. *could n't*, *Mary' s*) or the apostrophe replaced by a star with missing tokenization (e.g. *don*t*, *wasn*t*). Sentence segmentation is noisy (several short sentences are sometimes encoded as one single sentence). In the training corpus, 2,173 different sentences have been

1. Each name has been replaced by a generic name (*Jane, John, Mary*) and all addresses by the one of the Cincinnati Children's Hospital Medical Center.

```
INPUT FILE: 20080901735_0621.txt

John : I am going to tell you this at the last .
You and John and Mother are what I am thinking - I ca n't go on - my life is ruined .
I am ill and heart - broken .
Always I have felt alone and never more alone than now .
John .
Please God forgive me for all my wrong doing .
I am lost and frightened .
God help me ,
Bless my son and my mother .

OUTPUT FILE: 20080901735_0621.con.txt

c="You and John and Mother are what I am thinking - I can't go on - my
   life is ruined ." 2:0 2:21||e="hopelessness"
c="Always I have felt alone and never more alone than now ." 4:0 4:11||
e="sorrow"
c="I am lost and frightened ." 7:0 7:5||e="fear"
```

Table 11.2: Annotated example from the test corpus

# of lines	Train	Test
0	2076	790
1	1862	941
2	267	134
3	27	15
4	7	2
5	2	1

Table 11.3: Number of lines without annotation, with a single annotation or with several annotations in both training and test corpora

hand-annotated, among them 302 sentences received several category labels (see Table 11.3). Figure 11.2 shows an example of annotation from the test corpus with its reference annotation.

Lines with several annotated emotions are long sentences: the two lines composed of five emotions are between 73 and 82 tokens long. As an example, the longest line has been annotated with the five following emotions classes: *abuse*, *blame*, *guilt*, *hopelessness*, and *love*:

My Dearest Son Bill : Please forgive mother for taking this way out of my umbearable trouble with your Dad Smith - Son I 've loved you and Dad beyond words and have suffered the tortures of hell for Smith but his lies and misconduct to me as a wife is more than I can shoulder any more - Son God has been good to you and mother and please be big and just know that God needs me in rest.

We have found the task to be difficult for the following reasons.

- **Multiple labels per sentence.** In the following example, the two labels *hopelessness* and *instructions*: were provided by the annotators:

  In case of sudden death , I wish to have the City of Cincinnati burn my remains with the least publicity as possible as I am just a sick old man

and rest is what I want .

Multiple labeling makes the task more difficult for machine learning classifiers that normally work with a single label per sample.

- **No annotation.** When no annotation was assigned to a sentence, two interpretations are possible: either there is no emotion expressed, or there was a disagreement between the annotators. Here is an example, where a note could have been annotated with the *love*, but was left without annotation:

  ```
  I love you all, but I can't continue to be a burden to you.
  ```

  The ambiguous "no annotation" assumption adds noise to the training data.

- **Fine grained labels.** Certain labels have very close meanings and are consequently hard to distinguish from one another. As an example, *information* vs. *instructions*, *guilt* vs. *forgiveness*, or *sorrow* vs. *hopelessness* .

- **Unbalanced distribution of labels.** Certain labels in the training (and test) set appear much more frequently than others. The most frequent label *instructions* appears 820 times in the training set, while the label *forgiveness* appears only 6 times. This makes it all the more difficult to learn rare classes, due to possible biases during the training.

- **Lack of additional training data.** The task organizers provided the training corpus, however it is extremely difficult to find additional training material. To our knowledge, there is no publicly available text corpora of suicide letters or other similar resources. Construction of such a corpus is also problematic due to the nature of the task and lack of information about the guidelines used by the annotators.

11.2 Related textual analysis of suicide notes

One of the earliest approaches for automatic analysis of suicide notes was described by Stone and Hunt (1963). They have used a system called General Inquirer created at IBM to detect fake suicide notes. The core of the General Inquirer system is a dictionary containing 11,789 senses of 8,641 English words (i.e. certain words have several senses), each mapped to one or more of 182 categories, such as "positive", "negative", "self", "family", etc. The authors used the distribution of categories to distinguish between simulated and genuine suicide notes. The evaluation, using 33 simulated notes and 33 real notes, showed that the General Inquirer system was able to correctly identify 17 out of 18 test note pairs, which is a better performance than the one of random classification.

A more recent work by Pestian et al. (2010) used features ex-

tracted from the text of the notes to train different machine learning classifiers. The features were: number of sentences, word distribution statistics, distribution of part-of-speech tags, readability scores, emotional words and phrases. The performance of machine learning models were compared against the judgments of psychiatric trainees and mental health professionals. Experimental evaluations showed that the best machine learning algorithms accurately classified 78% of the notes, while the best accuracy obtained by the human judges was 63%.

To our knowledge, there is no published research on automatic emotion detection in suicide notes or similar topics.

11.3 Our approach to emotion detection

In order answer the challenge, we created a system that uses both a machine learning approach and hand-written rules to detect emotions. Our intention was to create a high-precision rule-based system backed up by a machine learning algorithm to improve recall and to generalize on unknown data.

11.3.1 Machine learning based approach

In our machine learning based approach, we trained an SVM classifier using different features extracted from the training set. We used the LIBLINEAR package by Fan et al. (2008) with a linear kernel and default settings. In order to perform multi-label classification, we employed the one-versus-all strategy, i.e., we trained an SVM classifier for each emotion independently. Each classifier provides a decision whether a given sentence contains the emotion it was trained to recognize or not. Such a setting allows us to have multiple labels per line or no labels at all, when all the classifiers returned a negative answer.

Here is a list of features that we have used to build our classification model:

- **N-grams**. N-gram models are widely used as a common approach for representing text in information retrieval, text categorization, and sentiment analysis. We used unigrams and bigrams, with normalized binary weights, such that for a given text T represented as a set of terms:

$$\mathrm{T} = \{t_1, t_2, \ldots, t_k\} \tag{11.1}$$

  we define the feature vector of T as

$$\mathrm{TF} = \{\frac{1}{\mathrm{avgtf}(t_1)}, \ldots, \frac{1}{\mathrm{avgtf}(t_k)}\} \tag{11.2}$$

  where avg.tf(t_i) is a normalization function based on average term frequency (Chapter 8, p. 75):

$$\mathrm{avg.tf}(t_i) = \frac{\sum_{\forall T, t_i \in T} \mathrm{tf}(\mathrm{t_i})}{|\forall T, t_i \in T|} \tag{11.3}$$

A procedure of attachment of the negation particle was performed to capture the negations, i.e., particles *no* and *not* were attached to a following word when generating n-grams.

- **POS-tags**. We use the TreeTagger by Schmid (1994) to obtain part-of-speech tags for words and also to perform sentence segmentation as some lines contain multiple sentences. To construct a feature vector, we used the frequencies of tags in a sentence. The important information provided by tags features are: the usage of auxiliary verbs, verb properties (tense, person, voice, mood), usage of adjectives and adverbs and their comparative or superlative forms, usage of cardinal numbers (important for distinguishing informative classes), and punctuations (such as the symbol $). We have shown in Chapter 6 (p. 51) that the distribution of POS-tags is different in subjective and objective texts, and texts with positive and negative polarities.
- **General Inquirer**. We use the dictionary from the General Inquirer (GI) system to create supplementary features as follows. Each word from a tested sample was lemmatized if possible. The lemma was searched in the GI dictionary and if found, all the associated categories were added to the bag of categories. Next, for each of the 182 GI categories, we counted the occurrences within the sentence. We got a 182-length feature vector. No disambiguation was done at this point. If multiple senses existed in the dictionary for a given lemma, all the categories associated with the senses were added to the bag.
- **ANEW**. In order to capture the mood of a text, we use the Affective Norms of English Words lexicon (§ 5.1.1, p. 37). The lexicon contains 1,034 English words with associated numerical scores of valence, arousal, and control. To construct a feature vector, we represented each word from ANEW in a 3-dimensional space, where each dimension represents a word's score. Next, we divided this space equally into N^3 buckets and counted the number of words from a sentence that fall into each bucket. The scores in ANEW dataset take a value between 1 and 9, thus all the words may have coordinates starting from (1, 1, 1) to (9, 9, 9). For example, we set $4^3 = 64$ buckets. Then, the first bucket would contain words with coordinates from (1, 1, 1) to (3, 3, 3), the second bucket: from (1, 1, 3) to (3, 3, 5) etc. Thus, we would obtain a 64-length feature vector.
- **D-grams**. We extracted subgraphs from the sentence dependency trees produced by the Stanford Lexical parser by de Marneffe et al. (2006) to produce d-grams as described in Chapter 7 (p. 65).
- **Heuristic features**. Finally, we added a number of heuristically produced features: the position of the sentence with respect to the beginning of the note, the presence of the following words in the sentence: "god", "thank", "please", "car", and "Cincinnati".

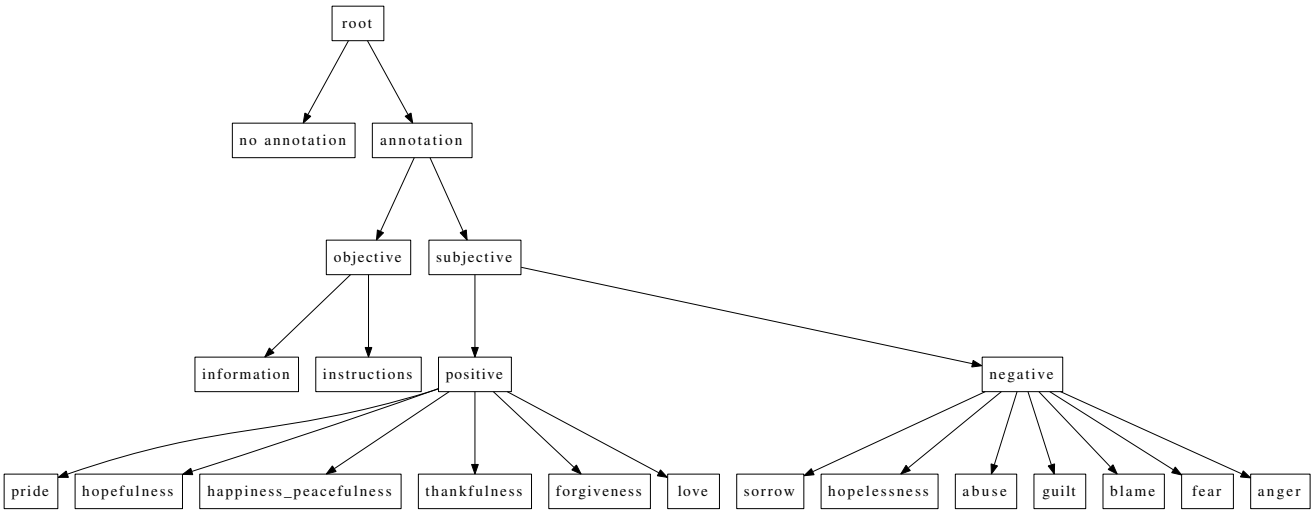


Figure 11.1: Emotions hierarchy

On different stages of classification, we used different combinations of the listed features. In order to combine features, we simply concatenated the produced feature vectors.

Yang and Lee (2009) have shown that hierarchical classifiers yield better results than flat ones, when classifying emotions. We have organized the labels into a hierarchy as shown in Figure 11.1.

Our final algorithms is as follows:

1. First, we have trained an annotation detector to distinguish sentences with annotations from unannotated ones. Features used: POS-tags, General Inquirer.
2. Next, the sentences considered to have annotations were fed to a subjectivity detector, to separate subjective sentences from objective ones. Features used: heuristic, POS-tags, General Inquirer.
3. Objective sentences were then classified between: *information* and *instructions*. Features used: unigrams, bigrams, General Inquirer, dependency graphs.
4. Subjective sentences were divided into emotions with a positive polarity and the ones with a negative polarity, using a polarity classifier. Features used: POS-tags, ANEW.
5. Sentences with a negative polarity were further classified according to 7 classes: *sorrow*, *hopelessness*, *abuse*, *guilt*, *blame*, *fear*, *anger*. Features used: unigrams, bigrams, d-grams, General Inquirer.
6. Sentences with a positive polarity were further classified among 6 classes: *pride*, *hopefulness*, *love*, *happiness/peacefulness*, *thankfulness*, *forgiveness*. Features used: unigrams, bigrams, d-grams, General Inquirer.

In order to estimate the task difficulty, we have plotted the data on a 2-dimension graph using principal component analysis for dimen-

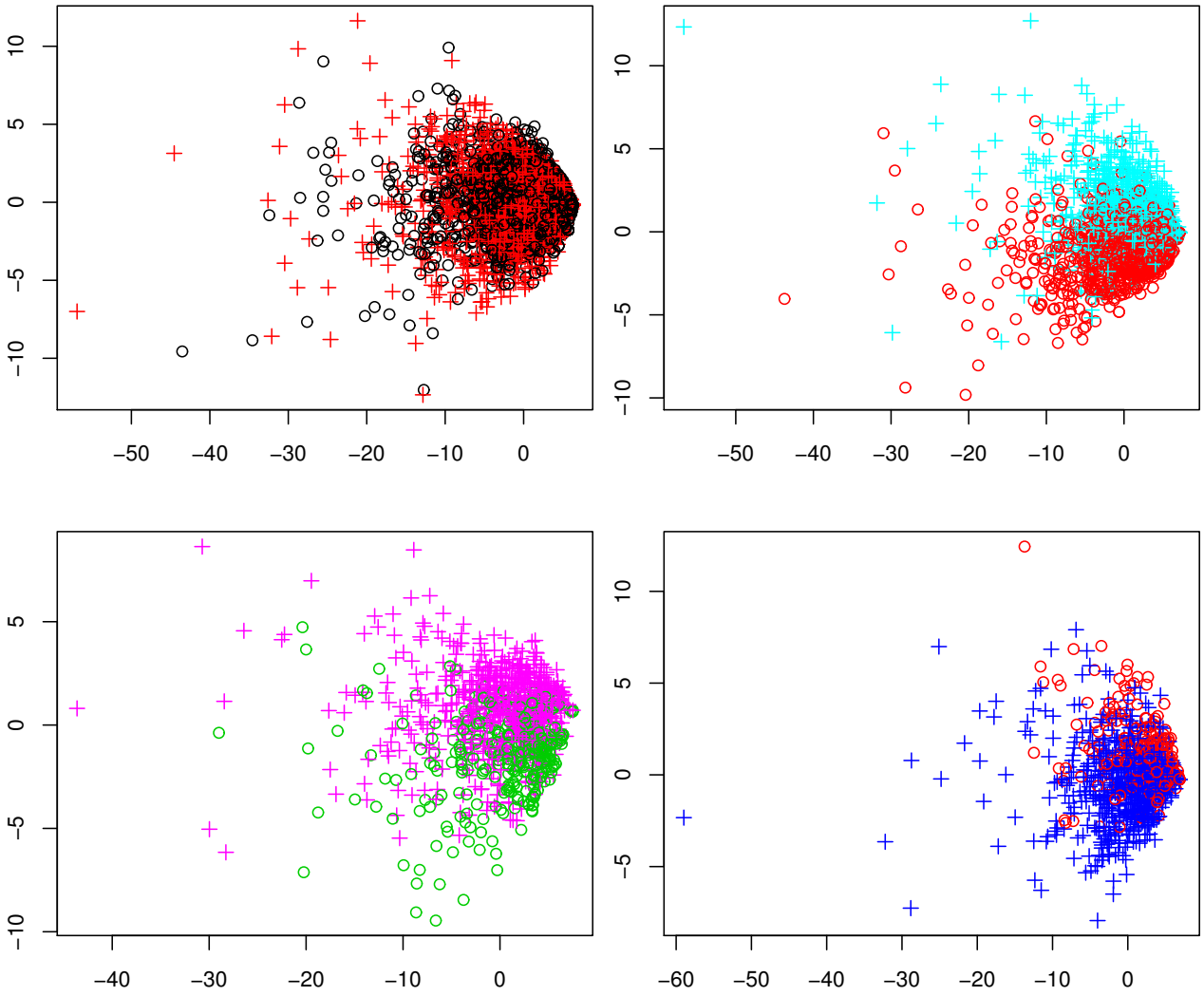


Figure 11.2: Visualizing samples in 2-dimensions: annotated (red crosses) vs. not annotated (black circles); subjective (blue crosses) vs. objective (red circles); positive (violet crosses) vs. negative (green circles); information (blue crosses) vs. instructions (red circles)

sion reduction and General Inquirer features as shown in Figure 11.2. As we can see from the figures, it is very difficult to separate annotated samples from unannotated ones. The distinction between subjective/objective and negative/positive emotions is much easier. Finally, *information* and *instructions* classes are less distinguishable.

11.3.2 Emotion detection using transducers

We also used an approach based on extraction patterns to identify emotions in suicide notes. Given the limited amount of training data and the number of target classes, we chose to define these patterns manually, rather than trying to identify them automatically. These patterns combine surface-level tokens, lemmas and POS (part-of-speech) tags and are detected in texts using finite-state transducers,

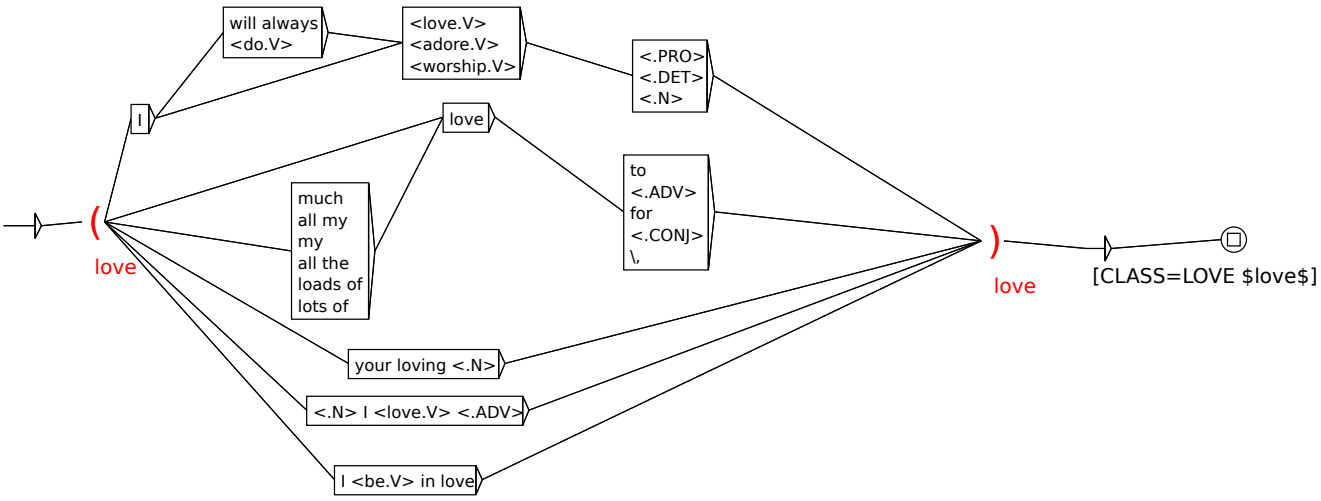


Figure 11.3: Example transducer for the emotion class `love`

which automatically tag pattern occurrences in the input text.

We have manually developed one transducer for each class using Unitex[2] by Paumier (2011) which provides also with its base configuration a tokenizer, a POS tagger and a lemmatizer. The transducers were created by careful investigation of the training corpus. For instance, the transducer built for the *love* category is shown in Figure 11.3. It can identify expressions such as *I will always love you*, or *your loving husband*.

Each valid path in the graph represents an emotion-specific pattern, which is subsequently marked in the input text. Nodes in the transducer may correspond to sequences of surface tokens, lemmas with a given POS (e.g. *love.V* for the verb "to love" and all its inflected forms) or POS tags (e.g., *ADV* for any adverb). As a consequence, the transducer is able to identify surface variants of the same pattern.

For the final classification, we applied all the transducers in a cascade, one after the other, in a specific order. A sentence is labeled with a given category if at least one expression has been recognized by the corresponding transducer.

2. **Unitex**
a corpus processing system, based on automata-oriented technology
http://www-igm.univ-mlv.fr/~unitex/

11.4 Experiments and results

In order to tune the system parameters of the machine learning component, we performed 10-fold cross validation on the training corpus. The task official performance measures are: micro-average precision, recall, F1-score. For our own purposes, we also calculated precision, recall, F1-score for each emotion category.

First, we analyzed the performance of the features used for emotion detection: GI, d-grams, unigrams, and bigrams. Figure 11.4 plots the classification F-measure of each emotion category and each feature using a flat classification scheme. The classification performance of more frequent classes is higher than those of rarer ones: *love*, *thank-*

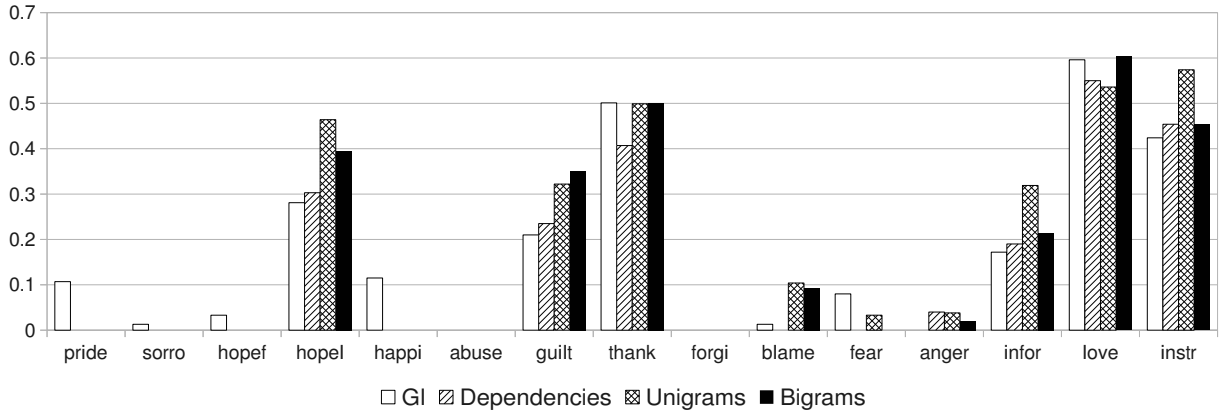


Figure 11.4: Performance of different features used for emotion detection across the classes

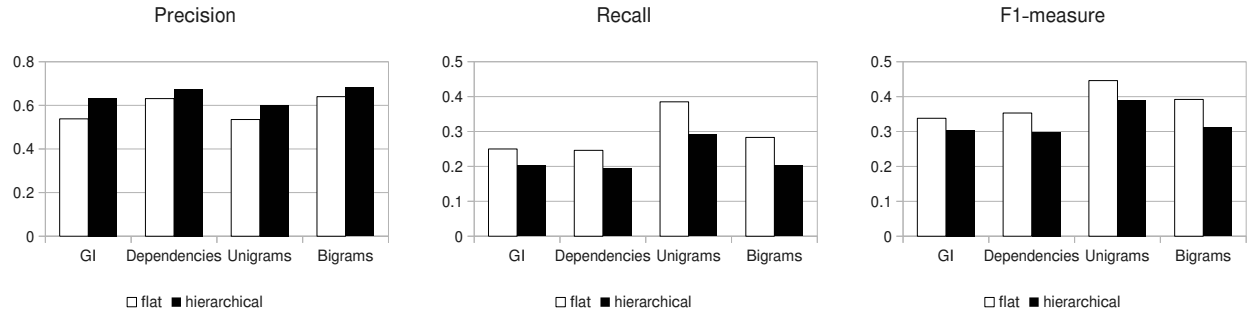


Figure 11.5: Hierarchical vs. flat classification performance (precision, recall and F-score)

fulness, *hopelessness*, and *guilt* are much better classified than *blame*, *fear*, and *anger*. Moreover, *pride*, *sorrow*, *hopefulness*, and *happiness* could be only detected with GI features, yet the performance is good. *Abuse* and *forgiveness* – the most rare classes in the corpus—are not detected by any features. As aforementioned, *information* and *instructions* classes are hardly distinguishable, which explains the low classification performance of the *information* and *instructions* classes, even though the later is the most frequent.

When performing hierarchical classification, we achieved 71% of accuracy on annotation detection, 84% on subjectivity detection, and 85% on polarity classification. The effect of the hierarchical classification is depicted on Figure 11.5. Micro-average precision, recall, F-score are presented for each feature. We can observe that precision augments when using hierarchical classification, but F-score drops due to the decrease of recall. To compensate this, we decided to use hierarchical classification with the mentioned features, but we added another classifier based on combination of unigrams and bigrams, which does a flat classification across all classes.

The final classification system consists of the rule-based component and the machine learning based one. We present the classifi-

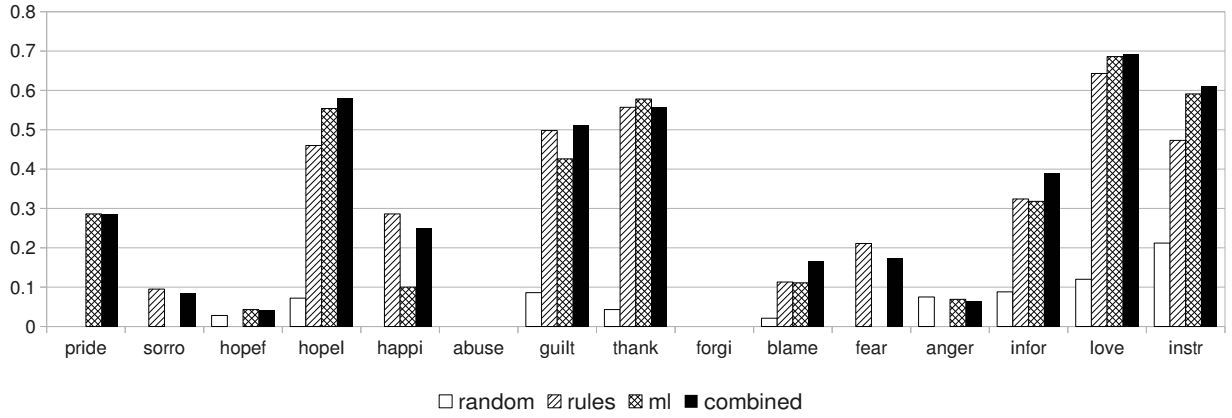
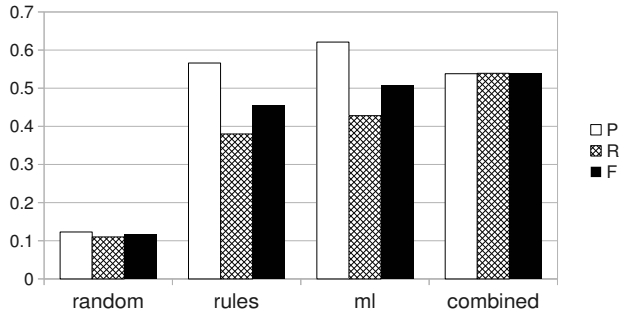


Figure 11.6: Performance of a random, rule-based, machine learning, and combined systems across the classes



	Precision	Recall	F1
random	0.123	0.110	0.116
rules	0.566	0.380	0.455
ml	0.621	0.428	0.507
combined	0.538	0.539	0.538

Figure 11.7: Micro-average performance of a random, rule-based, machine learning, and combined systems

cation performance of rule-based, machine learning, and the combination of both systems on the evaluation set in Figure 11.6 (across the classes) and in Figure 11.7 (micro-average). A baseline random classifier was added for a comparison.

Official evaluations results. The rule-based approach has obtained F-score = 0.4545, precision = 0.5662, recall = 0.3797, while the combined approach in the best run obtained F-score = 0.5383, precision = 0.5381, recall = 0.5385, and was ranked 6^{th} out of 26 participating teams. The official F1 mean score was 0.4875, min = 0.2967, max = 0.6139.

11.5 Conclusion

The emotion detection track of i2b2 2011 was a difficult task due to the nature of the data and the specificity of the annotation schema. We have developed a system combining two approaches for emotion detection and classification: machine learning and rule-based. Our best run obtained 0.5383 of F-measure, which is higher than the official mean score 0.4875 and was ranked 6^{th} among 26 participants.

PART IV

SUMMARY

CONCLUSION

12

When analysing sentiments in text, researchers have to deal with many issues, such as discourse analysis, coreference, figurative language, and others. In this work, we focused on polarity classification, a basic task of sentiment analysis which aimed at classification of user attitude towards an opinion target. Even with simplified settings, it is still a challenging task which becomes much more difficult if we have to deal with a multidomain and multilingual environment. Our approach aims at creating an automatic and adaptive sentiment analysis system which does not rely on domain or language specific resources. Our contribution is as follows:

- We have shown how the use of microblogging as a multilingual data source for opinion mining and sentiment analysis. We used Twitter to collect a dataset of opinionated messages.
- We have demonstrated an automatic way of labeling Twitter messages as positive or negative based on emoticons which were used as noisy labels. We obtained a set of positive and negative messages in four languages: 5,2 million messages in English, 672,800 in Spanish, 287,400 in French, and 7800 in Chinese. For English, we obtained an additional set of 100,000 neutral messages from tweets posted by journals and newspapers.
- We have performed a linguistic analysis of the collected dataset and observed that part-of-speech distribution differs between subjective and objective sets, and also between positive and negative ones. We proposed that POS distribution provided additional features for polarity classification and subjectivity detection.
- We have proposed a method for automatic construction of affective lexicons from Twitter. We constructed affective lexicons for English, Spanish, and French which were evaluated by checking a correlation with the ANEW dataset.
- We have applied the constructed lexicon to polarity classification of video game reviews in French from the DOXA project. Evaluation results showed that the performance of our approach is comparable to a supervised machine learning solution based on n-grams while our approach does not require an additional training data apart from Twitter.
- We have proposed a novel text representation model for senti-

ment analysis which we call d-grams and is based on dependency parse trees. Experimental evaluations performed with three different dependency parsers over a crossdomain dataset of product reviews in English and French shown that our model yields better polarity classification accuracy (up to 4.4%) than the traditional n-gram text representation.

- We have demonstrated a common problem of traditional supervised polarity classification approach when dealing with minor reviews. We showed that classifiers tend to rely on entity-specific features and as a result become biased towards the majority of opinions about a specific entity.
- We have proposed two measures: average term frequency and entity proportion for normalizing feature weights. Experimental evaluations on two English datasets with 5 different domains (movies, books, DVD, kitchen appliance, electronics) showed the improvement of polarity classification of minor reviews up to 12.5%.

The proposed framework for automatic and adaptive sentiment analysis has been successfully tested in following international evaluation campaigns:

- ROMIP'11: polarity classification in Russian product reviews. Our systems were ranked 1^{st} in the 5-class track in all topics, 3^{rd} in the 3-class track in movies domain, and 4^{th} in the binary classification track in cameras domain according to the official evaluation measures among a total of 20 participants.
- SemEval'10: disambiguation of sentiment ambiguous adjectives in Chinese. With our language independent approach, we obtained 64% of macro and 61% of micro accuracy and ranked 6^{th} among 7 participants.
- I2B2'11: emotion detection in suicide notes. Our system was ranked 6^{th} among 26 participants with F-measure 0.5383 which was much higher than the official mean score 0.4875.

13 FUTURE WORK

In our work, we have focused solely on polarity classification. We believe that it is necessary to solve this problem together with other tasks of sentiment analysis in order to be able to treat complex sentiment expressions. We consider the following framework for the perspective future work:

1. It is necessary to split the text into segments (paragraphs, sentences, or phrases) and analyze them individually rather than applying the bag-of-words approach disregarding word order and relations between them.
2. For each segment, we should determine its polarity, opinion topic and opinion holder.
3. We need to determine discourse relations between segments such as contradiction or agreement.
4. A probabilistic graphic model is trained taking into account links between text segments, type of links (discourse relations) and node information (opinion)

AUTHORS' PUBLICATIONS

14

14.1 International Journals

- Alexander Pak, Delphine Bernhard, Patrick Paroubek, Cyril Grouin, "A Combined Approach to Emotion Detection in Suicide Notes" to appear in *Biomedical Informatics Insights*
- Alexander Pak and Chin-Wan Chung, "A Wikipedia Matching Approach to Contextual Advertising" *World Wide Web*, 2009. Volume 13, Number 3, 251–274, DOI: 10.1007/s11280-010-0084-2.

14.2 Domestic Journals

- Alexander Pak and Patrick Paroubek, "Le microblogage pour la microanalyse des sentiments et des opinions" *Traitement Automatique des Langues*, 2010, vol. 51.3 (pp 75–100).

14.3 International conferences

- Alexander Pak and Patrick Paroubek, "Normalization of Term Weights for Sentiment Analysis" *LTC 2011. Language and Technology Conference*, Poznan, Poland: 2011
- Alexander Pak and Patrick Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining", *LREC 2010. Seventh International Conference on Language Resources and Evaluation*, Valetta, Malta: 2010. -6p
- Patrick Paroubek, Alexander Pak, Djamel Mostefa, "Annotations for opinion mining evaluation in the industrial context of the DOXA project", *LREC 2010. Seventh International Conference on Language Resources and Evaluation*, Valetta, Malta: 2010. -6p
- Alexander Pak, "Using Wikipedia to Improve Precision of Contextual Advertising", *LTC 2009. Lecture Notes in Computer Science*, 2009, Volume 6562/2011, 533-543, DOI: 10.1007/978-3-642-20095-3_49
  Best Student Paper award

14.4 Domestic conferences

- Alexander Pak and Patrick Paroubek, "Language Independent Approach to Sentiment Analysis (LIMSI Participation in ROMIP'11)" to appear in *Dialogue 2012*, Moscow, Russia
- Alexander Pak and Patrick Paroubek, "Sentiment Polarity Classification using Dependency Tree Subgraphs Text Representation" *TALN 2011. 18ème Conférence sur le Traitement Automatique des Langues Naturelles*, Montpelier, France: 2011. -6p
- Alexander Pak and Patrick Paroubek, "Constructing French Affective Lexicon using Twitter", *TALN 2010. 17ème Conférence sur le Traitement Automatique des Langues Naturelles*, Montréal, Canada : 2010. - 6p

14.5 Book chapters

- Alexander Pak and Patrick Paroubek, "Extracting Sentiment Patterns from Syntactic Graphs" to appear in *Social Media Mining and Social Network Analysis: Emerging Research*, IGI Global, 2012.

14.6 International workshops

- Alexander Pak, Delphine Bernhard, Patrick Paroubek, Cyril Grouin, "Emotion Detection in Clinical Reports. The LIMSI participation in the i2b2 2011/VA Challenge" to appear at I2B2, AMIA 2011
- Alexander Pak and Patrick Paroubek, "Twitter for Sentiment Analysis: When Language Resources are Not Available," DEXA, 2011 *22nd International Workshop on Database and Expert Systems Applications*, Toulouse, France: pp.111–115
- Alexander Pak and Patrick Paroubek. 2010. "Twitter Based System: Using Twitter for Disambiguating Sentiment Ambiguous Adjectives". *In Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval '10). Association for Computational Linguistics*, Uppsala, Sweden: 436–439
- Alexander Pak and Patrick Paroubek, "Text Representation using Dependency Tree Subgraphs for Sentiment Analysis", *DASFAA 2011. The 16th International Conference on Database Systems for Advanced Applications. In proceedings of Database Systems for Adanced Applications*, Xu J.,Yu G., Zhou S., Unland R., Lecture Notes in Computer Science vol 6637, Hong Kong, China: 2011. 323–332

14.7 Talks

- Alexander Pak, "Adaptive Sentiment Analysis" at Clunch 2011, University of Pennsylvania, Philadelphia, PA, USA.

BIBLIOGRAPHY

Abrams, M.H. and Harpham, G.G. 2011. *A Glossary of Literary Terms*. Cengage Learning. isbn 9780495898023. URL http://books.google.fr/books?id=SUEtEa9nUWQC. Cited on p. 15.

Arora, Shilpa; Mayfield, Elijah; Penstein-Rosé, Carolyn; and Nyberg, Eric. 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, pp. 131–139. Association for Computational Linguistics, Morristown, NJ, USA. URL http://portal.acm.org/citation.cfm?id=1860631.1860647. Cited on pp. 44 and 66.

Baccianella, Stefano; Esuli, Andrea; and Sebastiani, Fabrizio. may 2010. *SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining*. In Chair), Nicoletta Calzolari (Conference; Choukri, Khalid; Maegaard, Bente; Mariani, Joseph; Odijk, Jan; Piperidis, Stelios; Rosner, Mike; and Tapias, Daniel, eds., Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). European Language Resources Association (ELRA), Valletta, Malta. isbn 2-9517408-6-7. Cited on p. 40.

Bednarek, Monika. 2009. *Dimensions of evaluation: cognitive and linguistic perspectives.* In Pragmatics Cognition, pp. 146–175. Cited on p. 12.

Blitzer, John; Dredze, Mark; and Pereira, Fernando. Jun. 2007. *Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification*. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 440–447. Association for Computational Linguistics, Prague, Czech Republic. URL http://www.aclweb.org/anthology-new/P/P07/P07-1056.bib. Cited on pp. ix, 25, 75, and 76.

Blitzer, John; McDonald, Ryan; and Pereira, Fernando. 2006. *Domain adaptation with structural correspondence learning*. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP '06, pp. 120–128. Association for Computational Linguistics, Stroudsburg, PA, USA. isbn 1-932432-73-6. URL http://dl.acm.org/citation.cfm?id=1610075.1610094. Cited on p. 25.

Bradley, M M and Lang, P J. 1999. *Affective norms for English words (ANEW). Gainesville, FL. The NIMH Center for the Study of Emotion and Attention*. In University of Florida. Cited on p. 37.

Buvet, Pierre-André; Girardin, Chantal; Gross, Gaston; and Groud, Claudette. 2005. *Les prédicats d'<affect>*. In Reuve de linguistique et de didactique des langues, pp. 123–143. Cited on p. 37.

Candito, Marie; Nivre, Joakim; Denis, Pascal; and Anguiano, Enrique Henestroza. 2010. *Benchmarking of statistical dependency parsers for French*. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING '10, pp. 108–116. Association for Computational Linguistics, Stroudsburg, PA, USA. URL http://dl.acm.org/citation.cfm?id=1944566.1944579. Cited on p. 70.

Carvalho, Paula; Sarmento, Luís; Silva, Mário J.; and de Oliveira, Eugénio. 2009. *Clues for detecting irony in user-generated contents: oh...!! it's "so easy" ;-)*. In Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion, TSA '09, pp. 53–56. ACM, New York, NY, USA. isbn 978-1-60558-805-6. doi: http://doi.acm.org/10.1145/1651461.1651471. URL http://doi.acm.org/10.1145/1651461.1651471. Cited on pp. 16 and 17.

Charaudeau, Patrick. 1992. *Grammaire du sens et de l'expression*. Hachette Éducation. Cited on p. 12.

Chaumartin, François-Régis. 2007. *UPAR7: a knowledge-based system for headline sentiment tagging*. In Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval '07, pp. 422–425. Association for Computational Linguistics, Morristown, NJ, USA. URL http://portal.acm.org/citation.cfm?id=1621474.1621568. Cited on pp. 40, 41, and 67.

Daille, Béatrice; Dubreil, Estelle; Monceaux, Laura; and Vernier, Matthieu. Dec. 2011. *Annotating opinion–evaluation of blogs: the Blogoscopy corpus*. In Lang. Resour. Eval., vol. 45, no. 4, pp. 409–437. ISSN 1574-020X. doi:10.1007/s10579-011-9154-z. URL http://dx.doi.org/10.1007/s10579-011-9154-z. Cited on p. 2.

Dave, Kushal; Lawrence, Steve; and Pennock, David M. 2003. *Mining the peanut gallery: opinion extraction and semantic classification of product reviews*. In WWW '03: Proceedings of the 12th international conference on World Wide Web, pp. 519–528. ACM, New York, NY, USA. isbn 1-58113-680-3. doi:http://doi.acm.org/10.1145/775152.775226. Cited on p. 93.

Davidov, Dmitry; Tsur, Oren; and Rappoport, Ari. 2010. *Semi-supervised recognition of sarcastic sentences in Twitter and Amazon*. In Proceedings of the Fourteenth Conference on Computational

Natural Language Learning, CoNLL '10, pp. 107–116. Association for Computational Linguistics, Stroudsburg, PA, USA. isbn 978-1-932432-83-1. URL http://dl.acm.org/citation.cfm?id=1870568.1870582. Cited on p. 17.

Dobrov, Boris; Kuralenok, Igor; Loukachevitch, Natalia; Nekrestyanov, Igor; and Segalovich, Ilya. May 2004. *Russian Information Retrieval Evaluation Seminar*. In Proceedings of the Fourth International Conference on Language Resources and Evaluation. Lisbon, Portugal. Cited on pp. xii and 97.

Duh, Kevin; Fujino, Akinori; and Nagata, Masaaki. 2011. *Is machine translation ripe for cross-lingual sentiment classification?* In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2, HLT '11, pp. 429–433. Association for Computational Linguistics, Stroudsburg, PA, USA. isbn 978-1-932432-88-6. URL http://dl.acm.org/citation.cfm?id=2002736.2002823. Cited on pp. ix, 27, and 75.

Fan, Rong-En; Chang, Kai-Wei; Hsieh, Cho-Jui; Wang, Xiang-Rui; and Lin, Chih-Jen. June 2008. *LIBLINEAR: A Library for Large Linear Classification*. In J. Mach. Learn. Res., vol. 9, pp. 1871–1874. ISSN 1532-4435. URL http://portal.acm.org/citation.cfm?id=1390681.1442794. Cited on pp. viii, xiii, 61, 70, 80, 100, and 110.

Fišer, Darja and Sagot, Benoît. 2008. *Combining Multiple Resources to Build Reliable Wordnets*. In Proceedings of the 11th international conference on Text, Speech and Dialogue, TSD '08, pp. 61–68. Springer-Verlag, Berlin, Heidelberg. isbn 978-3-540-87390-7. doi:http://dx.doi.org/10.1007/978-3-540-87391-4_10. URL http://dx.doi.org/10.1007/978-3-540-87391-4_10. Cited on p. 38.

Généreux, Michel and Evans, Roger. 2006. *Towards a validated model for affective classification of texts*. In Proceedings of the Workshop on Sentiment and Subjectivity in Text, SST '06, pp. 55–62. Association for Computational Linguistics, Stroudsburg, PA, USA. isbn 1-932432-75-2. URL http://dl.acm.org/citation.cfm?id=1654641.1654649. Cited on p. 10.

González-Ibáñez, Roberto; Muresan, Smaranda; and Wacholder, Nina. 2011. *Identifying sarcasm in Twitter: a closer look*. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2, HLT '11, pp. 581–586. Association for Computational Linguistics, Stroudsburg, PA, USA. isbn 978-1-932432-88-6. URL http://dl.acm.org/citation.cfm?id=2002736.2002850. Cited on pp. 17 and 40.

Gross, Maurice. 1968. *Grammaire Transformationnelle du Français*. Larousse. Cited on p. 28.

Guo, Yuqing; van Genabith, Josef; and Wang, Haifeng. 2008. *Dependency-based n-gram models for general purpose sentence realisation*. In Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1, COLING '08, pp. 297–304. Association for Computational Linguistics, Stroudsburg, PA, USA. isbn 978-1-905593-44-6. URL http://dl.acm.org/citation.cfm?id=1599081.1599119. Cited on p. 67.

Habash, Nizar. 2004. *The use of a structural n-gram language model in generation-heavy hybrid machine translation*. In In Proceedings of the Third International Conference on Natural Language Generation (INLG04). Birghton, pp. 61–69. Springer. Cited on p. 67.

Harb, A.; Dray, G.; Plantié, M.; Poncelet, P.; Roche, M.; and Trousset, F. 2008. *Détection d'Opinion: Apprenons les bons Adjectifs!* vol. 8, pp. 59–66. URL http://hal.inria.fr/docs/00/27/77/85/PDF/papier6_Harb_FODOP08.pdf. Cited on p. 31.

Hovy, Eduard. November 2011. *Invited Keynote: What are Subjectivity, Sentiment, and Affect?* In Sentiment Analysis where AI meets Psychology, p. 1. Asian Federation of Natural Language Processing, Chiang Mai, Thailand. URL http://www.aclweb.org/anthology/W11-3701. Cited on p. 21.

Jackiewicz, A. 2010. *Structures avec constituants détachés et jugements d'évaluation.* In Document Numérique, pp. 11–40. Cited on p. 41.

Jansen, Bernard J.; Zhang, Mimi; Sobel, Kate; and Chowdury, Abdur. 2009. *Micro-blogging as online word of mouth branding*. In CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems, pp. 3859–3864. ACM, New York, NY, USA. isbn 978-1-60558-247-4. doi: http://doi.acm.org/10.1145/1520340.1520584. Cited on p. 51.

Jindal, Nitin and Liu, Bing. 2008. *Opinion spam and analysis*. In Proceedings of the international conference on Web search and web data mining, WSDM '08, pp. 219–230. ACM, New York, NY, USA. isbn 978-1-59593-927-2. doi:10.1145/1341531.1341560. URL http://doi.acm.org/10.1145/1341531.1341560. Cited on p. 18.

Jung, Yuchul; Park, Hogun; and Myaeng, Sung Hyon. 2006. *A hybrid mood classification approach for blog text*. In Proceedings of the 9th Pacific Rim international conference on Artificial intelligence, PRICAI'06, pp. 1099–1103. Springer-Verlag, Berlin, Heidelberg. isbn 978-3-540-36667-6. URL http://dl.acm.org/citation.cfm?id=1757898.1758055. Cited on p. 10.

Kendall, M.G. 1938. *A New Measure of Rank Correlation*. In Biometrika, vol. 30, no. 12, pp. 81–93. Cited on p. 58.

Kim, Soo-Min and Hovy, Eduard. 2004. *Determining the sentiment of opinions*. In Proceedings of the 20th international conference on Computational Linguistics, COLING '04. Association for Computational Linguistics, Stroudsburg, PA, USA. doi:http://dx.doi.org/10.3115/1220355.1220555. Cited on pp. 7 and 8.

Klenner, Manfred; Petrakis, Stefanos; and Fahrni, Angela. September 2009. *Robust Compositional Polarity Classification*. In Proceedings of the International Conference RANLP-2009, pp. 180–184. Association for Computational Linguistics, Borovets, Bulgaria. URL http://www.aclweb.org/anthology/R09-1034. Cited on p. 24.

Kobayashi, Nozomi; Inui, Kentaro; and Matsumoto, Yuji. 2007. *Extracting Aspect-Evaluation and Aspect-Of Relations in Opinion Mining*. In Proceedings of EMNLP-CoNLL 2007. URL http://acl.ldc.upenn.edu/D/D07/D07-1114.pdf. Cited on pp. 7 and 8.

Liu, Bing. 2010. *Sentiment analysis and subjectivity*. In Handbook of Natural Language Processing, Second Edition. Taylor and Francis Group, Boca. Cited on p. 13.

Lu, Yue; Castellanos, Malu; Dayal, Umeshwar; and Zhai, ChengXiang. 2011. *Automatic construction of a context-aware sentiment lexicon: an optimization approach*. In Proceedings of the 20th international conference on World wide web, WWW '11, pp. 347–356. ACM, New York, NY, USA. isbn 978-1-4503-0632-4. doi:http://doi.acm.org/10.1145/1963405.1963456. URL http://doi.acm.org/10.1145/1963405.1963456. Cited on p. 41.

Maas, Andrew L.; Daly, Raymond E.; Pham, Peter T.; Huang, Dan; Ng, Andrew Y.; and Potts, Christopher. June 2011. *Learning Word Vectors for Sentiment Analysis*. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 142–150. Association for Computational Linguistics, Portland, Oregon, USA. URL http://www.aclweb.org/anthology/P11-1015. Cited on pp. ix, 75, and 76.

Maritz, J.S. 1981. *Distribution-Free Statistical Methods*. In , p. 217. Cited on p. 58.

de Marneffe, MarieCatherine; Maccartney, Bill; and Manning, Christopher D. 2006. *Generating Typed Dependency Parses from Phrase Structure Parses*. In LREC. Cited on pp. 70 and 111.

Martin, J. R. and White, P. R. R. 2005. *The Language of Evaluation: Appraisal in English*. Palgrave Macmillan. Cited on p. 12.

Martineau, Justin and Finin, Tim. May 2009. *Delta TFIDF: An Improved Feature Space for Sentiment Analysis*. In Proceedings of the Third AAAI Internatonal Conference on Weblogs and Social Media. AAAI Press, San Jose, CA. (poster paper). Cited on pp. 44, 61, and 78.

Mathieu, Yvette Yannick. 2000. *Sciences du Langage*. In Les verbes de sentiment – De l'analyse linguistique au traitement automatique. CNRS Editions. Cited on p. 28.

———. 2006. *A Computational Semantic Lexicon of French Verbs of Emotion*. In Shanahan, James G.; Qu, Yan; and Wiebe, Janyce, eds., Computing Attitude and Affect in Text: Theory and Applications, vol. 20 of *The Information Retrieval Series*, chap. 10, pp. 109–124. Springer-Verlag, Berlin/Heidelberg. Cited on p. 41.

Matsumoto, Shotaro; Takamura, Hiroya; and Okumura, Manabu. 2005. *Sentiment Classification Using Word Sub-sequences and Dependency Sub-trees*. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 301–311. doi:10.1007/11430919_37. Cited on p. 42.

Meena, Arun and 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, pp. 573–580. Springer-Verlag, Berlin, Heidelberg. isbn 978-3-540-71494-1. URL http://portal.acm.org/citation.cfm?id=1763653.1763722. Cited on p. 67.

Messina, Diana; Morais, José; and Cantraine, Francis. April 1989. *Valeur affective de 904 mots de la langue française. / Emotional value of 904 words in the French language*. In Cahiers de Psychologie Cognitive/Current Psychology of Cognition, vol. 9, pp. 165–187. Cited on pp. 37 and 60.

Nakagawa, Tetsuji; Inui, Kentaro; and Kurohashi, Sadao. 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, pp. 786–794. Association for Computational Linguistics, Morristown, NJ, USA. isbn 1-932432-65-5. URL http://portal.acm.org/citation.cfm?id=1857999.1858119. Cited on pp. 44 and 66.

Navigli, Roberto and Ponzetto, Simone Paolo. 2010. *BabelNet: building a very large multilingual semantic network*. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10, pp. 216–225. Association for Computational Linguistics, Stroudsburg, PA, USA. URL http://dl.acm.org/citation.cfm?id=1858681.1858704. Cited on p. 38.

Neviarouskaya, Alena; Prendinger, Helmut; and Ishizuka, Mitsuru. 2010. *Recognition of affect, judgment, and appreciation in text*. In Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10, pp. 806–814. Association for Computational Linguistics, Stroudsburg, PA, USA. URL http://dl.acm.org/citation.cfm?id=1873781.1873872. Cited on p. 10.

Nishikawa, Hitoshi; Hasegawa, Takaaki; Matsuo, Yoshihiro; and Kikui, Genichiro. August 2010. *Opinion Summarization with Integer Linear Programming Formulation for Sentence Extraction and Ordering*. In Coling 2010: Posters, pp. 910–918. Coling 2010 Organizing Committee, Beijing, China. URL http://www.aclweb.org/anthology/C10-2105. Cited on p. 15.

Nivre, Joakim. 2005. *Dependency Grammar and Dependency Parsing*. Tech. rep., Växjö University: School of Mathematics and Systems Engineering. URL http://www.vxu.se/msi/~{}nivre/papers/05133.pdf. Cited on p. 66.

Nivre, Joakim; Hall, Johan; and Nilsson, Jens. 2006. *MaltParser: A data-driven parser-generator for dependency parsing*. In Proc. of LREC-2006. Cited on pp. 70 and 101.

Ogren, Philip. 2006. *Knowtator: A Protégé Plug-in for Annotated Corpus Construction*. In for Computational Linguistics, Association, ed., Proceedings of the Conference of the North American Chapter of the ACL on Human Language Technology: companion volume: demonstrations, pp. 273–275. New-York. Cited on p. 29.

Ott, Myle; Choi, Yejin; Cardie, Claire; and Hancock, Jeffrey T. 2011. *Finding deceptive opinion spam by any stretch of the imagination*. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, pp. 309–319. Association for Computational Linguistics, Stroudsburg, PA, USA. isbn 978-1-932432-87-9. URL http://dl.acm.org/citation.cfm?id=2002472.2002512. Cited on p. 18.

Paltoglou, Georgios and Thelwall, Mike. 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, pp. 1386–1395. Association for Computational Linguistics, Morristown, NJ, USA. URL http://portal.acm.org/citation.cfm?id=1858681.1858822. Cited on pp. 45 and 78.

Pang, Bo and Lee, Lillian. 2004. *A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts*. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, ACL '04. Association for Computational Linguistics, Stroudsburg, PA, USA. doi:http://dx.doi.org/10.3115/1218955.1218990. URL http://dx.doi.org/10.3115/1218955.1218990. Cited on pp. 13 and 23.

———. January 2008. *Opinion Mining and Sentiment Analysis*. In Found. Trends Inf. Retr., vol. 2, pp. 1–135. ISSN 1554-0669. doi:10.1561/1500000011. Cited on pp. iii, 1, and 7.

Pang, Bo; Lee, Lillian; and Vaithyanathan, Shivakumar. 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, pp. 79–86. Association for Computational Linguistics, Morristown, NJ, USA. doi:http://dx.doi.org/10.3115/1118693.1118704. URL http://dx.doi.org/10.3115/1118693.1118704. Cited on pp. 13, 22, 24, 42, 44, 78, and 93.

Parrott, Gerrod. 2001. *Emotions in Social Psychology*. Philadelphia: Psychology Press. Cited on p. 26.

Paumier, Sébastien. 2011. *Unitex 2.1 user manual*. URL http://igm.univ-mlv.fr/~unitex. Cited on p. 114.

Pazelskaya, A G and Solovyev, A N. May 2011. *A method of sentiment analysis in Russian texts*. In Proceedings of the Dialog 2011 the 17th International Conference On Computational Linguistics. Moscow region, Russia. Cited on pp. xii and 99.

Pennebaker, J. W.; Chung, C. K.; Ireland, M.; Gonzales, A.; and Booth, R. J. 2007. *The Development and Psychometric Properties of LIWC2007*. This article is published by LIWC Inc, Austin, Texas 78703 USA in conjunction with the LIWC2007 software program. URL http://www.liwc.net/LIWC2007LanguageManual.pdf. Cited on p. 18.

Pestian, John; Nasrallah, Henry; Matykiewicz, Pawel; Bennett, Aurora; and Leenaars, Antoon. Aug. 2010. *Suicide Note Classification Using Natural Language Processing: A Content Analysis.* In Biomedical informatics insights, vol. 2010, no. 3, pp. 19–28. ISSN 1178-2226. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3107011/. Cited on p. 109.

Plutchik, Robert. 2001. *The Nature of Emotions*. In American Scientist, vol. 89, no. 4, pp. 344+. ISSN 0003-0996. doi:10.1511/2001.4.344. Cited on p. 9.

Popel, Martin and Mareček, David. 2010. *Perplexity of n-gram and dependency language models*. In Proceedings of the 13th international conference on Text, speech and dialogue, TSD'10, pp. 173–180. Springer-Verlag, Berlin, Heidelberg. isbn 3-642-15759-9, 978-3-642-15759-2. URL http://dl.acm.org/citation.cfm?id=1887176.1887201. Cited on p. 67.

Prettenhofer, Peter and Stein, Benno. 2010. *Cross-language text classification using structural correspondence learning*. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10, pp. 1118–1127. Association for Computational Linguistics, Stroudsburg, PA, USA. URL http://dl.acm.org/citation.cfm?id=1858681.1858795. Cited on pp. viii, 27, and 70.

Read, Jonathon. 2004. *Recognising Affect in Text using Pointwise-Mutual Information*. Master's thesis, University of Sussex. URL http://www.cogs.susx.ac.uk/users/jlr24/papers/read-us04.pdf. Cited on p. 10.

———. 2005. *Using Emoticons to Reduce Dependency in Machine Learning Techniques for Sentiment Classification*. In ACL. Cited on p. 31.

Redondo, Jaime; Fraga, Isabel; Padrón, Isabel; and Comesaña, Montserrat. August 2007. *The Spanish adaptation of ANEW (affective norms for English words).* In Behavior research methods, vol. 39, no. 3, pp. 600–5. ISSN 1554-351X. URL http://www.ncbi.nlm.nih.gov/pubmed/17958173. Cited on pp. 38 and 41.

Rentoumi, Vassiliki; Giannakopoulos, George; Karkaletsis, Vangelis; and Vouros, George A. September 2009. *Sentiment Analysis of Figurative Language using a Word Sense Disambiguation Approach*. In Proceedings of the International Conference RANLP-2009, pp. 370–375. Association for Computational Linguistics, Borovets, Bulgaria. URL http://www.aclweb.org/anthology/R09-1067. Cited on p. 23.

Rentoumi, Vassiliki; Petrakis, Stefanos; Klenner, Manfred; Vouros, George A.; and Karkaletsis, Vangelis. may 2010. *United we Stand: Improving Sentiment Analysis by Joining Machine Learning and Rule Based Methods*. In Chair), Nicoletta Calzolari (Conference; Choukri, Khalid; Maegaard, Bente; Mariani, Joseph; Odijk, Jan; Piperidis, Stelios; Rosner, Mike; and Tapias, Daniel, eds., Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). European Language Resources Association (ELRA), Valletta, Malta. isbn 2-9517408-6-7. Cited on p. 23.

Reyes, Antonio and Rosso, Paolo. 2011. *Mining subjective knowledge from customer reviews: a specific case of irony detection*. In Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, WASSA '11, pp. 118–124. Association for Computational Linguistics, Stroudsburg, PA, USA. isbn 9781937284060. URL http://dl.acm.org/citation.cfm?id=2107653.2107668. Cited on pp. 17 and 40.

Rish, Irina. 2001. *An empirical study of the naive Bayes classifier*. In IJCAI-01 workshop on "Empirical Methods in AI". URL http://www.intellektik.informatik.tu-darmstadt.de/~{}tom/IJCAI01/Rish.pdf. Cited on p. 42.

Schmid, Helmut. 1994. *Probabilistic Part-of-Speech Tagging Using Decision Trees*. In Proceedings of the International Conference on New Methods in Language Processing, pp. 44–49. Cited on pp. 55 and 111.

Sharoff, Serge; Kopotev, Mikhail; Erjavec, Tomaz; Feldman, Anna; and Divjak, Dagmar. may 2008. *Designing and Evaluating a Russian Tagset*. In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08). European Language Resources Association (ELRA), Marrakech, Morocco. isbn 2-9517408-4-0. Http://www.lrec-conf.org/proceedings/lrec2008/. Cited on p. 101.

Soon, Wee Meng; Ng, Hwee Tou; and Lim, Daniel Chung Yong. Dec. 2001. *A machine learning approach to coreference resolution of noun phrases*. In Comput. Linguist., vol. 27, pp. 521–544. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=972597.972602. Cited on p. 14.

Stone, Philip J. and Hunt, Earl B. 1963. *A computer approach to content analysis: studies using the General Inquirer system*. In Proceedings of the May 21-23, 1963, spring joint computer conference, AFIPS'63 (Spring), pp. 241–256. ACM, New York, NY, USA. doi:http://doi.acm.org/10.1145/1461551.1461583. URL http://doi.acm.org/10.1145/1461551.1461583. Cited on pp. 38 and 109.

Stoyanov, Veselin and Cardie, Claire. 2008. *Topic identification for fine-grained opinion analysis*. In Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1, COLING '08, pp. 817–824. Association for Computational Linguistics, Stroudsburg, PA, USA. isbn 978-1-905593-44-6. URL http://dl.acm.org/citation.cfm?id=1599081.1599184. Cited on p. 14.

Strapparava, Carlo and Valitutti, Alessandro. 2004. *WordNet-Affect: an affective extension of WordNet*. In In Proceedings of the 4th International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA). Cited on p. 40.

Syssau, A. and Font, N. 2005. *Evaluations des caractéristiques émotionnelles d'un corpus de 604 mots.* In Bulletin de Psychologie, vol. 58, pp. 361–367. Cited on pp. 37 and 60.

Takamura, Hiroya; Inui, Takashi; and Okumura, Manabu. 2005. *Extracting semantic orientations of words using spin model*. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05, pp. 133–140. Association for Computational Linguistics, Stroudsburg, PA, USA. doi:http://dx.doi.org/10.3115/1219840.1219857. URL http://dx.doi.org/10.3115/1219840.1219857. Cited on p. 41.

Thompson, Geoff and Hunston, Susan. 2000. *Evaluation: An Introduction*. In Hunston, Susan and Thompson, Geoff, eds., Evaluation in Text: authorial stance and the construction of discourse, pp. 1–27. Oxford University Press, Oxford, England. Cited on p. 8.

Toprak, Cigdem; Jakob, Niklas; and Gurevych, Iryna. 2010. *Sentence and expression level annotation of opinions in user-generated discourse*. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10, pp. 575–584. Association for Computational Linguistics, Stroudsburg, PA, USA. URL http://dl.acm.org/citation.cfm?id=1858681.1858740. Cited on p. 28.

Tsur, Oren; Davidov, Dmitry; and Rappoport, Ari. 2010. *ICWSM - A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews*. In Cohen, William W. and Gosling,

Samuel, eds., Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010. The AAAI Press. doi:http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/view/1495. Cited on p. 17.

Turney, Peter and Littman, Michael. 2003. *Measuring Praise and Criticism: Inference of Semantic Orientation from Association*. In ACM Transactions on Information Systems, vol. 21, pp. 315–346. Cited on p. 41.

Vernier, M. and Monceau, L. 2010. *Enrichissement d'un lexique de termes subjectifs à partir de tests sémantiques*. In Traitement Automatique des Langues, pp. 125–149. Cited on p. 41.

Vetulani, Zygmunt; Walkowska, Justyna; Obrębski, Tomasz; Marciniak, Jacek; Konieczka, Paweł; and Rzepecki, Przemys. 2009. *An Algorithm for Building Lexical Semantic Network and Its Application to PolNet - Polish WordNet Project*, pp. 369–381. Springer-Verlag, Berlin, Heidelberg. isbn 978-3-642-04234-8. doi:10.1007/978-3-642-04235-5_32. URL http://dl.acm.org/citation.cfm?id=1616945.1616983. Cited on p. 38.

Vō, M. L.-H.; Conrad, M.; Kuchinke, L.; K.Urton; Hofmann, M. J.; and Jacobs, Arthur M. May 2009. *The Berlin Affective Word List Reloaded (BAWL-R)*. In Behavior research methods, vol. 41, no. 2, pp. 534–538. ISSN 1554-351X. Cited on pp. 38 and 41.

Vossen, Piek, ed. 1998. *EuroWordNet: a multilingual database with lexical semantic networks*. Kluwer Academic Publishers, Norwell, MA, USA. isbn 0-7923-5295-5. Cited on p. 38.

Whitelaw, Casey; Garg, Navendu; and Argamon, Shlomo. 2005. *Using appraisal groups for sentiment analysis*. In Proceedings of the 14th ACM international conference on Information and knowledge management, CIKM '05, pp. 625–631. ACM, New York, NY, USA. isbn 1-59593-140-6. doi:http://doi.acm.org/10.1145/1099554.1099714. URL http://doi.acm.org/10.1145/1099554.1099714. Cited on p. 42.

Wiebe, Janyce and Cardie, Claire. 2005. *Annotating expressions of opinions and emotions in language. Language Resources and Evaluation*. In Language Resources and Evaluation (formerly Computers and the Humanities, p. 2005. Cited on p. 28.

Wiegand, Michael; Roth, Benjamin; and Klakow, Dietrich. 2010. *A Survey on the Role of Negation in Sentiment Analysis*. Cited on p. 24.

Wu, Jun and Khudanpur, Sanjeev. 1999. *Combining Nonlocal, Syntactic And N-Gram Dependencies In Language Modeling*. In Proceedings of Eurospeech'99, vol, pp. 2179–2182. Cited on p. 67.

Wu, Yunfang; Jin, Peng; Wen, Miaomiao; and Yu, Shiwen. 2010. *SemEval-2010 Task 18: Disambiguating Sentiment Ambiguous Adjectives*. In SemEval 2010: Proceedings of International Workshop of Semantic Evaluations, pp. 275–278. ACL SigLex, Uppsala, Sweden. Cited on pp. xi and 91.

Yang, Dan and Lee, Won-Sook. 2009. *Music Emotion Identification from Lyrics*. In Proceedings of the 2009 11th IEEE International Symposium on Multimedia, ISM'09, pp. 624–629. IEEE Computer Society, Washington, DC, USA. isbn 978-0-7695-3890-7. doi: http://dx.doi.org/10.1109/ISM.2009.123. URL http://dx.doi.org/10.1109/ISM.2009.123. Cited on p. 112.

Yarowsky, David. 1995. *Unsupervised word sense disambiguation rivaling supervised methods*. In Proceedings of the 33rd annual meeting on Association for Computational Linguistics, ACL '95, pp. 189–196. Association for Computational Linguistics, Stroudsburg, PA, USA. doi:http://dx.doi.org/10.3115/981658.981684. URL http://dx.doi.org/10.3115/981658.981684. Cited on p. 22.

Zhuang, Li; Jing, Feng; and Zhu, Xiao-Yan. 2006. *Movie review mining and summarization*. In Proceedings of the 15th ACM international conference on Information and knowledge management, CIKM '06, pp. 43–50. ACM, New York, NY, USA. isbn 1-59593-433-2. doi: http://doi.acm.org/10.1145/1183614.1183625. URL http://doi.acm.org/10.1145/1183614.1183625. Cited on p. 66.

Zirn, Cacilia; Niepert, Mathias; Stuckenschmidt, Heiner; and Strube, Michael. November 2011. *Fine-Grained Sentiment Analysis with Structural Features*. In Proceedings of 5th International Joint Conference on Natural Language Processing, pp. 336–344. Asian Federation of Natural Language Processing, Chiang Mai, Thailand. URL http://www.aclweb.org/anthology/I11-1038. Cited on pp. 22 and 44.

INDEX