Learning Explainable User Sentiment and Preferences for Information Filtering

THÈSE N^{O} 6920 (2016)

PRÉSENTÉE LE 4 MARS 2016

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE L'IDIAP
PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Nikolaos PAPPAS

acceptée sur proposition du jury:

Prof. S. Süsstrunk, présidente du jury
Prof. H. Bourlard, Dr A. Popescu-Belis, directeurs de thèse
Prof. T. Hofmann, rapporteur
Prof. S.-F. Chang, rapporteur
Dr J.-C. Chappelier, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2016

To the loving memory of my grandfather Christos…

Acknowledgements

“Once we accept our limits, we go beyond them.” —Albert Einstein

Preparing the PhD thesis was one of the most fascinating, challenging and rewarding periods of my life. As I look back, this period was the most favorable time to reach my limits and strive to go beyond them. Having arrived at the end of this long journey, I am very grateful to have met bright, motivated and knowledgeable people with persistence and critical thinking along the way, from whom I learned a lot and with whom I shared numerous wonderful moments, extraordinary activities, stimulating discussions and fruitful collaborations which greatly enriched my experiences and positively influenced me in many different ways.

First and foremost, I thank my advisors Andrei Popescu-Belis and Hervé Bourlard for giving me the opportunity to work on interesting topics in a perfect environment. I especially thank Andrei for teaching me how to be more concise and organized, sharing my enthousiasm, as well as for providing me his constant support, guidance and availability throughout the past four years. I also would like to acknowledge the institutions which funded my research, namely the 7th Framework Program of the European Union (FP7), the Swiss National Science Foundation (SNF), The Ark Foundation and the Idiap Research Institute.

I am greatful to my thesis committee – Sabine Süsstrunk, Shih-Fu Chang, Jean-Cédric Chappelier and Thomas Hofmann who reviewed my thesis and provided me with interesting questions, constructive feedback and helpful remarks. Furthermore, I am greatly thankful to Shih-Fu Chang for hosting my internship at DVMM in Columbia University and for including me in stimulating and rigorous research discussions and exciting research projects; it was a very interesting and productive experience.

Thanks to all my colleagues at Idiap who made this journey enjoyable and memorable with cool discussions and awesome outside activities: skiing, climbing, swimming, slacklining, jogging, lake excursions, lunches, coffee breaks, babyfoot, card games, themed dinners, hanging out, concerts and many more; thank you Thomas, Manuel, Elie, Laurent E., Ivana, Tiago, André, Adolfo, Roy, Ilja, Arjan, Leo, Riwal, Gwenole, Samira, Majid, Sharid, Maryam, Serena, Nesli, Marco, Cijo, Tatjana, James, Alexandre, Petr, Aleksandra, Maryam, Rui, Darshan, Kenneth, Pranay, Gulcan, Pierre, Alexandros, Phil, Marc, Blaise, Raphael, Dayra, Darshan, Laurent N., Joan, Xiao, Quang, Parvaz, Yannis, Daniel, Ajay, Pedro, Dimitri, Remi, Joel and others. It has been a great pleasure to meet you all. Special thanks go to Majid, Ilja, Marc, Dimitri, Cijo, Leo, Marco, Thomas, Darshan and others for the interesting and helpful research discussions. Moreover, I would especially like to thank Thomas, Majid, and Ilja for support, friendship and amazing time we spent together.

Finally, I would like to heartily thank Amanda for her love, encouragement and patience. Thank you for making me smile, for being straightforward and always by my side. Furthermore, I would like to deeply thank my numerous long-lasting friends from Greece who spread all over Europe and those who stayed back. Lastly, I am very grateful to my family and relatives: Dimitra and Petros, my loving parents, who supported my education by all means and respected my choices; Evagelia and Christos, my beloved siblings and super awesome friends, who stood always by me in good and difficult times.

Lausanne, 14 January 2016 Nikolaos Pappas

Abstract

In the last decade, online social networks have enabled people to interact in many ways with each other and with content. The digital traces of such actions reveal people's preferences towards online content such as news or products. These traces often result from interactions such as sharing or liking, but also from interactions in natural language. The continuous growth of the amount of content and of digital traces has led to information overload: surrounded by large volumes of information, people are facing difficulties when searching for information relevant to their interests. To improve user experience, information systems must be able to assist users in achieving their search goals, effectively and efficiently.

This thesis is concerned with two important challenges that information systems need to address in order to significantly improve search experience and overcome information overload. First, these systems need to model accurately the variety of user traces, and second, they need to meaningfully explain search results and recommendations to users. To address these challenges, this thesis proposes novel methods based on machine learning to model user sentiment and preferences for information filtering systems, which are effective, scalable, and easily interpretable by humans.

We focus on two prominent types of user traces in social networks: on the one hand, user comments accompanied by unary preferences such as likes, and on the other hand, user reviews accompanied by numerical preferences such as star ratings. In both cases, we advocate that by better understanding user text through mining its semantics and modeling its structure, we can not only improve information filtering, but also explain predictions to users. Within this context, we aim to answer three main research questions, namely: (i) how do item semantics help to predict unary preferences; (ii) how do sentiments of free-form user texts help to predict unary preferences; and (iii) how to model fine-grained numerical preferences from user review texts. Our goal is to model and extract from user text the knowledge required to answer these questions, and to obtain insights on how to design better information filtering systems that are more effective and improve user experience.

To answer the first question, we formulate the recommendation problem based on unary preferences as a top-N retrieval task and we define an appropriate dataset and metrics for measuring performance. Then, we propose and evaluate several content-based methods based on semantic similarities under presence or absence of preferences. To answer the second question, we propose a sentiment-aware neighborhood model which integrates the sentiment of user comments with unary preferences, either through fixed or through learned mapping functions. For the latter type, we propose a learning algorithm which adapts the sentiment of user comments to unary preferences at collective or individual levels. To answer the third question, we cast the problem of modeling user attitude toward aspects of items as a weakly supervised problem, and we propose a weighted multiple-instance learning method for solving it. Lastly, we show that the learned saliency weights, apart from being easily interpretable, are useful indicators for review segmentation and summarization.

To maximize the practical impact of our study, we experiment on real-world data collected from online social networks or Web communities such as TED, Flickr, Vimeo, Audible, RateBeer and BeerAdvocate. In addition to providing answers to the questions above, we also show how to build novel applications derived from them, such as emotion-based recommendation, lecture transcript summarization, explaining sentiment or emotion scores, and predicting emotion attributes of lectures based on transcripts or comments.

Keywords: Sentiment Analysis, One-class Collaborative Filtering, Semantic Similarity, Recommender Systems, Content-based Recommendation, Multimedia Recommendation, Multiple-Instance Learning, Aspect Rating Prediction, Structured Learning, Review Summarization, Review Segmentation, Sentence Ranking

Résumé

Les réseaux sociaux en ligne ont permis à leurs utilisateurs, particulièrement durant la dernière décennie, d'interagir de manière variée avec d'autres utilisateurs ou avec des contenus numériques. Les traces de ces interactions révèlent leurs préférences envers des informations ou des produits disponibles en ligne. Ces traces proviennent souvent du partage en ligne ou des listes de favoris, mais également des textes plus ou moins détaillés écrits par les utilisateurs à leur propos. La croissance continue de la quantité de contenus et de traces laissées par les utilisateurs conduit à une surcharge informationnelle: entourés de nombreuses informations, les individus trouvent difficilement celles qui sont réellement pertinentes pour eux. Afin d'améliorer leur expérience, les systèmes d'information devraient être capables de guider plus efficacement les utilisateurs pour leur accès à l'information.

Cette thèse s'intéresse à deux importants défis qui se posent aux systèmes d'information pour pouvoir améliorer le processus de recherche et gérer la surcharge informationnelle. D'abord, ces systèmes doivent analyser plus précisément la diversité des traces laissées par les utilisateurs, et ensuite, ils doivent être capables d'expliquer aux utilisateurs les résultats de leurs recherches ou les recommandations proposées. Pour aider à relever ces défis, cette thèse propose des méthodes novatrices fondées sur l'apprentissage automatique qui permettent de modéliser les sentiments et les préférences des utilisateurs, au sein de systèmes de filtrage d'information, de manière efficace, extensible et facile à interpréter pour des humains.

Nous nous intéressons à deux types importants de traces que les utilisateurs des réseaux sociaux laissent derrière eux: d'un côté, les commentaires, accompagnés d'indications unaires de préférences (*like*), et de l'autre côté les critiques, accompagnées d'appréciations numériques (par exemple de une à cinq étoiles). Dans les deux cas, nous montrons qu'une meilleure compréhension des textes, par l'analyse sémantique et structurelle, permet d'améliorer le filtrage d'information et expliquer les prédictions du système à ses utilisateurs. Dans ce cadre, nous répondons à trois questions de recherche : (i) comment le contenu sémantique des entités aide-t-il à prédire les préférences unaires ? (ii) comment les sentiments de les commentaires aident-ils à améliorer la prédiction de ces préférences unaires ? (iii) comment modéliser des préférences numériques pour des aspects spécifiques à partir des critiques ? Notre but est de modéliser et d'extraire à partir des textes rédigés par les utilisateurs les informations nécessaires pour répondre à ces questions, et d'en déduire des méthodes pour concevoir des systèmes d'information plus efficaces et qui améliorent l'expérience de l'utilisateur.

Pour répondre à la première question, nous formulons le problème de la recommandation basée sur des préférences unaires comme une tâche de recherche des N documents les plus pertinents, et nous mettons en place un jeu de données et une métrique pour l'évaluation. Puis, nous proposons et évaluons plusieurs méthodes de recommandation utilisant la similarité sémantique entre les contenus, en présence des préférences ou bien en son absence. Pour répondre à la deuxième question, nous proposons un modèle de voisinage prenant en compte les sentiments, qui combine les préférences unaires et le sentiment des commentaires laissés par les utilisateurs, soit de manière rigide, soit en adaptant leur combinaison aux données. Dans ce dernier cas, nous proposons un algorithme d'apprentissage qui adapte l'amplitude des sentiments extraits automatiquement des commentaires au niveau des préférences unaires, soit individuellement, soit pour tous les utilisateurs. Pour répondre à la troisième question, nous formulons la modélisation de l'attitude des utilisateurs envers les aspects des items comme un problème d'apprentissage faiblement supervisé, et proposons pour le résoudre une approche fondée sur l'apprentissage à base d'instances multiples pondérées. Nous montrons alors que les poids appris par notre modèle sont non seulement faciles à interpréter, mais constituent des traits pertinents pour la segmentation et le résumé des critiques.

Afin de maximiser l'impact pratique de nos propositions, nous utilisons dans nos expériences des données provenant de réseaux sociaux ou de communautés en ligne, obtenus à partir de sites Internet tels que TED, Flickr, Vimeo, Audible, RateBeer et BeerAdvocate. Les implémentations permettent d'apporter des réponses aux questions énoncées ci-dessus, mais aussi de construire des applications pratiques nouvelles, notamment pour la recommandation fondée sur les émotions, le résumé des transcriptions de conférences, l'explication des niveaux de sentiment, et la prédiction des émotions suscitées par une conférence à partir de sa transcription ou des commentaires reçus.

Mots-clefs: Analyse des Sentiments, Filtrage Collaboratif à une Classe, Similarité Sémantique, Systèmes de Recommandation, Recommandation Basée Sur le Contenu, Recommandation Multimédia, Apprentissage à Instances Multiples, Prédiction des Aspects, Apprentissage Structuré, Résumé de Critiques, Segmentation de Critiques, Classement de Phrases

Contents

List of Figures

List of Tables

List of Abbreviations

NLP	Natural Language Processing
ML	Machine Learning
RS	Recommender Systems
SA	Sentiment Analysis
CF	Collaborative Filtering
ICF	Item-Based Collaborative Filtering
UCF	User-Based Collaborative Filtering
CB	Content-Based
VSM	Vector Space Model
POS	Part-Of-Speech
BOW	Bag-of-Words
TF-IDF	Term Frequency - Inverse Document Frequency
ESA	Explicit Semantic Analysis
RP	Random Projections
LSI	Latent Semantic Indexing
LDA	Latent Dirichlet Allocation
CRF	Conditional Random Field
MIL	Multiple Instance Learning
MIR	Multiple Instance Regression
SVM	Support Vector Machines
SVR	Support Vector Regression
NN	Nearest Neighbors
SANN	Sentiment-Aware Nearest Neighbors
OCCF	One-Class Collaborative Filtering
RB	Rule-Based
AMAU	All Missing as Unknown
AMAN	All Missing as Negative
EMAN	Equal-to-positive Missing as Negative
SVD	Singular Value Decomposition
NMF	Non-Negative Matrix Factorization
SNMF	Sparse Non-Negative Matrix Factorization

MF	Matrix Factorization
KL	Kullback-Leibler divergence
RLS	Regularized Least Squares
EM	Expectation Maximization
AP$_1$	Alternating Projections
AP$_2$	Average Precision
AR	Average Recall
AF	Average F-measure
MAP	Mean Average Precision
MAR	Mean Average Recall
MAF	Mean Average F-measure
MSE	Mean Squared Error
MAE	Mean Absolute Error
AUC	Area Under Curve
TI	Title
PC	Pearson Correlation Coefficient
COS	Cosine Similarity
DE	Description
RTA	Related Tags
RTH	Related Themes
TRA	Transcript
SP	Speaker
TE	TED Event
API	Application Program Interface
MPQA	Multi-Perspective Question Answering
pos	Positive
neg	Negative
neu	Neutral

Notations

Notations in the recommendation framework

U	The set of users indexed by u with size N_u or n, pages 6–7, 37–38, 61–68, 70–71.
I	The set of items indexed by i with size N_i or m, pages 6, 37–39, 61–68.
r_{ui}, $r(u,i)$	The preference of user u for item i, $r_{ui} \in \{1, ?\}$ where 1 denotes presence of action and ? absence (unary feedback) or $r_{ui} \in \{1, \ldots, b\}$ i.e a number from 1 to b (numerical feedback), pages 5, 14, 41–42, 61, 63, 65–67.
R_{known}	The set of user item ratings which are known, pages 65–68.
$\mathscr{M}_u$	The favorite items of a user u in the training set, pages 37–38, 43.
$\mathscr{T}_u$	The favorite items of a user u in the test set, pages 37–38, 43–47, 70–71.
$\mathscr{T}_i$	The ground-truth items related to i, pages 37–38.
$\mathscr{R}_u$	The set of top-N recommended items for each user u, page 34, 70–71.
$\mathscr{R}_i$	The recommended items for each item i, page 35.
$P(N)$	The precision at specified N, pages 38, 45–47, 59, 70–71, 75–79, 81–82.
$R(N)$	The average recall and recall at specified N, pages 38, 45–47, 59, 70–71, 75–79, 81–82.
$F(N)$	The F-measure at specified N, pages 38, 43–45, 48–49, 59, 71, 75–78, 80.
V	The vocabulary of words contained in a corpus, pages 39.
S	The similarity matrix of items based on a VSM such as TF-IDF or LSI, pages 39–40.
s_{ij}	The similarity of item i with item j, pages 40–42, 62.
$\hat{r}_{ui}$	The estimated relevance score of an item u for an item i, pages 40–42, 61, 63, 65–67.
b_{ui}	The bias estimate for user u and i, pages 41–42, 61–63.
$\bar{r}_u$	The average rating of a user u, pages 57-58.
$\bar{r}_i$	The average rating of an item i, pages 57-58.
d_{ij}	The similarity of item i with item j computed using s_{ij} and a shrinking factor λ which varies from 0 to 1, pages 41, 61–63, 69, 72–73.
n_{ij}	The number of common raters between item i and j, page 36, 62.
t_{ij}	The coefficient of word j for document i in a VSM, pages 6, 39.
tf_{ij}	The frequency of word j for document i in a VSM, pages 41.
idf_j	The inverse document frequency of word j, pages 41.
t	Number of topics (or latent factors) of topic modeling methods, page 39, 45.
$D^k(u;i)$	The k most similar items to item $i \in I$ of user u, pages 40–42, 61–63, 67
$\vec{i}$	The feature vector of item i derived from the co-rating matrix, pages 40–41, 62.

Notations in the sentiment-aware recommendation framework

r'_{uj} The piecewise rating function for item u and item j that accounts both for explicit ratings and for those inferred from comments, page 63, 65.

m_{uj} The mapping function of sentiment scores to ratings, pages 63–65.

C_{uj} The set of comments made by a user u to an item j, pages 63–65, 67.

$pol_{RB}(s)$ The hypothesized polarity of sentence s by RB classifier, page 58, 64.

$pol_{RB}(c)$ The polarity of comment c computed by the sum of polarities of each sentence $s \in c$ normalized by the length of s in terms of words, page 58, 64–65.

$pol_{RB}(C_{uj})$ The average polarity of C_{uj} comments computed with $pol_{RB}(c)$, pages 64–65.

$sign_{RB}(C_{uj})$ The sign of the average polarity of the comments $c \in C_{uj}$, $pol_{RB}(c)$, page 64, 67.

κ The inter-annotator agreement according to Fleiss' kappa, pages 58–59.

θ, θ_{pos} The global parameter for the positive class of a comment, pages 7–8, 64–67, 74.

α, θ_{neg} The global parameter for negative class of a comment, pages 7–8, 64–67 74.

υ, θ_{neu} The global parameter for neutral class of a comment, pages 7–8, 64–67, 66, 74.

θ_u The parameter for the positive class of a comment indexed by u, pages 7, 66–67.

α_u The parameter for negative class of a comment indexed by u, pages 7, 66–67.

υ_u The parameter for neutral class of a comment indexed by u, pages 7, 66–67.

η The offset parameter of the linear relationship, page 66.

ζ The slope parameter of the linear relationship, page 66.

ζ_u The user-specific slope parameter of the linear relationship, page 66.

ϵ The regularization term of the parameters of the SANN models, pages 66–67.

Notations in the multiple-instance learning framework

$B = \{b_{ij}\}^{d}_{n_i}$ The set of m bags of n_i d-dimensional instances which are indexed by bag i and instance j in a given dataset, pages 9–10, 87–92, 94, 96, 111–113.

$Y = \{y_i\}^{k}_{m}$ The set of m k-dimensional target numerical or categorical labels (e.g. aspect ratings) per bag i in a given dataset, pages 9, 90–92, 94, 96, 102, 111–113.

$\Psi = \{\psi_{ijk}\}^{1}_{n_i}$ The set of n_i-dimensional instance weights per bag i learned for a given aspect rating class k, page 9; the index k is dropped (Ψ) when referring to a single aspect rating class, pages 91–93.

$X = \{x_{ik}\}^{d}_{m}$ The set of m exemplar bag representations for each rating class k indexed by i and results from the convex combination of the n_i bag d-dimensional instances, pages 9–10; the index k is dropped (X) when referring to a single aspect rating class, pages 87–88, 90–91.

Φ The regression coefficients of size d learned for a given aspect rating class k; pages 9–10; the index k is dropped (Φ) when referring to a single aspect rating class, pages 87, 90–94, 102–103, 111.

b_{ij} The j-th d-dimensional instance of bag i, pages 9–10, 90–92, 94, 111–113.

y_i The numerical label of the i-th bag, pages 9, 90–92, 94, 96.

ψ_{ij} The contribution weight of the j-th instance of review i, pages 9–10

x_i The d-dimensional representation of the i-th bag, pages 9–10, 87–88, 90–91.

y_{ij} The categorical label of j-th instance of bag i, pages 111–113, 122.

$\hat{y}_{ij}$ The predicted categorical label of j-th instance of bag i, 122.

$\mathscr{L}$ The loss function based on least squares to be minimized, page 87, 91.

f Abstract notation for a regression models, page 88–90. 96.

D_{agg} The set of bag vectors and class labels that Aggregated-MIR is trained on, page 88.

D_{ins} The set of instance vectors and class labels that Instance-MIR is trained on, page 88.

D_{pri} The set of prime bag vectors and class labels that Prime-MIR is trained on, page 89.

b_i^p The prime instance for the bag i, page 89.

f_1 The abstract notation of the regularized regression model with constraints which learns Ψ instance weights per bag, pages 91–93, 96.

f_2 The abstract notation of the regularized regression model which learns regression coefficients Φ, 90–94, 96, 111.

f_3 The abstract notation of the regularized regression model which learns O regression coefficients to predict Ψ, pages 91–94, 96.

D_w The set of pairs of instances and learned relevance weights to be used for the training of f_3, page 92.

ϵ_1 The regularization term of f_1 regression model, pages 91, 94, 96.

ϵ_2 The regularization term of f_2 regression model, pages 91, 94, 96.

ϵ_3 The regularization term of f_3 regression model, pages 92, 94, 96.

$\hat{\psi}_i$ The estimated instance weights for bag i, pages 88–89, 96–97, 101–102, 111–113.

$\hat{y}, f_2(B_i)$ The estimated target label of an unseen bag i, pages 85–86, 96–97, 102, 111–112.

T_{ap} The time complexity of the AP procedure in Algorithm 2, page 87.

T_{f_s} The time complexity of the f_1, f_2 or f_3 model, page 87.

$\hat{n}$ The average size of bags, page 87.

$\hat{m}'$ The number of bags in the training set, page 92.

ℓ_1 The regularization of the model using ℓ_1 norm, pages 82, 92–95.

ℓ_2 The regularization of the model using ℓ norm, pages 82, 92–95.

$p(y|b)$ The conditional distribution of y given b based on CRF, pages 112, 113, 121.

ψ_s The function which models the aspect variable at a given position, pages 112.

$\psi_{s\curvearrowright}$ The function which models the transition between the aspect variables at positions j and $j+1$ in the linear chain, pages 112.

$f_s(y_{ij}, b_{ij})$ The feature vector of the j_{th} sentence of i_{th} review, pages 112.

ψ_s The model of the aspect label of the j_{th} sentence of i_{th} review, pages 112.

$A_{s,l}^{B_i}$ A bipartite graph constructed for bag B_i which maps n_i sentences to their most likely aspect indexed by s and l respectively, page 113.

f_a An assignment function which matches sentences to aspects, page 113.

1 Introduction

1.1 Context and Motivation

The information age has seen an increasing interest in methods for processing large amounts of data in reasonable time, taking into consideration its volume, variety and growth. The Web in its early stages enabled people to view content across the world regardless of their location. In the last decade, apart from simple actions such as viewing online content, people have become able to interact at multiple levels with each other and with content in social networks such as Facebook, YouTube and Twitter. The digital traces of such actions reveal the behavior of people, namely their preferences towards entities, products, news, events or other types of online content. These traces often result not only from interactions such as sharing, liking, rating or marking as a favorite, but also from interactions in natural language, for example, through comments or reviews.

The volume of data and of traces left by users increases dramatically over time. Some examples of social network statistics highlight the importance of the data volume, variety and growth. For instance, Facebook monthly active users increased from 100 million to 1.4 billion from 2008 to 2015[1]. These users connect with other users through friendship relations, create groups of interest, generate content such as comments, posts or shares, and express their appreciation for online content such as news or products. Similarly, Youtube has presently more than one billion users who upload 300 hours of videos every minute and generate billions of views every day[2]. Twitter has 300 million users who send 500 million messages per day[3], share messages that they like, and follow other users. These statistics exemplify the concept of information overload, which every user in these social networks faces when searching for information that is relevant to their interests. To help users, information systems should be able to assist them to retrieve relevant information and access large volumes of information on the Web, effectively and effortlessly.

[1] First Quarter 2015 Results, Facebook, 2015, http://investor.fb.com/releasedetail.cfm?ReleaseID=908022/.
[2] Product statistics, Youtube, 2015, https://www.youtube.com/yt/press/statistics.html.
[3] Twitter turns six, Twitter, 2012, https://blog.twitter.com/2012/twitter-turns-six/.

Recommender systems (RSs) are information filtering systems which are designed to cope with the aforementioned data deluge. These systems attempt to predict the preferences of users for items that they have not seen yet and to recommend to them items that might be of interest to them (Resnick and Varian, 1997; Koren and Bell, 2011; Ricci et al., 2010). The recommendations that are made are related to users' decision-making processes, such as what products to buy, what movies to watch, what music to listen or what news to read[4]. RSs typically use the properties of items and the explicit feedback collected from users to predict users' preferences for unseen items. In recent years, recommender systems have become popular within online social networks, but also in commercial websites such Amazon and E-bay because they help users find useful information among the vastness of available information, enhance user experience, and reinforce the users' engagement with the website.

The elementary user actions that are typically exploited by recommender systems, such as liking or rating, reveal the preferences of individual users. However, these systems do not have access to information about why they were performed. On the contrary, the user actions that result in written text, such as reviewing or commenting, provide more insights and often provide reasons why users express preferences the way they do. Automatic sentiment analysis can identify and extract subjective information from written texts, such as the overall polarity expressed in them or the attitude of a speaker or writer towards some topic (Pang and Lee, 2008). The attitude may be the evaluation or judgment of the writer, or her affective state, or an intended emotional communication. Progress towards a better understanding of opinionated text has been made by observing that the overall sentiment can often be decomposed into sentiments towards various aspects of an item (Lu et al., 2011). For instance, the overall appreciation of a restaurant expressed in a review can be a function of the food, the service, and the setting. The mining of such information enables the summarization of online reviews, which is helpful for semantic indexing and user access, and for the recommendation to users of specific items that satisfy their multiple-aspect preferences.

1.2 Challenges Addressed in this Thesis

The two research areas introduced above, namely recommender systems and sentiment analysis, handle complementary types of digital traces left by users. However, they have until recently progressed separately. Interestingly, there are relations between them which can provide benefits to information system design. For example, in social networks for reviewing products, the recovery of missing ratings used for recommendation can be achieved through sentiment analysis of text (McAuley et al., 2012). The modeling of latent factors in recommender systems from both text reviews and sets of ratings has been shown to be more beneficial than handling them individually (McAuley and Leskovec, 2013). Such studies highlight the need to consider both types of digital traces, namely explicit preferences such as ratings and implicit preferences expressed in natural language.

[4] Some examples are, respectively, Amazon, http://www.amazon.com/, IMDb, http://www.imdb.com/, Last.fm, http://www.last.fm/ and Google News http://news.google.com/.

In the context of these developments, this thesis identifies two major research challenges.

Challenge 1: Dealing with free-form text and unary preferences

Typically, when users review products online, they enter into an information system a textual review and a corresponding real-valued rating (e.g. from 1 to 5). These reviews have an explicit and informative nature which greatly facilitates the joint consideration of these two traces and the modeling of user preferences into recommendation systems. However, writing them requires considerable effort from users. On the contrary, there are other types of social networks, such as those resulting from user interaction on TED, Vimeo, Flickr, YouTube, Facebook, in which users provide a less detailed type of traces, which require much less effort, but are available in great quantities. In such social networks, users express their preferences in terms of unary feedback, such as liking or marking an item as favorite, which are not necessarily followed by corresponding text. This type of feedback reveals only what users liked, however what they do not like remains unknown, i.e. the notion of negative or fine-grained feedback is absent. At the same time, users may write free-form comments, which may or may not refer directly to items,by engaging with other users in discussions.

The arbitrary creation of comments and unary feedback makes the joint consideration of both types of digital traces more challenging. The first difficulty lies in inferring sentiment from free text and incorporating this information into recommender systems. The second difficulty lies in mapping sentiments inferred form text optimally to the unary feedback. Addressing this challenge has a potentially large impact on numerous large online social networks, as it extends considerably the information that can be leveraged for making recommendations.

Challenge 2: Mining user text to support the explanation of recommendations

Even though the joint modeling of user ratings and text has the potential to improve the performance of recommender systems, the output of such systems, i.e. the recommended items, is not easily interpretable by users. This happens because these systems are unable to explain or provide hints to the users about why a particular item is recommended to them. This lack of interpretability and transparency contributes negatively to the persuasiveness of the system and to the trust of users toward it, which may lead users to make less informed decisions or simply ignore the assistance provided by the system. Currently, solutions to gain trust from users rely on showing them ratings of similar users (Herlocker, 2000), shared content attributes with known user items (Tintarev and Masthoff, 2007), or confidence intervals of the models themselves (Hu et al., 2008). Still, the generated explanations do not take into account what users write about items in their texts. Consequently, it may be hard for the users to relate to explanations that are remote from their own beliefs and opinions.

Therefore, an important challenge is how to design systems with the ability to mine user texts in order to support recommendations with explanations. The difficulty lies in identifying segments of user texts which optimally represent users' opinions towards items in general, as well as opinions on individual aspects of items specifically. These segments can be used for explaining to users the recommendation of unknown items that share similar sentiment attributes with known items. Building such systems with interpretable and transparent functionalities will improve their effectiveness and, again, the overall user experience.

1.3 Contributions of the Thesis

To address the above challenges, this thesis proposes models of user sentiment and preferences which are effective, efficient and easily interpretable by humans, for information filtering systems. The thesis advances the state of the art relevant to Natural Language Processing and Machine Learning, and explores two different types of feedback, namely user preferences (see Definition 1 below) and user text (Definition 2). These types of feedback cover a wide range of options that are present in social networks, namely user comments and unary feedback such as likes and favorites, and user reviews followed by numerical feedback such as star ratings. In both cases, we advocate that by better understanding user text through mining its semantics, modeling its structure and leveraging its knowledge, we can design more interpretable models which improve information filtering but also can be explained to users.

Definition 1: *User preferences* are the user actions performed on items in a social network indicating their attitude towards them. We consider two types of preferences of user u for item i, noted r_{ui}. The first one is unary feedback, $r_{ui} \in \{1, ?\}$, i.e. 1 for presence of action and ? for absence of action. The second one is numerical feedback, $r_{ui} \in \{1, \dots, b\}$ i.e. a number from 1 to b.

Definition 2: *User texts* are the texts written by users in a social network, which may refer to items. We consider two types of such texts: comments, which are in free form, and reviews, which are accompanied by numerical feedback (e.g. from 1 to 5).

Figure 1.1 displays a diagram of the context and overview of this thesis: users interact with items in a social network, and find relevant items with the help of information filtering systems. Throughout this thesis, we focus on solving real-world information filtering problems using large datasets collected from online social networks such as TED, Flickr, Vimeo, BeerAdvocate, RateBeer, and Audible. These problems include recommending products or videos in personalized or non-personalized recommendation settings, predicting sentiment and emotion from text, summarizing and segmenting reviews, and identifying useful textual units in user texts.

Within this context of social networks, we aim to answer three main research questions:

1. How do item semantics help to predict unary preferences?
2. How do sentiments of free-form user texts help to predict unary preferences?

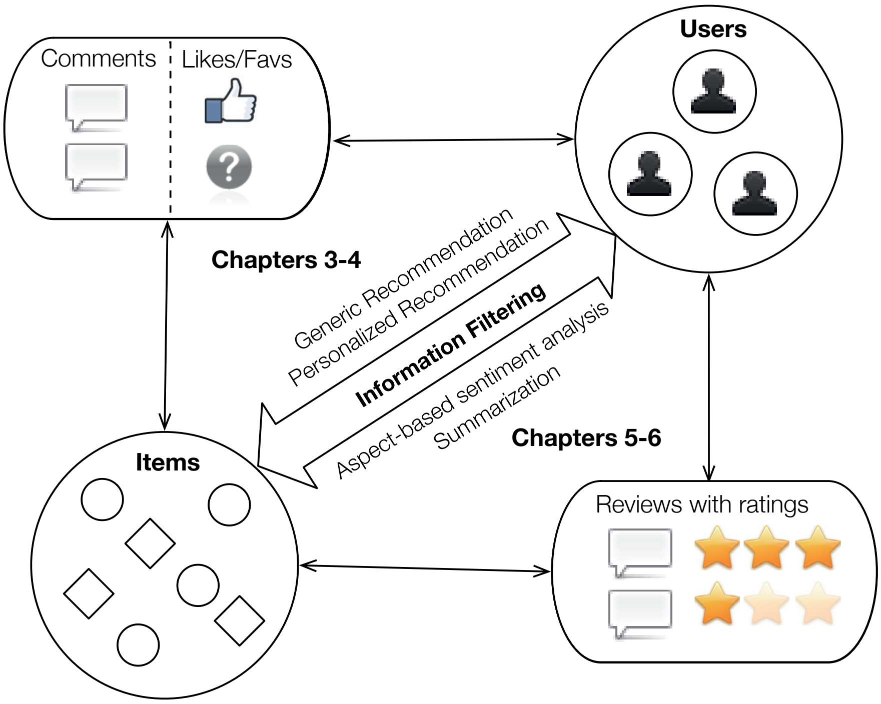


Figure 1.1: Diagram of the thesis context and overview. We consider users who interact with items in a social network through two different sets of actions: (a) comments and unary feedback such as likes, (upper left part) and (b) reviews and numerical feedback such as star ratings (lower right part). These digital traces of users are processed by an information filtering system (center) which enables users to find items that are relevant to them.

3. How to model fine-grained numerical preferences from user review texts?

We aim to provide principled yet testable answers to these questions, as well as to obtain insights on how to design better information filtering systems to improve user experience and access. The first two research questions address Challenge 1 by mining semantic information from free-form text to improve the learning of unary preferences. The third research question addresses Challenge 2 by modeling fine-grained preferences from user review text. To answer the first research question we will analyze the role of the information content of the items on predicting user preferences, and the level of difficulty of this prediction task when information from free-form user text is unavailable. To answer the second research question, which is complementary to the first one, we will investigate the importance of the sentiment information expressed in free-form user text and its influence on user preferences. Lastly, to answer the third research question, we will design models with interpretable components which are able to predict but also explain fine-grained preferences from user review text. In the following sub-sections, we briefly introduce the main contributions of this thesis.

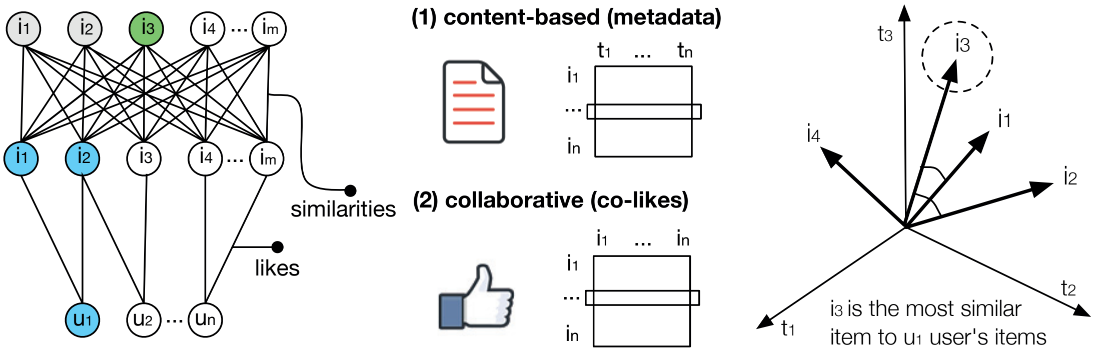

i1
i2
i3
i4
im
i1
i2
i3
i4
im
similarities
likes
u1
u2
un
(1) content-based (metadata)
t1 ... tn
i1
...
in
(2) collaborative (co-likes)
i1 ... in
i1
...
in
t3
i3
i1
i4
i2
t2
t1
i3 is the most similar item to u1 user's items

Figure 1.2: The proposed framework for generic or personalized recommendation. The similarity between items is based either on content or on collaborative information.

1.3.1 How item semantics help to predict unary user preferences

The recommendation of items to users can leverage the content descriptors of items or the information from the preferences of users or both. While in some domains, such as movie recommendation, user ratings are available on a large scale, in other domains such as lectures or videos these can be scarce. In Chapter 3, we investigate such domains with infrequent ratings – specifically, the domain of scientific lectures or courses, the content of which plays a significant part in deciding what to recommend. We present first a dataset that we created with lecture metadata downloaded and processed from the TED website (http://ted.com/) which contains more than 100k favorites and 200k comments made by about 70k users on 1,200 items[5]. Then, we propose a recommendation framework, represented in Figure 1.2, in which we compare content-based, collaborative-filtering, and combined methods for personalized recommendation (Pappas and Popescu-Belis, 2013a)[6]. On the left of the schema, a set of user nodes $U = \{u_1, \ldots, u_n\}$ are connected with the items $I = \{i_1, \ldots, i_m\}$ through 'like' actions. The items are in turn linked to other items through a similarity measure, defined over content-based or collaborative feature spaces, represented at the center. This framework allows to recommend the most similar items to a given item (generic recommendation), but also to recommend to each user the items that are most similar to the ones he/she has liked so far (personalized recommendation). For instance, on the right, for user u_1, the item i_3 (green node) is the highest ranking recommendation because it is the most similar to her liked items i_1 and i_2 (blue nodes) in the example feature space. The gray nodes i_1 and i_2 are not recommended since they are already known to the user u_1.

We show that among the content-based representations, the semantic-based methods (ESA, RP and LSI) provide more predictive descriptors of unary user preferences than keyword-based ones (TFIDF) when user feedback is absent, making them particularly applicable to

[5] The dataset is freely available at https://www.idiap.ch/dataset/ted/.
[6] 11th International Workshop on Content-Based Multimedia Indexing, 2013.

personalized recommendations in repositories into which new items are inserted frequently. A similar conclusion is drawn when we evaluate these methods on generic recommendations, i.e. recommending most similar items to a query item, with a ground truth defined by human experts (Pappas and Popescu-Belis, 2015)[7]. Moreover, we propose a combined method which utilizes content similarity and the popularity bias, and we show that it performs better than popularity bias alone and at close levels with pure collaborative filtering. To show the applicability of this study, we developed a content-based lecture recommender system in the context of the InEvent project[8]. Moreover, we participated in two international competitions on video annotation, search and hyperlinking, in both of which our team was ranked first (Bhatt et al., 2013a,b, 2014)[9].

1.3.2 How sentiments of free-form user texts help to predict unary preferences

Predicting unary preferences of users with content-based methods is beneficial when user feedback is scarce, but is less effective than collaborative filtering methods when user feedback is dense, as shown in our first contribution. In other words, content information itself cannot help to predict user preferences as precisely as the collaborative information derived from social networks. The second research question arising at this stage is: can we improve the prediction of user preferences based on the sentiment expressed in user comments, in addition to

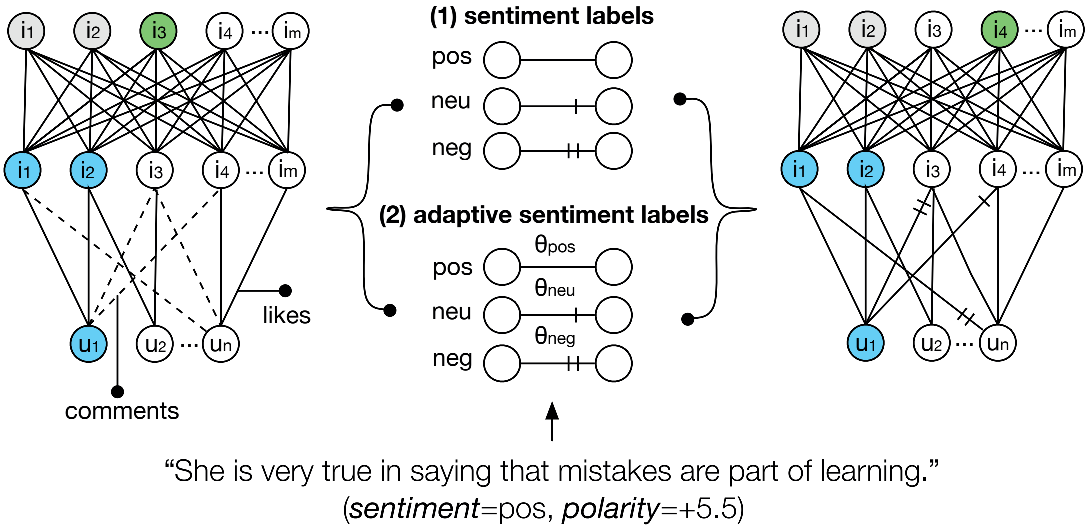

i1
i2
i3
i4
im
u1
u2
un
likes
comments
(1) sentiment labels
pos
neu
neg
(2) adaptive sentiment labels
θpos
θneu
θneg
"She is very true in saying that mistakes are part of learning."
(sentiment=pos, polarity=+5.5)

Figure 1.3: The proposed framework for sentiment-aware recommendation.

[7] Multimedia Tools and Applications Journal, 2015.

[8] European FP7 project InEvent: Accessing Dynamic Networked Multimedia Events, 2012–2014. Our recommendation demo is hosted at http://www.inevent-project.eu/demos/automatic-recommendation-of-lectures/, and its code is also used in the InEvent portal at http://www.inevent-project.eu/demos/inevent-portal/.

[9] MediaEval video search and hyperlinking task, 2013; ACM Multimedia Grand Challenge on video lecture annotation and search, 2013; paper presented at the ACM International Conference on Multimedia Retrieval, 2014.

the collaborative information? To answer this question, in Chapter 4, we propose a sentiment-aware neighborhood model which incorporates the sentiment from user comments to predict unary preferences (Pappas and Popescu-Belis, 2013b)[10]. Moreover, we investigate whether there are individual or collective opinion effects which influence the unary preferences of users by devising an adaptive algorithm that is able to map the sentiments of comments to one-class feedback, globally or per user (Pappas and Popescu-Belis, 2016a)[11].

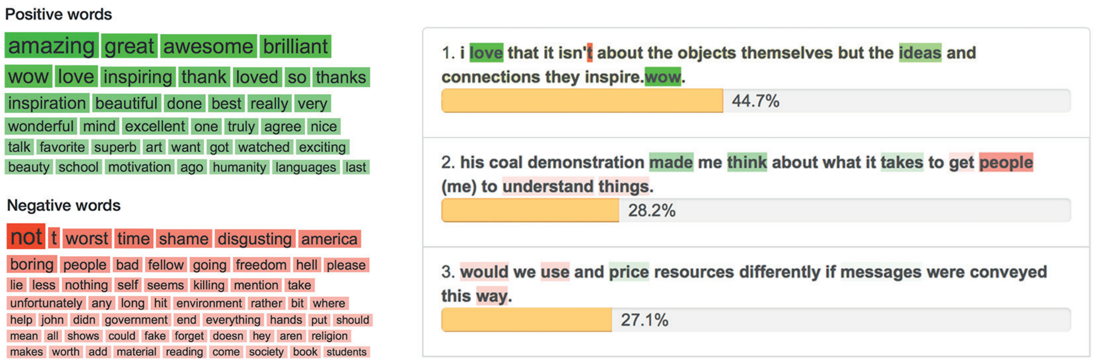
Figure 1.3 displays the framework proposed for leveraging the sentiment of comments to improve recommendation. On the left, in the graph of users and items, each solid edge represents an explicit feedback action (e.g. like) and each dashed edge represents an implicit action (e.g. comment) of a user towards an item. Our goal is to integrate the sentiment of comments (example shown at the bottom of the figure) with the unary preferences of the graph through fixed or learned mapping functions obtained by: (1) directly using the sentiment labels computed by a sentiment classifier, or (2) learning to adapt the sentiment labels to individual or collective unary preferences. In the latter case, the learned parameter values (θ_{pos}, θ_{neu}, θ_{neg}) indicate the importance of, respectively, positive, negative or neutral comments for the recommendation task. The parametrization enables us to capture complex patterns from the data and to reveal the underlying relationships between the sentiments of comments and the unary preferences. We compare the proposed models against state-of-the-art ones over three real-world datasets – lectures from TED, videos from Vimeo, and images from Flickr – demonstrating consistent improvements that and increase with the frequency of comments.

1.3.3 How to model fine-grained numerical preferences from user review texts

As shown in our second contribution, the sentiment information which lies in the user comments can improve the modeling of unary user preferences in addition to the collaborative filtering information. Although this approach enables systems to suggest new content that is actually relevant to users, the lack of explanations for the recommendations that are made, compromise the system's trustworthiness and transparency. As mentioned in Section 1.2, current methods for explaining recommendations systems are uninformed about why the users have particular preferences about items and their aspects. Such information can be found in social networks where users write reviews of products and rate them globally or according to specific aspects. Hence, there is a need to understand the fine-grained numerical preferences or opinions of users about items and how they are expressed in texts. Another related problem is how to summarize the user sentiments and how to identify sentences that are the most related to the global or aspect-specific ratings of items, both of which can be used for explaining recommendations of items that the user has not yet seen.

To address this review understanding challenge, we solve two important problems, namely learning to predict fine-grained numerical preferences of users from text and learning to

[10] Annual ACM SIGIR Conference on Research and Development in Information Retrieval, 2013.
[11] Expert Systems with Applications Journal, 2015.


Positive words
amazing great awesome brilliant
wow love inspiring thank loved so thanks
inspiration beautiful done best really very
wonderful mind excellent one truly agree nice
talk favorite superb art want got watched exciting
beauty school motivation ago humanity languages last
Negative words
not t worst time shame disgusting america
boring people bad fellow going freedom hell please
lie less nothing self seems killing mention take
unfortunately any long hit environment rather bit where
help john didn government end everything hands put should
mean all shows could fake forget doesn hey aren religion
makes worth add material reading come society book students

1. i love that it isn't about the objects themselves but the ideas and connections they inspire.wow.

44.7%

2. his coal demonstration made me think about what it takes to get people (me) to understand things.

28.2%

3. would we use and price resources differently if messages were conveyed this way.

27.1%

Figure 1.4: Sentiment analysis of a user comment using our multiple instance learning model which predicts the relevance of each sentence to the overall sentiment (here, the overall positive sentiment is mostly carried by the first sentence).

segment and summarize user reviews. Figure 1.4 displays an example of sentiment analysis of a user comment using the proposed models, which are presented hereafter. On the right, a positive comment (4 out of 5 stars) comprised of three sentences is displayed. The yellow bar shows the contribution of each sentence to the overall sentiment value. The highlighted words belong to the most predictive words in our dataset, a subset of which is displayed on the left, separating positive words (top, in green) from negative ones (bottom, in red).

Learning to predict the aspect ratings of reviews

Determining the aspect ratings of reviews is a challenging task, which may seem to require the engineering of a large number of features to capture each aspect. By treating a text globally, existing methods ignore that the sentences of a text have diverse contributions to its overall sentiment or to the sentiment towards a specific aspect of an item. In Chapter 5, we model the aspect sentiment expressed in text as a weakly supervised learning problem and we propose a new feature-agnostic solution based on multiple-instance regression (MIR) which assumes that not all the parts of a text have equal contribution to its rating (Pappas and Popescu-Belis, 2014)[12]. Thus, we aim for a finer-grained analysis of text meaning and for a better understanding of how aspects contribute to people's opinion.

We consider a set of m reviews, where each review (bag of sentences) B_i is represented by its n_i d-dimensional instances (sentences), i.e. $B_i = \{b_{ij}\}_{n_i}^{d}$. The set of the k aspect ratings per bag i is $Y = \{y_i\}_m^k, y_i \in \mathbb{R}^{\Bbbk}$. We model the sentence contribution to the ratings through saliency weights $\Psi = \{\psi_{ijk}\}_{n_i}^{1}$ per class k with unit constraints per review (see Figure 1.5). This allows us to represent each review in the feature space as an exemplar vector per class k with the set $X = \{x_{ik}\}_m^d$ obtained from the convex combination of the instances. By learning the model parameters we are able to predict more accurately the target scores, but also to reveal their relations to text structure at the phrase, sentence, or paragraph levels. Our evaluation on

[12] Conference on Empirical Methods in Natural Language Processing, 2014.

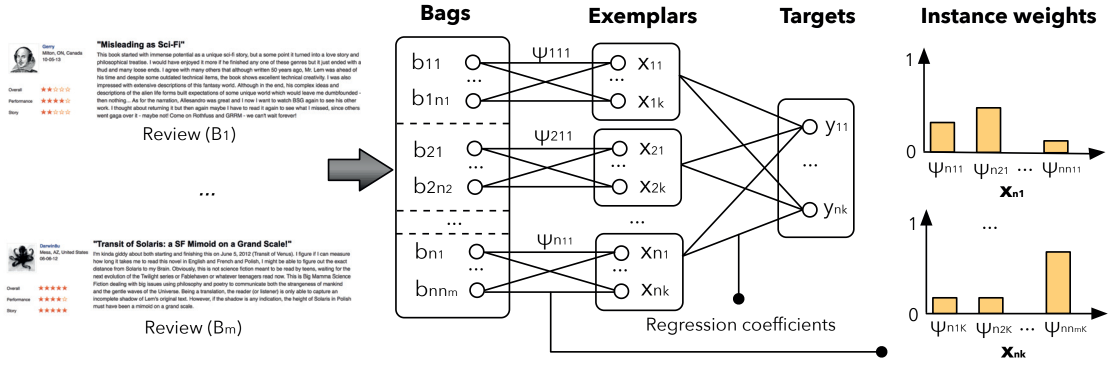


Figure 1.5: The proposed weighted multiple instance learning framework.

aspect rating datasets of restaurants, toys, books, beers, sentiment of comments, and emotion attributes of lectures, shows that weighted MIR consistently outperforms previous strongly and weakly supervised methods.

Learning to segment and summarize reviews

The above model provides interpretable saliency weights which represent the relevance of textual parts, such as sentences, paragraphs, documents or comments, to the target sentiment or emotion scores. In Chapter 6, we seek to evaluate these components in a quantitative manner, by assessing the value of the learned weights as features for three tasks pertaining to the understanding of reviews : (1) segmentation: identify the sentences of a review which discuss each ratable aspect; (2) summarization: choose the most likely sentence per aspect in a review; and (3) rating prediction: determine the attitude of the review toward each aspect. Using six large publicly available datasets with 1.3K–1.6M reviews and 3–6 aspects, our model learns saliency and relevance weights for the sentences of reviews. Then, we propose a Conditional Random Field (CRF) model which uses the learned saliency weights and sentiments of sentences to perform aspect-based segmentation and summarization of online reviews. The experimental results demonstrate that our model improves over or has similar performance to structured and unstructured state-of-the-art models, especially when using supervised learning across all examined datasets. Moreover, to show the applicability of this study, we trained our model on emotion attributes of lectures (such as 'fascinating', 'inspiring' or 'boring') using features from their transcripts to improve navigation and recommendation, as well as to summarize parts of the transcripts with respect to each emotion category, as shown in an online demonstrator implemented for the (aforementioned) InEvent project [13].

[13] http://www.inevent-project.eu/demos/emotion-based-analysis-and-recommendation-of-lectures/.

1.4 Plan of the Thesis

The thesis is organized as follows. In Chapter 2, we provide an overview of the published studies in areas related to our thesis, namely recommender systems, sentiment analysis, and multiple-instance learning, and we discuss the contributions of the thesis with respect to each of them. In Chapter 3, to answer the first research question above, we formulate the recommendation problem based on unary preferences as a one-class collaborative filtering problem. To solve this problem, we propose several content-based models which are based on semantic similarities, and combine them with the popularity of items for generic and personalized recommendation. In Chapter 4, to answer the second research question, we extract sentiment from user comments to improve one-class collaborative filtering using a sentiment-aware neighborhood model. To further improve the performance, we propose a learning algorithm which automatically adapts the sentiment of user comments to unary preferences. In Chapter 5, to answer the third research question, we concentrate on aspect-based sentiment analysis, which we cast as a weakly supervised problem. We propose a weighted multiple-instance learning method for solving this problem and then we analyze qualitatively its interpretability. In Chapter 6, to show that the learned saliency weights are useful indicators for opinionated text summarization and segmentation, we propose and evaluate a structured learning method which makes use of these saliency weights on several real-world datasets. In Chapter 7, the main achievements of the thesis are summarized and the proposed models are discussed from a broader perspective along with future directions for improvement.

2 Related Work

In this chapter, we provide a comprehensive overview of related studies spanning over three main research areas, namely recommender systems, sentiment analysis and multiple-instance learning. We analyze the merits and limitations of the existing studies and highlight the contributions of this thesis against this background. Section 2.1 surveys previous work on automatic recommendation, starting with a brief overview of content-based and collaborative filtering methods in Subsection 2.1.1. We review studies on semantic and multimodal information for recommendation respectively in Subsections 2.1.2 and 2.1.3. In Subsection 2.1.4, we review studies on top-N recommendation, and in Subsection 2.1.5, we discuss studies which have leveraged comments for various predictive tasks. Secondly, in Section 2.2, we provide background knowledge and an overview of studies on sentiment analysis. Particularly, in Subsection 2.2.1, we describe studies on predicting the sentiment of reviews, while in Subsection 2.2.2, we focus on the prediction of aspect ratings of reviews. In Subsection 2.2.3, we describe previous studies on segmenting and summarizing reviews, and in Subsection 2.2.4 we survey studies which use sentiment analysis to improve recommendation. Lastly, in Section 2.3, we provide background knowledge and review studies on multiple-instance learning, focusing on those which attempt to solve regression problems and text-based tasks.

2.1 Recommender Systems

Recommender Systems (RSs) are software tools and techniques which seek to identify and provide suggestions for useful or interesting items to a user among previously unseen ones (Ricci et al., 2010). The suggestions relate to various decision-making processes of the users, such as selecting which product to buy or which book to read, and they can be personalized or non-personalized (e.g. top popular books). These systems are primarily directed towards users who need to evaluate potentially overwhelming number of alternatives in order to make their decisions (data deluge problem).

2.1.1 Definitions and Main Recommendation Methods

In order to achieve its function, a recommender system must be able to predict the usefulness of an item. i.e. whether the item is worth recommending. The usefulness is estimated from available user profiles which contain feedback that users have provided to the system. There are two main categories of feedback that can be used by recommender systems, namely explicit and implicit feedback, as presented respectively in Definitions 3 and 4 below.

Typical recommender systems are built based on single criteria (overall rating for an item), but it is also possible to use multiple criteria (ratings on aspects or attributes of items) instead of one, namely with multi-criteria recommender systems (Adomavicius et al., 2011). Capturing explicit feedback is usually more demanding and time-consuming from a user perspective, though it is a more accurate source of information than implicit feedback. Hence, it often happens that ratings are missing from the collected data, especially for multi-criteria systems, which is detrimental to the quality of recommendation. In Chapters 4 and 5, we propose solutions to overcome this problem by inferring missing ratings from available free-form comments and reviews respectively.

Types of user feedback

Definition 3: *Explicit feedback* is a specific, direct and straight-forward way to capture user preferences, for instance by asking users to provide: (a) categorical ratings (e.g. awful, bad, indifferent, good, or amazing) or numerical ones (e.g. 1 to 5 stars); (b) unary or binary ratings (e.g. bookmarks, favorites or like/dislike); (c) preference rankings (e.g. ranking pairs or larger sets of items in a collection).

Definition 4: *Implicit feedback* is an indirect and less transparent way to capture user preferences, for instance through recording clicking and browsing behaviors, item viewing times or frequencies, or purchasing histories.

The prediction step may vary among recommendation algorithms but it can be described using a unifying model which represents the general role of recommender systems (Adomavicius and Tuzhilin, 2005). The degree of utility of an item i for a user u is modeled as a real valued function $r(u,i)$, exemplified in Equation 2.1 below. The goal is to predict the values of r over pairs of users $u \in U$ and items $i \in I$ (where the size of U is n and the size of I is m), i.e. to compute $\hat{r}(u,i)$ where $\hat{r}$ is the estimation of r. Once the estimations of utilities to a user u of a set of items $i_1,\ldots,i_N$ are computed, the system will recommend the top ranked items per user.

$$r(u,i) = \begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m,1} & r_{m,2} & \cdots & r_{m,n} \end{pmatrix} \text{ e.g. } \begin{pmatrix} 5 & 1 & ? & 2 \\ 1 & ? & ? & 5 \\ 5 & 1 & 4 & ? \\ 5 & 1 & 1 & ? \end{pmatrix} \text{ or } \begin{pmatrix} 1 & 1 & ? & 1 \\ 0 & ? & ? & 1 \\ 1 & 1 & 0 & ? \\ 1 & 1 & 0 & ? \end{pmatrix} \qquad (2.1)$$

Recommender systems commonly leverage either the characteristics of the items (content-based systems, see Lops et al. (2011)), or the user's social environment (collaborative filtering systems, see Koren and Bell (2011)) or both (hybrid systems). Below, we provide a brief overview of methods which belong to these three categories. More extensive overviews of content-based, collaborative filtering and hybrid methods, including techniques for their evaluation, have been provided by Sarwar et al. (2001); Ricci et al. (2010); Lops et al. (2011); Koren and Bell (2011).

Content-based (CB) methods recommend items that are similar to the ones that the user liked in the past (Magnini and Strapparava, 2001; Semeraro et al., 2009b; Lops et al., 2011). The similarity is calculated based on the features that represent items or are associated with them. For example, if a user has previously rated positively a movie that belongs to a specific genre, then the system will recommend items from this genre. The cosine similarity between feature vectors (e.g. tf-idf) that represent items can be used as item similarity. A key problem of these methods is that usually the system is limited to recommending content of the same type of what a user has already seen (Lops et al., 2011).

Collaborative Filtering (CF) is considered to be the most popular and widely used recommendation technique (Adomavicius and Tuzhilin, 2005; Sarwar et al., 2001; Herlocker et al., 2004; Schafer et al., 2007; Koren and Bell, 2011). The original implementation of this approach recommends to a user the items that the other users with similar tastes liked in the past (Schafer et al., 2007). The similarity of the tastes of two users is calculated based on the similarity of their previous rating histories. There are many extensions of the original approach for building CF systems, including neighborhood models, latent factor models such as singular value decomposition (Koren and Bell, 2011) or low rank matrix factorization (Takács et al., 2008; Koren et al., 2009; Ma et al., 2011). Latent factor models make it possible to address effectively user feedback, either explicit (Koren, 2008) or implicit (Hu et al., 2008), temporal dynamics (Koren, 2009), and rich content information. However, their predictive power comes with a high computational cost and the learned factors are not easy to interpret. On the contrary, neighborhood models are more scalable and interpretable because they perform predictions based on locality, but they generally have less learning capacity than latent factor models. The models designed in Chapters 3 and 4 (see specifically Sections 3.5.2 and 4.4.2) are built upon state-of-the-art neighborhood models.

The hybrid methods are based on combinations of the content-based with collaborative filtering ones mentioned above, to overcome some limitations of each approach. For example, *CF* methods suffer from the sparsity problem (i.e. they cannot recommend items that have not yet been rated), an issue that is not present in content-based approaches. Several ways of combining them have been proposed (Burke, 2007). Moreover, the context of the user can be also considered for recommendation, through a better personalization of the output of the system (Adomavicius and Tuzhilin, 2011). Ensemble methods, which are considered as hybrid methods, combine several individual predictors together to produce a final prediction. The performance of such methods is guaranteed to be greater than or equal to the best performing individual predictor. Interestingly, the winner of the Netflix competition was an ensemble of about 100 individual recommendation models (Bell et al., 2010). The Netflix Prize was an open competition awarding one million dollars for the best collaborative filtering algorithm to predict user ratings for films, held yearly from 2006 to 2008. The participants were not allowed to use any other information about users or films other than the provided ratings.

2.1.2 Mining Text for Semantic Representations of Items and Users

Semantic analysis enables learning accurate profiles of users and items thanks to references to external knowledge bases, such as ontologies or semi-structured encyclopedic knowledge. A recommender system can benefit from such analyses, which provide conceptual, linguistic and cultural background knowledge. Several research studies build semantic representations based on lexical semantic resources such as WordNet[1]. SiteIF (Magnini and Strapparava, 2001) is a personal agent for a multilingual news Web site using MultiWordNet[2] as an external knowledge source to model user interests. The ITem Recommender system or ITR (Degemmis et al., 2007; Semeraro et al., 2009a) can provide recommendations for items in several domains, using WordNet to build a document representation model called bag-of-synsets, which is an extension of the bag-of-words model (Semeraro et al., 2007). QuickStep (Middleton et al., 2004) is a system for the recommendation of on-line academic research papers using an ontology obtained from the DMOZ open directory project and semantically annotating documents using k-nearest neighbor classification. The main limitation of using ontologies is that they require significant human effort in their creation and maintenance.

Other semantic analysis approaches make use of semi-structured encyclopedic knowledge sources such as Wikipedia or the Yahoo! Web Directory. Leveraging such sources is beneficial, because they are constantly evolving, so they have up-to-date real world knowledge, and they are collaboratively curated by hundreds of thousands of people around the world. Wikipedia was used to estimate similarity between movies (Lees-Miller et al., 2008) in order to provide recommendations for the Netflix Prize competition by using a k-nearest neighbor and a pseudo-SVD algorithm. Smirnov and Krizhanovsky (2008) present an approach for filtering RSS feeds and e-mails which makes use of Wikipedia to automatically generate the user profiles from the user's document collection. Another approach which uses the WordSpace model and Wikipedia for content analysis was presented by Semeraro et al. (2009b). The dimensions of the WordSpace model represent semantic concepts and the points in the space represent documents (Sahlgren, 2006). In Chapter 3, we compare the features obtained using external knowledge from Wikipedia using Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007) with semantic methods obtained from in-domain data, namely dimensionality reduction and topic modeling methods (Hofmann, 1999; Blei et al., 2003; Chappelier and Eckard, 2009), to identify the most appropriate one for generic and personalized recommendation. A

[1] A lexical database for English, https://wordnet.princeton.edu/.

[2] A multilingual lexical database where English and non-English senses such as Italian, Spanish, Portuguese, Hebrew, Romanian and Latin are aligned, http://multiwordnet.fbk.eu/english/home.php.

detailed overview and analysis of topic models for text information access focusing on text classification and clustering is presented in (Chappelier, 2013).

2.1.3 Integrating Multimodal Information for Recommendation

Apart from using textual information to represent items, several authors have highlighted the need for integrating various modalities in the process of item recommendation where the 'modalities' include not only text, audio, or video channels of the items, but also their metadata, along with multimodal information from user interaction. MadFilm (Johansson, 2003) is a multimodal movie recommendation system that uses both modalities from natural language and direct manipulation. Arapakis et al. (2009a) proposed a multimodal video recommendation system, which predicts the topical relevance of a video by analyzing affective aspects of user behavior. Shin et al. (2009) designed a digital TV content recommendation system based on descriptive metadata collected from versatile sources. They used a combined multimodal approach which integrates classification-based and keyword-based similarity predictions. Mei et al. (2011) presented a contextual video recommendation system which was based on multimodal content relevance and user feedback based on visual, audio and textual information.

Arapakis et al. (2009b) proposed a multimodal recommender system which can predict topical relevance, by exploiting interaction data, contextual information as well as users' affective responses. Di Massa et al. (2010) used multimodal information from radio and television channels, websites, written and spoken content. The personal interests are inferred using natural language processing of the users' blogs. Latent semantic analysis was used to find relationships between user's interests and items to recommend. Yang et al. (2007) presented a video recommendation system based on multimodal fusion and relevance feedback. They defined the multimodal relevance as a textual, visual and aural relevance and calculated the different intra-weights for each modality and inter-weights among them.

This thesis differentiates from previous studies, as it focuses on integrating two user feedback modalities, namely user comments and unary ratings, in Chapter 4. Even though our methods do not include the processing of audio and video modalities, they could benefit from them in the future, as discussed in the perspectives of the thesis in Section 7.2.

2.1.4 Top-N Recommendation Task

In contrast to mainstream recommender systems that aim to predict numerical ratings for each item, *top-N recommender systems* are used to recommend N items that are most likely to be of interest or appealing to users (Cremonesi et al., 2010), as in Chapters 3 and 4 of this thesis. Common evaluation methods based on error metrics are not a natural fit for evaluating the top-N recommendation task, rather it can be directly measured by alternative metrics such as accuracy metrics (Cremonesi et al., 2010). Top-N recommendation systems operate on both discrete and real-valued feedback, although they are mostly applied to unary feedback obtained from user behavior data, because in this case numerical rating prediction is difficult (Schwab et al., 2000). The recommendation problem with unary feedback is also known as one-class CF problem Pan et al. (2008). The CF methods for top-N recommendation can be broadly divided in two categories: memory-based vs. model-based ones (Deshpande and Karypis, 2004). These methods typically originate from traditional CF recommendation methods (e.g. Koren and Bell, 2011; Lops et al., 2011) tailored to the top-N task, as we will explain in Section 3.5. However, a combination of the two is possible, for example with neighborhood models that are able to learn bias effects and interpolation weights from data, as proposed by Koren (2008). Our sentiment-aware neighborhood model defined in Section 4.4 belongs to this type of methods, although with a different kind of parametrization, namely bias effects and sentiment mapping weights.

Many studies have focused on model-based methods for top-N recommendation. Hu et al. (2008) adapted CF to datasets with implicit feedback by considering positive and negative preferences with varying confidence levels, which they used to provide explanations. Ning and Karypis (2011) proposed sparse linear methods (SLIM) to generate top-N recommendations by solving a regularized optimization problem. Other studies have formulated top-N recommendation as a ranking problem. Rendle et al. (2009) adopted a Bayesian perspective and proposed an optimization criterion, named Bayesian Personalized Ranking (PBR). Shi et al. (2012) proposed an approach called collaborative less-is-more filtering (CLiMF) which directly maximizes the Mean Reciprocal Rank for top-N recommendation with binary relevance data. The same authors generalized CLiMF for multiple levels of relevance (Shi et al., 2013). Kabbur et al. (2013) reduced the sparsity of the datasets for top-N recommendation by learning an item similarity matrix using structural equation modeling. Aiolli (2014) optimized the Area Under the Curve (AUC) within a maximum margin framework for CF top-N recommendation. Elbadrawy and Karypis (2015) proposed a sparse factor model, which learns feature-based item similarity models that are able to exploit global and user-specific preferences.

Pan et al. (2008) formulated the one-class CF problem as dealing only with positive instances of user feedback. The inherent problem of one-class CF is the lack of explicit negative feedback, in other words the uncertainty of the class to which an unknown rating belongs. An approach that is commonly used for one-class CF problems (see also Section 2.1.4) is to make an assumption about the distribution of the negative class. Several schemes were proposed by Pan et al. (2008) to weigh the negative class in a discriminative fashion, formulated within a matrix factorization framework. These weighting mechanisms performed better than the baseline assumptions that treat all the missing instances as negative or unknown. We obtained similar results with our proposed assumption for balancing the missing instances in Section 4.4.3 below. Sindhwani et al. (2009) suggested to treat zero-valued pairs as optimization variables computed from the training data. Thus, instead of making a uniform assumption about the negative class, the distribution of the negative class was learned. Li et al. (2010c, 2014) incorporated rich user information to improve one-class CF, such as search history, purchasing and browsing activities. Paquet and Koenigstein (2013) addressed the lack of a negative class using a Bayesian generative model for the latent signal with an unobserved random graph which connects users with items they might have considered. Yuan et al. (2013) considered the richer user and item content information, such as user posts and item titles, abstracts, authors and keywords, for better weighting the unknown data.

In Chapter 4, we solve the top-N recommendation problem through a ranking function based on an adaptive sentiment-aware neighborhood model, which uses both user comments and unary ratings. The proposed model is a significant extension of item-based CF models such as those proposed by Cremonesi et al. (2010) and Koren and Bell (2011). To complement unknown ratings, we propose to infer user ratings from free-form user comments, which occur frequently in online repositories and social networks. Instead of hypothesizing the values of missing instances, we attempt to infer some of them from available textual data, and demonstrate the value of such information in combination with three different assumptions about missing instances.

2.1.5 Leveraging User Comments for Recommendation

User comments in online communities have captured the attention of researchers due to their high availability, and the personal, opinionated and rich information they contain. Many predictive tasks and applications have benefited from the analysis of textual user comments, though most of the applications are not directly related to recommender systems, since they aim to predict the popularity or mood of news articles, blogs, or user profiles The few existing studies of comments for recommender systems do not target personalized recommendation in the one-class setting – despite the importance of this setting, emphasized in this thesis. Pavlou and Dimoka (2006) utilized content analysis to quantify comments from sellers on a popular online auction website and to match them with purchasing data from buyers that had transacted with them. The addition of text comments to numerical ratings helped to explain a greater part of the variance in seller's benevolence and credibility compared to ratings only. Li et al. (2007) proposed to include comments on blog posts for clustering blogs and found that they increased discriminative effects compared to using only the blogs' contents. Tsagkias et al. (2009) hypothesized that the number of user comments on a news article may be indicative of its importance and attempted to predict the volume of comments on an article prior to its publication as a binary classification task (high or low volume).

More recent studies of user comments have *refined the above trends on news and profiling*. They include ideas such as: comparing several text analysis strategies to automatically gather profile data from user comments on news articles (Messenger and Whittle, 2011); predicting the popularity of online articles during a short observation period using a simple linear prediction model (Tatar et al., 2011); predicting the political orientation of news stories (Park et al., 2011); exploiting the mood of tweets to predict stock market time series (Bollen et al., 2011); improving social tag recommendation by connecting user comments with tags (Yin et al., 2013); predicting user affective comments based on image content (Chen et al., 2014, 2015); analyzing the influence of Facebook user comments on relationship status updates (Ballantine et al., 2015); and detecting hate speech in online user comments by learning distributed low-dimensional representations of comments and using them as features for classification (Djuric et al., 2015).

The studies which have a similar goal to ours in Chapter 4, namely leveraging user comments for content recommendation, exhibit a number of significant differences with our approach: they focus on the *generic (i.e. non-personalized) recommendation of tags, comments, or news for commenting*, emphasizing their *semantic content over their polarity*. As also confirmed by the literature review of Sun et al. (2014), studies of personalized recommendation from one-class feedback augmented with comments remain scarce. For instance, Wang et al. (2010b) presented a framework for non-personalized news recommendation which is based on constructing news topic profiles based on their text description and along with associated reader comments. Wang et al. (2010c) and Li et al. (2010b) used structural, semantic and authority information from user comments to improve recommendation. Agarwal et al. (2011) attempted to rank the comments associated with a news article according to personalized user preferences, i.e. liking or disliking a comment. Shmueli et al. (2012) presented a model that predicts news stories that are likely to be commented by a given user. Kim et al. (2012) proposed a query expansion method that utilizes user comments in order to consider user's different preferences in finding movies. San Pedro et al. (2012) analyzed the user comments to detect opinions about the aesthetic quality of images for image search. Jain and Galbrun (2013) proposed to organize user comments in semantic topics which enable users to discover significant topics of discussions in comments and allow to explicitly capture the immediate interests of users on news articles. In one of the few studies on multimedia content, Siersdorfer et al. (2010) analyzed dependencies between comments, views, comment ratings, topic categories and comment sentiment influence in a large dataset from Youtube, to predict comment ratings, i.e. number of feedback votes on comments.

The only other study apart from the one presented in this thesis on *sentiment analysis for one-class CF over multimedia content* is the study of Sun et al. (2014), to which we compare our scores in Section 4.8.5, Table 4.13 below. This study uses the metadata set that we created from the TED talks (Pappas and Popescu-Belis, 2013a, 2015) and made available online (see Section 4.2), though not the Flickr and Vimeo datasets additionally used in Chapter 4. Sun et al. (2014) leverages sentiment information extracted with ensemble classifier from user comments to improve one-class CF task. Moreover, they showed that a matrix factorization framework, which reaches higher recommendation scores than neighborhood models, can be combined with them to increase performance.

2.2 Sentiment Analysis

Sentiment analysis is one of the major topics researched in NLP, and its goal is to determine the overall attitude of a writer or speaker, or their attitude towards specific topics, from their texts or speeches respectively. The attitudes of the speakers or writers have several components. According to appraisal theory in psychology, they may refer to their own assessment or judgment (expressed), to the assessments of listeners or readers (perceived), to their affective state when writing or talking, or to the intended emotional communication. Opinions or sentiments are essential to human activities because they are able to influence our behaviors and decisions. This fact, along with the recent growth of social media content and user activity, has created an increased commercial interest for marketing, public relations and political campaigns. Consequently, sentiment analysis has nowadays gained a lot of popularity because of its wide and pervasive application to real-life problems. Another factor which contributed to its popularity is that it is a challenging research problem in NLP, and covers a variety of sub problems as we describe below.

Types of sentiment analysis

Definition 5: At the *document level*, the task is to classify whether a whole opinionated document has a positive, negative or neutral sentiment.

Definition 6: At the *sentence level*, the task is to classify whether an individual sentence has a positive, negative or neutral sentiment.

Definition 7: At the *fine-grained level*, the task is to classify the sentiment of individual sentences or phrases inteded towards certain entities or aspects.

Below we provide an overview of the methods which have been used for predicting sentiment of review, including aspect-level sentiment, followed by methods for segmentation and summarization of reviews. Lastly, we provide an overview of studies which have used sentiment analysis to improve recommendation.

2.2.1 Predicting the Overall Sentiment of Reviews

Sentiment analysis typically aims to detect the polarity of a given text, and is commonly formulated as a classification problem for discrete labels such as ‘positive’ and ‘negative’ or a regression one for real-valued labels (Pang and Lee, 2005, 2008). Pang and Lee (2008) survey the large range of features that have been engineered for rule-based sentiment analysis methods as the one used in Chapter 4 of this thesis (Hatzivassiloglou and Wiebe, 2000; Hu and Liu, 2004; Wilson et al., 2005) and for corpus-based ones (Pang et al., 2002; Thomas et al., 2006). Machine learning techniques for sentiment classification have been introduced quite early (e.g. Pang et al., 2002), including unsupervised techniques based on the notion of semantic orientation of phrases (e.g. Turney, 2002). A related family of studies focused on subjectivity detection, i.e. whether a text expresses opinions or not (Wiebe et al., 2004).

A large volume of research focused on the prediction of sentiment, i.e. the overall rating of a review. Most of these studies have focused on feature engineering (Pang and Lee, 2008), and more recently on feature learning (Maas et al., 2011; Socher et al., 2011; Tang et al., 2014), including the use of deep neural networks (Socher et al., 2013; Mikolov et al., 2013; Tang, 2015). These methods do not require careful engineering of features to go beyond state-of-the-art alternatives, however the learned features which capture semantic information are difficult to be interpreted by humans. Deep learning, to our knowledge, has not been applied yet to aspect rating prediction, but as it would likely not provide interpretable features, we advocate here the use of a different type of learning methods, namely multiple-instance ones (MIR, see Section 2.3), to capture high-level semantics in terms of aspect saliency in Chapter 6.

The above methods treat the sentiment analysis problem as a strongly supervised learning problem, which assumes that the target labels correspond to the entire input text. In Chapter 5, we put forward a new formulation for the problem based on weakly supervised learning which assumes that parts of a text have unequal contributions to its target label.

2.2.2 Predicting the Aspect Ratings of Reviews

The fine-grained analysis of opinions regarding specific aspects or features of items is known as *multi-aspect sentiment analysis*. This task usually requires aspect-related text segmentation, followed by prediction or summarization (Hu and Liu, 2004; Zhuang et al., 2006). Most attempts to perform this task have engineered various feature sets, augmenting words with topic or content models (Mei et al., 2007; Titov and McDonald, 2008; Sauper et al., 2010; Lu et al., 2011), or with linguistic features (Pang and Lee, 2005; Baccianella et al., 2009; Qu et al., 2010; Zhu et al., 2012). Other studies have advocated the joint modeling of multiple aspects (Snyder and Barzilay, 2007) or of multiple reviews for the same product (Li et al., 2011). McAuley et al. (2012) introduced new corpora of multi-aspect reviews, which we also partly use here, and proposed models for aspect detection, sentiment summarization and rating prediction. Lastly, joint aspect identification and sentiment classification have been used for aggregating product review snippets by Sauper and Barzilay (2013).

Previous studies on aspect rating prediction have used for training segmented text obtained automatically (Zhu et al., 2012; McAuley et al., 2012), or have modeled the relationships between different aspect ratings (Lin and He, 2009; Gupta et al., 2010; McAuley et al., 2012) to go beyond standard supervised models such as SVM with bags-of-words. To our knowledge, none of the previous studies considered in their modeling the weak relationship between text labels and the parts of texts (e.g. sentences) as we propose in Chapter 5. Moreover, in Chapter 6, the proposed MIR method is trained over the entire unsegmented text, reducing the computational cost and human intervention which is required to obtain segmented text. In addition, MIR captures meaningful structural information of the input text, instead of the output labels only, and thus provides interpretable sentence saliency weights, which can be used for segmenting and summarizing reviews.

2.2.3 Segmenting and Summarizing Reviews

Most studies of review segmentation and summarization focused mainly on unsupervised learning (Titov and Mcdonald, 2008; Zhu et al., 2009; Wang et al., 2010a; Brody and Elhadad, 2010; Lu et al., 2011), while fewer studies explored supervised learning (Li et al., 2010a). Recently, the availability of annotated data (McAuley et al., 2012; Pontiki et al., 2014) has increased the interest in supervised methods, e.g. with constrained structured models (McAuley et al., 2012), or with linear chain CRF models (Patra et al., 2014; Hamdan et al., 2015). While sentence sentiment has been shown to be useful for inferring sentence aspects (Brody and Elhadad, 2010; Ganu et al., 2009), the aspect saliency and sentiment of sentences from in-domain corpora have not been considered before. In Chapter 6, we augment common word or topic spaces with such high-level semantics.

2.2.4 Improving Recommendation through Sentiment Analysis

Since their appearance, sentiment analysis techniques have attracted the interest of the research community because they help capturing high-level meaning in language and offer a wide variety of applications. Several studies have performed sentiment analysis of textual reviews of items to improve recommendation. Most of them focus on *learning to infer numerical ratings* from a set of already labeled textual reviews (arguing in favor or against particular items), unlike the free-form unlabeled comments that are exploited in this thesis (Chapter 4). Leung et al. (2006, 2011) proposed a probabilistic rating inference framework which mines user preferences from text reviews and then maps them onto numerical rating scales. Similarly, Kawamae (2011) proposed a hierarchical topic modeling approach for integrating sentiment analysis with CF by modeling each author's preference and writing attitude as latent variables. Such frameworks provide a convenient way of combining feedback from preferences and from reviews, but their main drawback is that they are only applicable to review websites, and cannot be easily transferred to other situations. Singh et al. (2011) performed two-stage filtering with CF and sentiment classification of user reviews, keeping however the modeling of text and ratings separate, unlike our model which optimizes their combination.

Moshfeghi et al. (2011) addressed the *cold-start problem* by considering item-related emotions and semantic information extracted from movie plots as well as text reviews, using LDA and gradient boosted trees. The benefits of this method were mostly observed when the amount of user ratings was very small or zero. In cases when user ratings are unavailable, Zhang et al. (2010, 2013) proposed to perform online video recommendation by using virtual ratings extracted from sentiment analysis of text reviews, instead of actual user ratings. Similarly, when ratings are absent, Karampiperis et al. (2014) examined the benefits of using sentiment analysis on user review comments followed by explicit numerical ratings to improve recommendation in educational repositories. In contrast, we will show that our proposed model in Chapter 4 is beneficial on various proportions of free-form user comments.

Several recent methods have focused on situations were both *review text and ratings are* *available.* Pero and Horváth (2013) proposed a simple, scalable and effective rating prediction framework based on matrix factorization which utilizes both user ratings and opinions inferred from their reviews. García-Cumbreras et al. (2013) categorized users according to the average polarity of their comments, in the context of movie reviews. These categories were then used as features to improve CF models, thus following a less personalized and item-oriented recommendation strategy than ours. McAuley and Leskovec (2013) and Ling et al. (2014) combined latent rating dimensions (such as those of latent-factor recommender systems) with latent topics of reviews learned by topic models. Similarly, Diao et al. (2014) proposed a probabilistic model based on CF and a topic model, which jointly captures the interest distribution of users and the content distribution for movies. The advantage of such models, apart from their improved accuracy, is that the learned latent dimensions can be more easily interpreted than pure latent-factor models. Zhang et al. (2015) first extracted hidden dimensions from reviews with topic modeling, and then applied a traditional CF model to capture correlations between hidden dimensions in reviews and ratings. All these studies used explicitly labeled reviews for evaluation, therefore it is unclear whether their improvements still hold when using free-form comments and a more challenging recommendation setting such as with unary ratings, as in our study presented in Chapter 4.

A promising line of sentiment analysis research, for structured reviews, is to recognize the *aspects of items and their ratings.* For instance, Ganu et al. (2009) proposed a regression approach which considers various aspects of a restaurant to improve recommendations using a k-NN method. Similarly, Jakob et al. (2009) proposed to extract movie aspects for improving movie recommendations in a CF model. Faridani (2011) generalized the concept of sentiment analysis of reviews to multiple dimensions (such as service or price) using Canonical Correlation Analysis, with applications to product search and recommendation. Levi et al. (2012) addressed the cold-start problem by mining aspects and their sentiment, and profiling users according to their intent and nationality using context groups extracted from reviews.

Personalization of quality rankings for products using aspect information from reviews was investigated by Musat et al. (2013), who also proposed new evaluation methods to rate explanations and to predict pairwise user preferences. Zhang et al. (2014) extracted attribute-value pairs from product reviews and integrated them into a latent matrix factorization model, resulting in an explicit factor model which is able to provide explanations of its recommendations in terms of aspects preferred by users. To capture the importance given by different users to different items, Nie et al. (2014) used tensor factorization to automatically infer the weights of different aspects in forming the overall rating. D'Addio and Manzato (2015) proposed a vector-based representation of items computed from user reviews, which considers the sentiment of those reviews towards specific aspects, within a neighborhood-based CF model. Wu and Ester (2015) proposed a unified probabilistic model which combines the advantages of CF and aspect-based opinion mining to learn personalized sentiment polarities on different aspects of items. He et al. (2015) proposed to cast the recommendation task as vertex ranking and devised a generic personalized algorithm for ranking in tripartite graphs of user-item-aspect relations, named TriRank. To be applicable, these methods require even more demanding explicit feedback information, namely repositories which include multiple-aspect reviews and aspect-specific ratings, which are currently available only for limited range of item types.

Most of the above studies aim to predict ratings (on numeric scales) from reviews, generally by training the predictor on similar reviews that are accompanied by ground-truth ratings given by their authors. Unlike such studies, we will analyze in Chapter 4 of this thesis the sentiment of user comments which are never accompanied by ratings. Free-form comments differ from text reviews as they are not necessarily purposed to refer to the items which are considered for recommendation, due to their unconstrained and spontaneous nature. Rather, they reflect the written interactions among the users of an online community. In addition, the existing approaches which make use of reviews composed of ratings and text have a high adaptation cost to a new domain if no ground-truth ratings are initially available. Another novelty presented in Chapters 3 and 4 of this thesis is that, unlike previous work on this topic, we consider explicit user feedback in the form of unary ratings, which is a common form of feedback in social networks such as YouTube, Facebook, Flickr, Vimeo, Twitter and others.

2.3 Multiple Instance Learning

We turn now to a specific category of machine learning which will be used in Chapters 5 and 6 of this thesis, namely multiple-instance learning (MIL). MIL was originally proposed by Dietterich et al. (1997), and is able to deal effectively with 'weak' or 'coarse-grained' input labels. The MIL algorithms receive as input a set of labeled bags, each of which contains a variable number of instances, instead of individual labeled instances, as in traditional supervised learning. The goal of MIL is either (a) to be able to learn a classifier which assigns correct labels to individual instances or (b) predict the labels of the bags without necessarily inducing the labels of the individual instances. Consider the simple example of a MIL problem shown in Fig. 2.1, adapted from Dietterich et al. (1997) by Babenko (2009). There are several faculty members, and each owns a key chain that contains a few keys, but only some of them are able to open a certain room, and some are not. The goal is then to predict whether a certain key or a certain key chain can get us into this room. To solve this we need to find the key that all the key chains which can get us into the room ("positive") have in common. By finding which key is the correct one (here the 3rd and 2nd key in Sanjoy's and Lawerence's key chain respectively), we can also correctly classify an entire key chain – either it contains the required key, or it does not.

The solution of the above MIL problem is based on one of the first assumptions proposed, which is the primary instance assumption, i.e. that there is a single instance called 'primary' which is responsible for the label of the bag. Foulds and Frank (2010) surveyed the various MIL assumptions and concluded that they are differently suited to different tasks, and should be stated clearly when describing a model. MIL has been successfully applied to a variety of domains such as image classification, molecule classification for drug discovery, drug activity prediction, remote sensing (e.g. crop yield prediction) and text or document categorization.

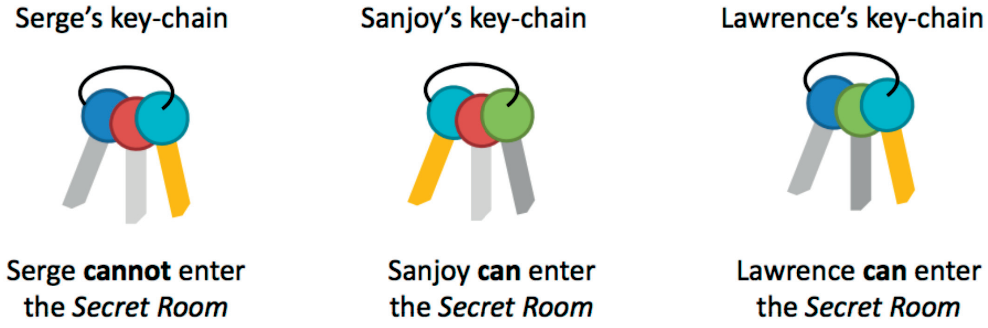

Serge's key-chain
Sanjoy's key-chain
Lawrence's key-chain
Serge **cannot** enter the *Secret Room*
Sanjoy **can** enter the *Secret Room*
Lawrence **can** enter the *Secret Room*

Figure 2.1: Key-chain example: each faculty member owns a key chain which can or cannot open a certain room. How to predict which key or key chain open the room?

The majority of the MIL studies focused on classification (Andrews et al., 2003; Bunescu and Mooney, 2007; Settles et al., 2008; Wang et al., 2011; Doran and Ray, 2014; Cheplygina et al., 2015; Zhu et al., 2015), while fewer focused on regression (Ray and Page, 2001; Davis et al., 2007; Wagstaff et al., 2008; Wagstaff and Lane, 2007; Wang et al., 2012). A comprehensive review of multiple-instance classification along with comparison of major methods is presented by Amores (2013).

Multiple-instance regression (MIR) belongs to the class of MIL problems for real-valued output, and is a variant of multiple regression where each data point may be described by more than one vector of values. Related to document analysis, several MIR studies have focused on news categorization (Zhang and Zhou, 2008; Zhou et al., 2009) or web-index recommendation (Zhou et al., 2005) but, to our knowledge, no study has attempted to use MIR for text regression tasks with real-valued labels such as aspect rating prediction, sentiment and emotion prediction, as we do in Chapters 5 and 6 of this thesis.

MIR was firstly introduced by Ray and Page (2001), proposing an EM algorithm which assumes that one primary instance per bag is responsible for its label. Wagstaff and Lane (2007) proposed to simultaneously learn a regression model and to estimate instance weights per bag for crop yield modeling; however their method is not applicable to prediction. A similar method which learns the internal structure of bags using clustering was later proposed by Wagstaff et al. (2008) for crop yield prediction, and we will use it for comparison in Chapter 5. Later, the method was adapted to map bags into a single-instance feature space by Zhang and Zhou (2009). Wang et al. (2008) assumed that each bag is generated by random noise around a primary instance, while Wang et al. (2012) represented bag labels with a probabilistic mixture model. The main disadvantages of the above methods for text regression tasks is that they do not scale well to high-dimensional feature spaces, and that some of them are not applicable to prediction.

In Chapter 5 of this thesis, we propose the first weighted MIR model (to our knowledge) suited for text regression tasks, and evaluate it on three such tasks, namely aspect rating prediction, sentiment prediction and emotion prediction. Moreover, in Chapter 6 we analyze the quality and the utility of the learned MIR features by using them as features for review understanding tasks such as review segmentation and summarization. Perhaps the most related work to our own on sentiment analysis with MIL is the study from Kotzias et al. (2015). The authors combined MIL with deep learning features and applied it on sentiment prediction with the goal of transferring label information from group labels (review) to instance labels (sentences). However, this study focused solely on sentiment classification rather on regression, and it did not take into account the sentiments towards aspects.

This thesis attempts to overcome the aforementioned limitations of previous studies and to build upon the contributions from a variety of domains that have been presented in this chapter. Specifically, many of our proposals will benefit from the leveraging of knowledge from certain domains for application to others. Examples of this transfer include the use of sentiment analysis as means to extract preference information from text and thus improve recommendation, and the use of multiple-instance learning to model weakly-supervised text regression tasks such as sentiment or aspect-based rating prediction.

3 Semantic Analysis of Item Content for Recommendation

Understanding the importance of content for predicting user preferences is a challenging problem. Knowing which attributes and representations of items are more predictive of preferences, enables a system to make relevant and accurate recommendations in large and constantly growing repositories of items, even when user preferences are scarce, i.e. in a cold-start setting. In this chapter, we attempt to address this problem by introducing a novel dataset and by comparing several methods for the recommendation of non-fiction audio visual material, namely lectures from the TED website. The TED dataset contains 1,149 talks and 69,023 profiles of users, who have made more than 100,000 ratings and 200,000 comments. The corresponding metadata, which have been made available, can be used for training and testing generic or personalized recommender systems.

We define content-based, collaborative, and combined recommendation methods for TED lectures and use cross-validation to select the best parameters of keyword-based (TF-IDF) and semantic vector space-based methods (LSI, LDA, RP, and ESA). We compare these methods on a personalized recommendation task in two settings, a cold-start and a non-cold-start one. In the cold-start setting, semantic vector spaces perform better than keywords. In the non-cold-start setting, where collaborative information can be exploited, content-based methods are outperformed by collaborative filtering ones, but the proposed combined method shows comparable performances to the latter ones, and can be used in both settings. For the generic recommendation task, LSI and RP again outperform TF-IDF.

3.1 Introduction

The recommendation of multimedia content to users can leverage either the content descriptors (*content-based methods, CB*) or information from the preferences of users (*collaborative filtering, CF*) or both types of information (*hybrid systems*). While in some domains, such as movie recommendation, user ratings are available on a large scale, as in the Movielens data with millions of ratings, in other domains these can be scarce.

In this chapter, we compare recommendation techniques for lecture recordings, that is, non-fiction audiovisual material with informative purposes, the content of which plays a significant part in deciding what to recommend. We compare the merits of CB and CF methods and propose a new method for combining semantic features (based on distances in semantic vector spaces) with user preferences, which are defined as the list of recordings explicitly marked as favorites, following common practice in recommender systems (Shani and Gunawardana, 2011). Following appropriate training to identify the best performing features, we show that CB recommendation using Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007) is the best performing method in a cold-start setting, when no user preferences are known, including the case of anonymous viewers. In a non-cold-start setting, we will show that pure CF methods perform best, but only slightly above the combined CB and CF method with keyword-based distance, showing the importance of using content features in both settings.

The methods will be tested on a new dataset acquired from the TED repository of lectures on scientific and social topics. We will show how this dataset can be used for the evaluation of lecture recommendations, given its rich content and metadata (to be used as features) along with explicit feedback from users (to be used as ground truth for training and testing). Our results thus constitute the first benchmark scores on this promising data set, which we made public.

This chapter is organized as follows. We introduce the TED dataset and the metadata we extracted from it in Section 3.2. Then, we define the generic and personalized recommendation tasks that can be tested using this data in Section 3.3. We present semantic vector spaces in Section 3.4 and use them to define CB recommendation methods, as well as combined CB + CF ones, in Section 3.5. The results of feature selection are given in Section 3.6, while results over test data are given in Section 3.7 for personalized recommendations, and in Section 3.8 for generic ones, i.e. for anonymous users.

3.2 The TED Collection: A Dataset for Recommendation Evaluation

The TED website is the online repository of audio-visual recordings of the popular TED lectures given by prominent speakers (see www.ted.com). The recordings and the metadata accompanying them are made available under a Creative Commons non-commercial license. The website provides extended metadata as well as user-contributed material such as discussion threads related to the talks. The TED speakers are scientists, writers, journalists, artists, and businesspeople from all over the world who are generally given a maximum of 18 minutes to present their ideas. The talks are given in English and are usually transcribed and then translated into several other languages by volunteers. The quality and interest of the talks has made TED one of the most popular online lecture repositories. An important characteristic of TED is that the metadata for the audio visual content is human-made.

In Figure 3.1 an example of a TED talk page is shown. On the left, the main audio visual player which displays the talk is at the top, just below the speaker's name and title of the talk. On the

	Total	**Talk**		**User**		**Active user**	
Attribute	Count	Avg	Std	Avg	Std	Avg	Std
Talks	1,149	-	-	-	-	-	-
Speakers	961	-	-	-	-	-	-
Users	69,023	-	-	-	-	-	-
Active Users	10,962	-	-	-	-	-	-
Tags	300	5.83	2.11	-	-	-	-
Themes	48	2.88	1.06	-	-	-	-
Related Videos	3,002	2.62	0.74	-	-	-	-
Transcripts	1,102	0.95	0.19	-	-	-	-
Favorites	108,476	94.82	114.54	1.57	8.94	9.90	20.52
Comments	201,934	176.36	383.87	2.92	16.06	4.87	23.42

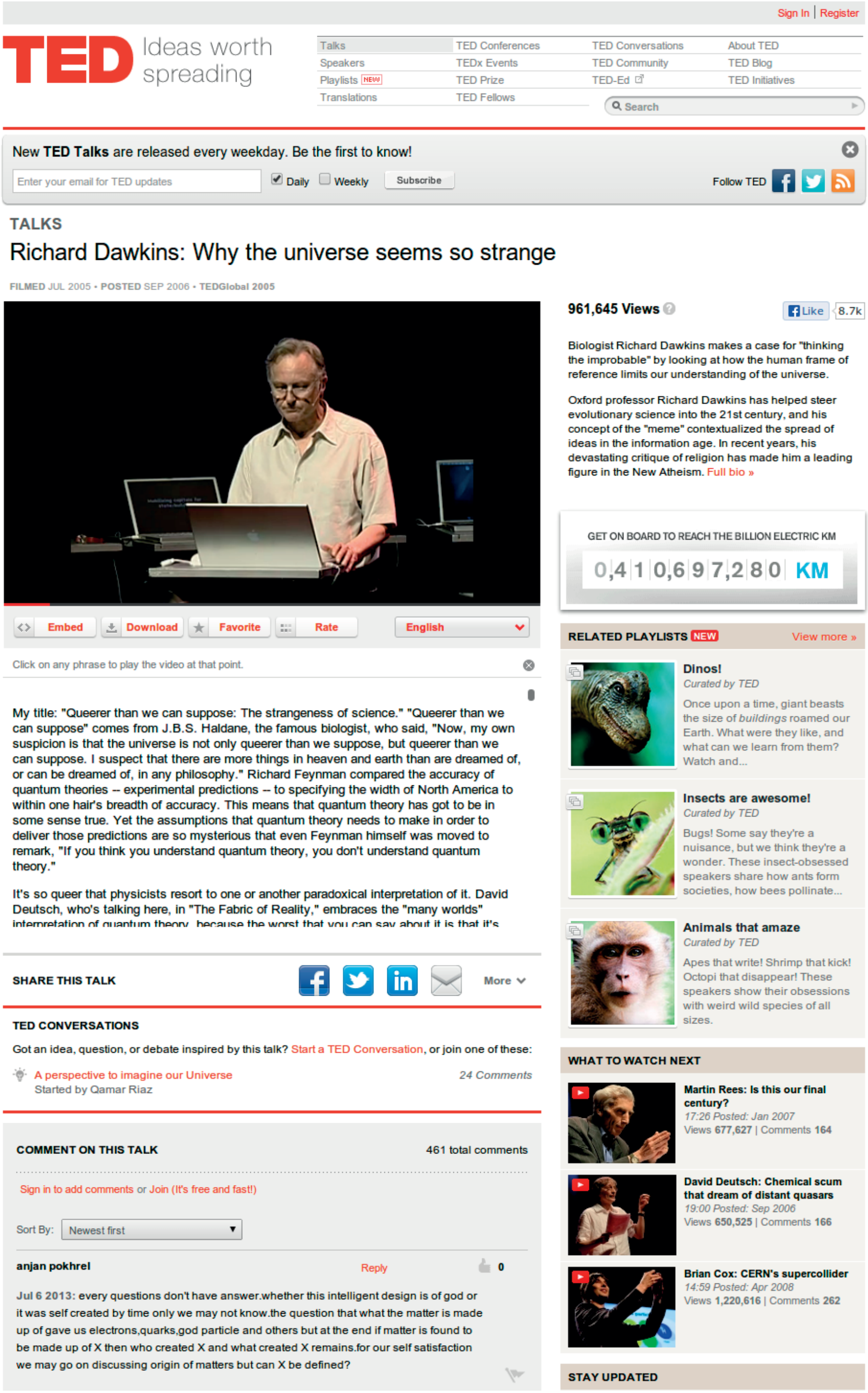
Table 3.1: Statistics for the TED data: total counts and averages (‘avg’) with standard deviations (‘std’) per talk, user and ‘active user’, for each of the attributes. Active users are those who have indicated at least one favorite talk.

right, a short description of the talk is provided, along with the speaker’s bio and the number of total views of the talk. Below the video player is the transcript of the talk, in a separate sub-frame that can be scrolled. To the right of the transcript, the TED website recommends to the user three talks that are related to the one that is currently displayed, which are presented as “what to watch next”. Apart from the talk recommendation, TED website recommends three thematic collections of TED talks i.e. playlists related to the currently displayed talk, presented as “related playlists”. The major part of the area below the player and the transcript is dedicated to the user comments, organized in threads.

3.2.1 Metadata Structure and Statistics

We crawled the TED dataset in April 2012 and gathered the metadata only, excluding the audio-visual recordings. Two main entry types constitute the metadata: talks and users. The talks have the following data fields: identifier, title, description, speaker name, TED event at which they were given, transcript, publication date, filming date, number of views. Each talk has user comments, organized in threads. In addition, we consider three metadata fields that were assigned by the TED editorial staff: tags, themes, and pointers to related talks (generally three per talk). The tags and the themes were additional descriptive keywords for each talk provided by TED during our crawl in 2012, but they are now obsolete, as they were removed in later versions of the TED website. Conversely, the related playlists were not available at crawling time, and thus they are not included in the dataset. For 95% of the talks, a high-quality manual transcript is available. Table 3.1 provides the main statistics of the dataset, which includes 1,149 talks from 961 speakers.

The *users* are all the visitors of the TED website who have created an individual profile and have



New **TED Talks** are released every weekday. Be the first to know!

TALKS

Richard Dawkins: Why the universe seems so strange

FILMED JUL 2005 • POSTED SEP 2006 • TEDGlobal 2005

961,645 Views

Biologist Richard Dawkins makes a case for "thinking the improbable" by looking at how the human frame of reference limits our understanding of the universe.

Oxford professor Richard Dawkins has helped steer evolutionary science into the 21st century, and his concept of the "meme" contextualized the spread of ideas in the information age. In recent years, his devastating critique of religion has made him a leading figure in the New Atheism. Full bio »

GET ON BOARD TO REACH THE BILLION ELECTRIC KM

0,410,697,280 KM

Click on any phrase to play the video at that point.

My title: "Queerer than we can suppose: The strangeness of science." "Queerer than we can suppose" comes from J.B.S. Haldane, the famous biologist, who said, "Now, my own suspicion is that the universe is not only queerer than we suppose, but queerer than we can suppose. I suspect that there are more things in heaven and earth than are dreamed of, or can be dreamed of, in any philosophy." Richard Feynman compared the accuracy of quantum theories -- experimental predictions -- to specifying the width of North America to within one hair's breadth of accuracy. This means that quantum theory has got to be in some sense true. Yet the assumptions that quantum theory needs to make in order to deliver those predictions are so mysterious that even Feynman himself was moved to remark, "If you think you understand quantum theory, you don't understand quantum theory."

It's so queer that physicists resort to one or another paradoxical interpretation of it. David Deutsch, who's talking here, in "The Fabric of Reality," embraces the "many worlds" interpretation of quantum theory, because the worst that you can say about it is that it's

SHARE THIS TALK

TED CONVERSATIONS

Got an idea, question, or debate inspired by this talk? Start a TED Conversation, or join one of these:

A perspective to imagine our Universe — 24 Comments
Started by Qamar Riaz

COMMENT ON THIS TALK — 461 total comments

Sign in to add comments or Join (It's free and fast!)

anjan pokhrel

Jul 6 2013: every questions don't have answer.whether this intelligent design is of god or it was self created by time only we may not know.the question that what the matter is made up of gave us electrons,quarks,god particle and others but at the end if matter is found to be made up of X then who created X and what created X remains.for our self satisfaction we may go on discussing origin of matters but can X be defined?

RELATED PLAYLISTS NEW — View more »

Dinos!
Curated by TED
Once upon a time, giant beasts the size of *buildings* roamed our Earth. What were they like, and what can we learn from them? Watch and...

Insects are awesome!
Curated by TED
Bugs! Some say they're a nuisance, but we think they're a wonder. These insect-obsessed speakers share how ants form societies, how bees pollinate...

Animals that amaze
Curated by TED
Apes that write! Shrimp that kick! Octopi that disappear! These speakers show their obsessions with weird wild species of all sizes.

WHAT TO WATCH NEXT

Martin Rees: Is this our final century?
17:26 Posted: Jan 2007
Views **677,627** | Comments **164**

David Deutsch: Chemical scum that dream of distant quasars
19:00 Posted: Sep 2006
Views **650,525** | Comments **166**

Brian Cox: CERN's supercollider
14:59 Posted: Apr 2008
Views **1,220,616** | Comments **262**

STAY UPDATED

Figure 3.1: Presentation of a lecture on the TED website. The audio visual player (top) is followed by the transcript (in its own sub-frame) and by user comments (not shown here entirely), while on the right side is a short description followed by suggestions of related playlists and talks. (Screen shot from http://www.ted.com/talks/richard_dawkins_on_our_queer_universe.html used here for illustrative purposes only.)

indicated a list of talks as public favorites. Although 69,023 users are registered, only 10,962 of them (i.e. 14%) have explicitly indicated one or more favorite talks, and we will refer to them as *active users*, for reasons related to ground truth and evaluation which will be explained in the next section. Moreover, we will only use the subset of 2,427 users who have made 12 or more ratings each. This value strikes a balance between having enough ratings per user and enough users in the subset, according to standard practice in recommendation system evaluation. All lists of favorites (more than 100,000) and comments (more than 200,000) are included in the metadata set that we distribute and use here for training and testing.

We made available the TED metadata set[1] under the same Creative Commons non-commercial license as the TED talks, and by permission of the TED managers. The metadata, excluding audio and video signals, was acquired using web crawlers developed with the Scrapy toolkit (from http://scrapy.org.), one for the talks and one for the user profiles. The data was anonymized in the process, by replacing the public user IDs with hashes and discarding full names. With a polite rate of one request per second, the crawling lasted a couple of hours on April 27, 2012. The extraction of the attributes from talks and user profiles was done with hand-crafted patterns that exploit HTML attributes and CSS classes using the XPath query language.

3.2.2 Ground Truth

The explicit user preferences in a given dataset constitute the ground truth which can be used for training and evaluating recommendation algorithms for *personalized recommendations*. A common form of such preferences are numeric ratings (e.g. from 1 to 5) that are assigned by users to items. In the TED dataset, the fact that a user has listed a talk among her favorite talks will count as the explicit preference. This corresponds to unary feedback or rating, coded as '1' for a favorite talk, and '?' for a talk not included in the list of favorites. The latter case can mean two things: either the talk was not seen, or it was seen but was not liked. The ambiguity cannot be solved because viewing information for each profile is not available.

Therefore, we are not interested in predicting explicit rating values, but rather in ordering items according to the hypothesized user's preferences as defined by favorite lists (Shani and Gunawardana, 2011). We should note that this evaluation is considerably different from conducting user studies to judge the performance of recommender systems and from modeling detailed user preferences recorded with ontology-based approaches (Dasiopoulou et al., 2010; Martinez, 2002; Tsinaraki and Christodoulakis, 2006). The former, aside from the biases, is time-consuming and challenging. The latter is based on fine-grained semantic modeling of user preferences, but such models are difficult to construct and cannot be compared directly. Instead, modeling user preferences only based on individual properties (e.g. favorites, purchases) is typical of large-scale collaborative filtering systems and is helpful to compare the output of such systems. However, ontologies could be used to extract user and item features (see Section 2.1.2).

[1] https://www.idiap.ch/dataset/ted/.

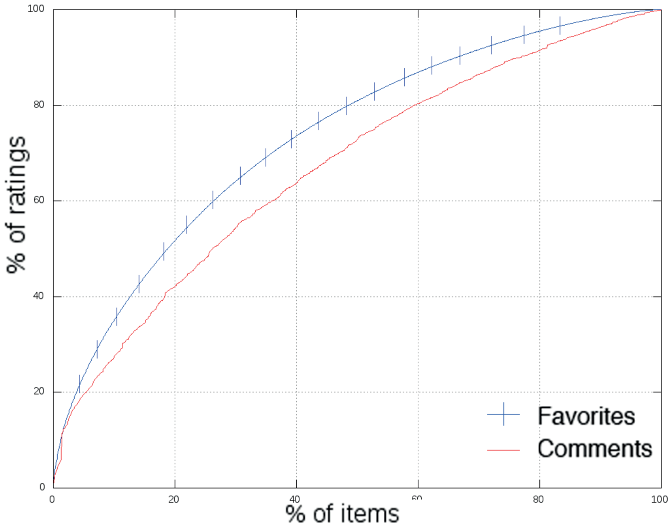


Figure 3.2: Distributions of user feedback (favorites and comments). The percentage of items covered is on the x-axis and the percentage of ratings is on the y-axis.

As the goal of our recommender system is to predict favorite lectures, we will evaluate it, following common practice, by hiding some of the favorite talks of active users and measuring how well the system predicts them simply by comparing the system's output with the ground truth. For this measure, only the profiles of active users can be used, because for the others, no favorites are available. Moreover, personalized recommendation algorithms must be tested on user profiles that contain a sufficient number of ratings to serve as training data for each profile (Herlocker et al., 2004; Shani and Gunawardana, 2011). This is why only active users with at least 12 favorites are kept in our experiments.

Things would be different if we tried to predict the commenting behavior, because this task is distinct from recommendation. In our view, commenting does not always signal positive interest – though it likely signals that the talk has been at least partly viewed – because the meaning of comments is uncertain without further analysis: they may indicate that a talk was liked or disliked, or they may be mere replies to an argument from previous comments. Given that the goal of many recommender systems is to predict purchase, we consider that this is more closely related to marking talks as favorites rather than just commenting on them. Therefore, we did not experiment here with prediction of commenting behavior. Nevertheless, we show in Chapter 4 that the *polarity* of user comments obtained through automatic sentiment analysis can be used to augment rating information.

3.2.3 Distributions of User Feedback

Figure 3.2 displays the distributions of favorites and comments in the TED dataset. The favorite marks are less sparse than comments, since the percentage of the former is higher than the percentage of the latter for the same percentage of items. In Figure 3.3, the TED talks

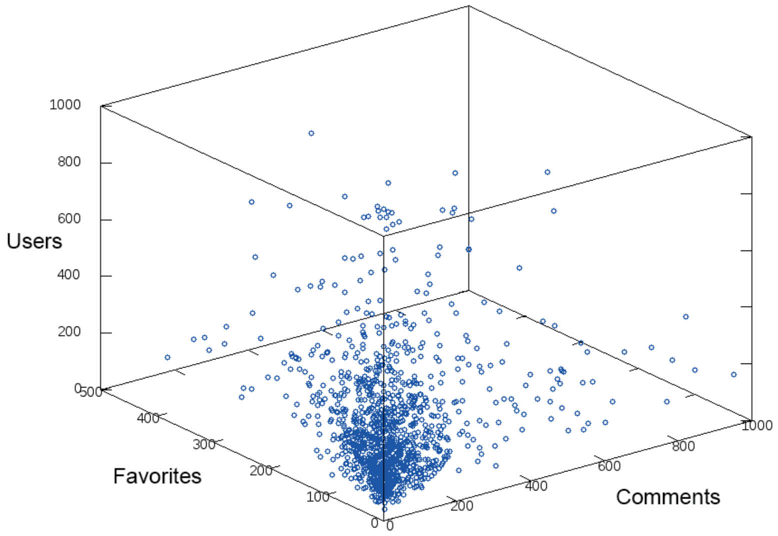


Figure 3.3: Three-dimensional representation of the numbers of favorites and comments, and the unique users that made them for each talk, showing the skewed distribution of user feedback. The number of comments is on the x-axis, the number of favorites is on the y-axis, and the number of unique users that gave feedback is on the z-axis.

are displayed in a three-dimensional space, which shows more clearly the density of favorites and comments. The majority of the talks receive feedback from 1 to 500 unique users, with 1 to 250 favorite marks and 1 to 400 comments (including comments on comments, etc.). As explained above, in this paper, we use favorites as explicit ratings for training and testing, while noting that comments could be used as additional ratings on condition that their polarity is analyzed.

Corroborating the well known long-tail distribution of rated items found in data from many commercial systems, here as well the majority of ratings are condensed over a small fraction of the most popular items (Anderson, 2006). We examined the TED dataset to find out whether this property applied to its distribution of explicit ratings (favorites) as well, and found that 23% of the ratings apply to the top 5% of the items (short-tail) and the rest are distributed over the remaining set of 77% less popular items (long-tail). Hence, the ratings in the TED dataset do follow a long-tail distribution, but it is less long-tailed than other distributions known in the literature: for instance, 33% of ratings apply to the top 5.5% movies in the Movielens dataset, and 33% of the ratings apply to the top 1.7% movies in the Netflix dataset. The fact that the distribution of ratings is less skewed is likely due to the young age of the TED dataset (6 years old) and its rather slow growth.

A marked long-tail distribution may introduce a bias to the recommendation process since an algorithm which recommends only the most popular items may have good performance, but does not always bring benefits to the users because the recommendations may not be novel to them, as shown by Cremonesi et al. (2010). In the TED dataset, this effect should be less observed since the distribution of ratings is less long-tailed.

Collection	**Basic**	**Speaker**	**Transcripts**	**Tags**	**Implicit**	**Explicit**	**CC**
VideoLectures	✓	✓	✓		✓		
KhanAcademy	✓	✓			✓		
Youtube EDU	✓		✓		✓	✓	
DailyMotion	✓				✓	✓	
TED	✓	✓	✓	✓	✓	✓	✓

Table 3.2: Comparison of TED with other repositories in terms of available metadata and user feedback. The properties are: **Basic** (title and description), **Speaker**, **Transcript**, **Tags** (categories in form of keywords), **Implicit** (implicit feedback such as comments or views), **Explicit** (explicit feedback sich as ratings, favorites or bookmarks), and **CC** (Creative Commons Non-Commercial License).

3.2.4 Comparison with other Collections

The aforementioned properties of the TED data cannot be easily found in other alternative lecture repositories such as Khan Academy[2], VideoLectures.NET[3], YouTube EDU[4], or Dailymotion[5] – as shown in Table 3.2, which compares various properties of these data sets. Khan Academy is an online learning community that contains more than 3,200 videos on scholarly topics. It shares some properties with TED in terms of providing transcripts and offering commenting capabilities, but it lacks descriptive fields, annotation with thematic tags and explicit feedback. Similarly, VideoLectures.NET, Youtube EDU or Dailymotion do not provide transcripts and do not provide all the TED metadata fields. The dataset provided for the VideoLectures.NET recommender system challenge (Antulov-Fantulin et al., 2011) includes the viewing history of the lectures as a ground truth for predicting future views of each lecture, along with content-related features, author and event information. However, information that is particularly useful for recommendation tasks such as explicit user feedback and detailed content information such as lecture transcripts is not made available.

The TED dataset thus appears as particularly valuable since it provides ground truth from explicit user preferences along with human-made recommendations, which are critical for evaluating, respectively, personalized and generic recommendation tasks. Besides, the dataset has been used for evaluating other tasks, in particular in the fields of automatic speech recognition (e.g. Rousseau et al., 2012) and machine translation (e.g. Federico et al., 2012; Hardmeier et al., 2015), where the transcripts are distributed as the WIT3 corpus (**?**).

[2] http://www.khanacademy.org/.

[3] http://www.videolectures.net/.

[4] http://www.youtube.com/education/.

[5] http://www.dailymotion.com/.

3.3 Definition of Recommendation Tasks

In this section, we specify two complementary recommendation tasks that can be evaluated using the TED dataset, namely a personalized and a generic one. The first one considers the global history of each user (embodied in the list of favorites) to recommend to the user new content of interest, while the second one aims at recommending content that is related to a given talk, regardless of the user watching it. Of course, a combined task could also be defined, in which a given user watching a given talk receives further recommendations – an instance of context-aware recommendation (Adomavicius and Tuzhilin, 2011). However, the available TED metadata does not offer ground-truth data to evaluate such a task, and thus it is beyond the scope of this study.

3.3.1 Personalized Recommendations

Given a set of unary ratings as a ground truth, the goal of personalized recommendation is to predict whether unseen items will be interesting or relevant for each individual user (Shani and Gunawardana, 2011). In other words, we try to predict the N most interesting items, a problem which is known as top-N recommendation (Cremonesi et al., 2010). The problem of dealing with unary ratings in collaborative filtering is also known as one-class collaborative filtering (OCCF), originally defined by Pan et al. (2008). Such problems are particularly challenging due to the fundamental uncertainty of the '0' class. In such a scenario of offline prediction, the recommendation models are classically trained on fragments of user's histories, and evaluated by hiding some of the preferred user items and then trying to predict them. The performance is evaluated using classification accuracy metrics.[6]

For the TED dataset, we propose that for each user u in the set of users U (or a subset of it, such as users having made more than a number of ratings, similar to the assumption made in Section 3.2.1), her ratings (favorites) are randomly split into training and test sets, noted $\mathscr{M}$ and $\mathscr{T}$, typically 80% and 20%. A recommendation model is trained (possibly with cross-validation) on $\mathscr{M}$, and then tested on the held-out set $\mathscr{T}$ by comparing its output with the actual ratings of user u over $\mathscr{T}$.

3.3.2 Generic Recommendations

The generic or user-independent recommendation task corresponds to scenarios in which the users' history of ratings is absent, e.g. for anonymous users. The goal of this task is to predict the most similar items to a given one, which can also be seen as a non-personalized top-N

[6] The scenario of this task does not presuppose that the user is currently viewing a talk, but considers only the user's past history. As a consequence, if a user is interested in several different topics, it is likely that in the resulting recommendations each topic will be present with its probability of appearance in the user's past history. On the contrary, if one aims for contextual personalized recommendation (mentioned above), then the topic of the talk that is currently viewed should be considerably boosted with respect to the others in the resulting recommendations.

recommendation task. Given the set of human-made, user-independent recommendations for each item in a dataset – here, the three related videos (or “what to see next”) selected by TED editors for each TED talk – a model can be trained and evaluated using only this information as ground truth and ignoring user preferences or the talks previously viewed. Again, the set of items I can be split into a training set $\mathscr{M}$ and a testing set $\mathscr{T}$ for evaluation.

3.3.3 Evaluation Metrics

For the top-N personalized recommendation task, error metrics such as RMSE are not the most appropriate ones, since a top-N recommender is not necessarily able to infer the exact rating of a user $u \in U$ for any item $i \in I$ (Cremonesi et al., 2010). Instead, this task can be evaluated more informatively by using the classification accuracy metrics of precision, recall and F-measure (Shani and Gunawardana, 2011). Precision and recall at N are respectively given by the following formulas:

$$P(N) = \frac{1}{|U|}\sum_{u\in U}\frac{|\mathscr{T}_u \cap \mathscr{R}_{u@N}|}{N};\; R(N) = \frac{1}{|U|}\sum_{u\in U}\frac{|\mathscr{T}_u \cap \mathscr{R}_{u@N}|}{|\mathscr{T}_u|}, \tag{3.1}$$

where N is the bound of top recommendations, $|U|$ is the total number of users in U, $\mathscr{T}_u$ is the set of items in user’s u history and $\mathscr{R}_{u@N}$ are the top-N recommendations of the model for the user u. Recall is computed in a similar way, but dividing by the number of items in the user’s u history, $|\mathscr{T}_u|$, instead of N. The F-measure is the harmonic mean $F(N)$ of $P(N)$ and $R(N)$, which is computed as:

$$F(N) = 2\cdot\frac{P(N)\cdot R(N)}{P(N)+R(N)} \tag{3.2}$$

Similarly, applying Equation 3.1 directly to the items I in a test set $\mathscr{T}$, we obtain the definitions of precision and recall for generic recommendations as follows:

$$P(N) = \frac{1}{|I|}\sum_{i\in I}\frac{|\mathscr{T}_i \cap \mathscr{R}_{i@N}|}{N};\; R(N) = \frac{1}{|I|}\sum_{i\in I}\frac{|\mathscr{T}_i \cap \mathscr{R}_{i@N}|}{|\mathscr{T}_i|}, \tag{3.3}$$

where $\mathscr{T}_i$ are the ground-truth items related to i, $\mathscr{R}_i$ are the recommended items for i and the remaining variables are the same as above.

3.4 Semantic Vector Space Models

Content-based recommender systems use similarity measures between items that rely on their content descriptors. Here, we consider semantic vector space models (VSM) to define such similarities, and compare in Sections 3.7 and 3.8 their merits for recommendation over the TED dataset. Benchmarking these models is a contribution to the ongoing debates on semantic-based approaches to recommendation (Lops et al., 2011).

Semantic VSMs are considered to be able to reduce the effect of the curse of dimensionality and data sparseness of standard VSMs, such as those based on TF-IDF weighting (Sahlgren, 2006). The proximity of two vectors in a semantic space, which is usually computed with cosine similarity, can be interpreted as a measure of the semantic relatedness between the objects that are represented by those vectors. This semantic similarity can then be used to compute similarities between items or users in recommender systems.

When using a VSM, each document i is first represented as a feature vector composed of terms $(t_{i1}, t_{i2}, \ldots, t_{ij})$, where each position j corresponds to a word of the vocabulary V. The coefficients t_{ij} can be computed using various models: Boolean values ('1' if the document contains the word, '0' if it does not), counts of words, term frequencies, inverse document frequencies, or TF-IDF coefficients. [7] The TED talks, noted as items I, can thus be represented by creating vectors of words from their metadata, which can be pre-processed to remove stop words or to apply stemming. In this study we performed the following pre-processing steps:

I → Tokenization → Stop words removal → Stemming → V

There are several methods for creating semantic representations in VSMs. In our experiments, we use a VSM with TF-IDF as the baseline weighing model (Salton and Buckley, 1988) and evaluate four representative semantic VSMs from the three main existing categories, as follows:

1. Two dimensionality reduction methods, namely Latent Semantic Indexing (LSI) and Random Projections (RP), respectively proposed by Hofmann (1999) and Sahlgren (2005).

2. A topic modeling method, namely Latent Dirichlet Allocation (LDA) by Blei et al. (2003).

3. A concept space based on external knowledge from Wikipedia, namely Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2007).

These techniques have generalization capabilities, as they project the data from the original vector space to a topic or concept space with a reduced number of dimensions – apart from ESA which actually augments the dimensionality to the number of Wikipedia concepts. In terms of free parameters, LSI, RP and LDA rely on the number of topics t (latent factors). Moreover, LDA relies on two parameters traditionally noted α and β for the Dirichlet priors of topic and word distributions. For the implementation of LSI, RP and LDA we used the Python Gensim library (Řehůřek and Sojka, 2010), while for ESA we used the Wikipre-ESA implementation of the method described in (Gabrilovich and Markovitch, 2007), over a 2005 snapshot of Wikipedia.

[7] For example, TF-IDF is computed as follows: $t_{ij} = tf_{ij} \cdot idf_j$, where tf_j is the term frequency of word j in document i and idf_j is classically the inverse document frequency of word j.

3.5 Recommendation Algorithms

In this section, we define two types of recommendation algorithms for personalized lecture recommendation, namely content-based and collaborative filtering ones, which are based on item-based similarities (Papagelis and Plexousakis, 2005). Then, we show how to combine minimal collaborative information, i.e. item popularity, with content-based item similarities.

3.5.1 Content-based Algorithms

For content-based methods, we first pre-compute an item similarity matrix for each of the VSMs above, noted respectively S_{TF-IDF}, S_{LSI}, S_{RP}, S_{LDA} and S_{ESA}. Each matrix S is an $m \times m$ matrix, m being the number of talks. The value of each element s_{ij} of each S is the cosine similarity of the vectors representing items i and j in the given VSM.

We then define a ranker based on content similarities, noted as CB. Given a similarity function that outputs a score for two items (two TED talks), CB recommends to a user u a list of ranked items based on the k most similar items to those already known to be her favorites from the training data $\mathscr{M}_u$. Therefore, CB recommends items to user u based on their estimated relevance $\hat{r}_{ui}$ defined as:

$$\hat{r}_{ui} = \sum_{j \in D^k(u;i)} s_{ij} \tag{3.4}$$

where $D^k(u;i)$ are the k most similar items from I to the ones in the training set of the user $\mathscr{M}_u$ and s_{ij} is the similarity between items i and j, both of them computed according to one of the five matrices S. The summation is limited to a set of k neighbors ($D^k(u;i)$) for tractability and efficiency reasons.

3.5.2 Collaborative Filtering Algorithms

For collaborative filtering, we use the model from (Cremonesi et al., 2010) applied to unary ratings. First, we pre-compute the item similarity matrices based on the ratings between pairs of items in the user-item matrix built from the training set. The similarity of each pair is based on two popular metrics, namely Pearson correlation, yielding the S_{PC} matrix as in (Mahmood and Ricci, 2009), and cosine similarity, yielding the S_{COS} matrix, as in (Cremonesi et al., 2010). The elements of these matrices are computed respectively as follows:

$$S_{COS_{ij}} = \frac{\vec{i} \cdot \vec{j}}{||\vec{i}||_2 \times ||\vec{j}||_2}; \; S_{PC_{ij}} = \frac{E[(\vec{i} - \mu_i)(\vec{j} - \mu_j)]}{\sigma_i \sigma_j}, \tag{3.5}$$

where $\vec{i}$ and $\vec{j}$ are the feature vectors of items i and respectively j derived from the item-item co-rating matrix[8].

Then, we use a neighborhood model defined in Equation 3.6, which is commonly used for collaborative filtering. The prediction function $\hat{r}_{ui}$ estimates the rating of a user u for an unseen item i, based on the bias estimate b_{ui} of u for item i, computed using Equation 3.8, and on a score that is calculated from the k most similar items to i (according to either S_{PC} or S_{COS}) which the user u has already rated, i.e. the neighborhood $D^k(u;i)$ as above. The denominator ensures that the predicted ratings will fall in the same range of values as the known ones.

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in D^k(u;i)} d_{ij}(r_{uj} - b_{uj})}{\sum_{j \in D^k(u;i)} d_{ij}} \tag{3.6}$$

The term r_{uj} is the rating value of a user u for a given item j. The coefficient d_{ij} expresses the similarity between item i and item j, and is computed as in Equation 3.7 by using the similarity s_{ij} between items i and j multiplied by a factor varying from 1 (when the number of common raters n_{ij} is considerably larger than λ) to 0 (when n_{ij} is considerably smaller than λ). Typically, $\lambda \approx 100$.

$$d_{ij} = s_{ij} \frac{n_{ij}}{n_{ij} + \lambda} \tag{3.7}$$

The bias estimate b_{ui} is the sum of the average ratings μ of items in the dataset, the average of the ratings of a user u, noted b_u, and the average of the ratings for a given item i, noted b_i, as shown in Equation 3.8:

$$b_{ui} = \mu + b_u + b_i \tag{3.8}$$

We consider two representative variants of this model. First, we use a normalized neighborhood model (as defined in Equation 3.6) with Pearson Correlation for vector similarity; this model is noted as *CF(PC)*. Second, we use a non-normalized model, noted with a preceding 'u' for 'unnormalized', obtained by removing the denominator in Equation 3.6 and using the cosine similarity distance, hence this model is referred to as *uCF(COS)*. In previous studies (Cremonesi et al., 2010), non-normalized models were found to perform better for the top-N recommendation task than normalized ones.

3.5.3 Combining Collaborative Filtering with Content Similarity

We incorporate into the neighborhood model presented above (non-normalized version) information about content-based similarity, by replacing in Equation 3.6 the d_{ij} similarity with the content-based one from Equation 3.4, and using the non-normalized version. Hence

[8] In other formulations, the feature vectors can be derived from the user-item matrix, where each item is represented by a vector of user ratings.

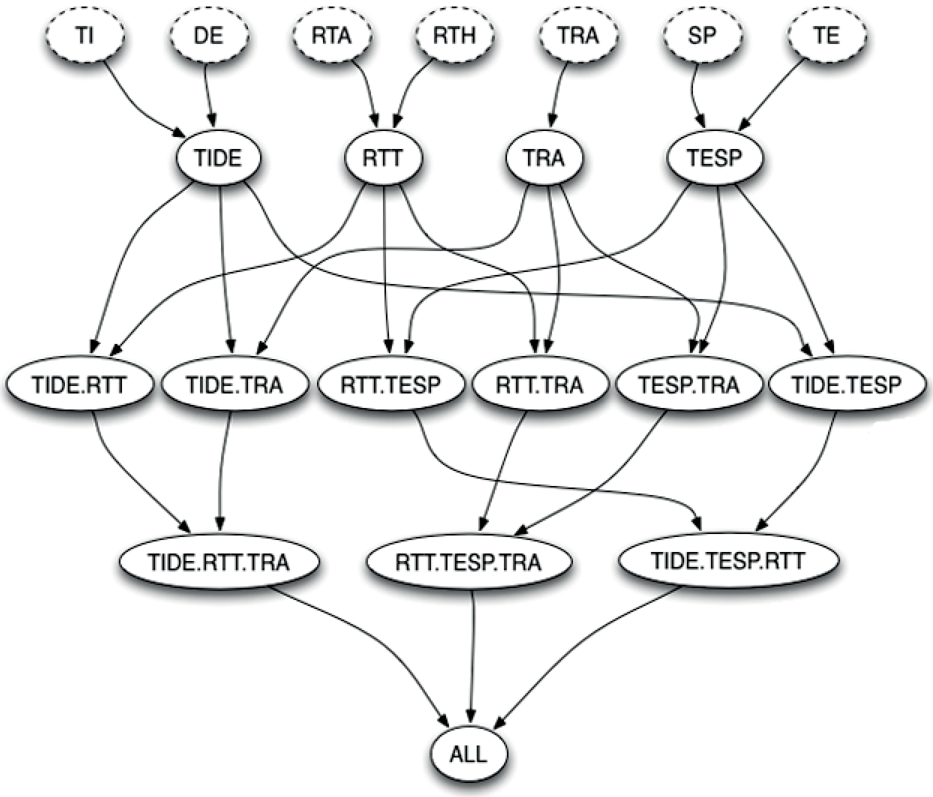


Figure 3.4: Combinations of item attributes from which features are extracted for comparison. The atomic attributes are title (TI), description (DE), related tags (RTA), related themes (RTH), transcript (TRA), speaker (SP) and TED event (TE).

the estimated rating in the combined model is:

$$\hat{r}_{ui} = b_{ui} + \sum_{j \in D^k(u;i)} s_{ij}(r_{uj} - b_{uj}) \tag{3.9}$$

This new model allows us to exploit at the same time the semantic-based similarities and the bias estimate, therefore to combine the two types of information, content-based and collaborative one. This is especially useful when collaborative information is sparse, and the similarity computed using it is less reliable than the content-based one.

We consider this time only the non-normalized versions of the model (noted again with 'u') because they outperformed the normalized ones in our preliminary experiments. We also indicate in the notation the type of content-based similarity that is used in combination to the CF neighborhood model. Hence, these combined models are referred to as *uCF(TF-IDF)*, *uCF(LSI)*, *uCF(LDA)*, *uCF(RP)* and *uCF(ESA)*. For comparison purposes, we finally consider a user-independent recommender noted *TopPopular*, which always recommends the items with the highest popularity, based on the total number of ratings, regardless of a user's preferences.

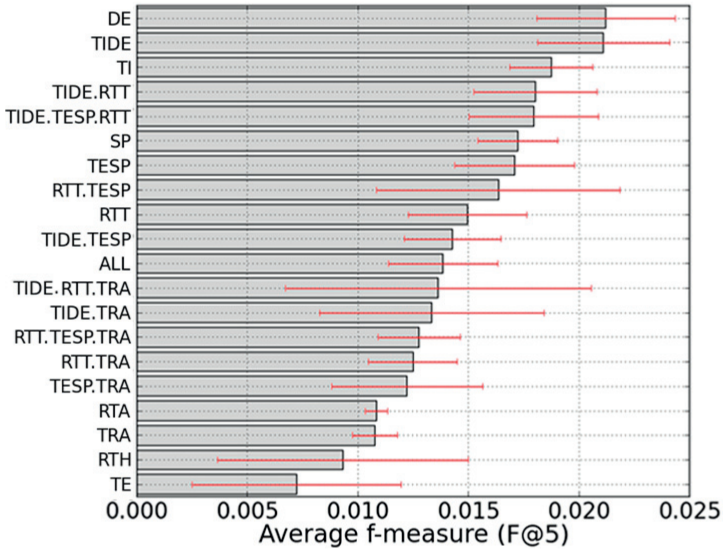


Figure 3.5: Ranking of individual and combined attributes based on the decreasing average of F-measure over all five content-based methods. Atomic attributes are title (TI), description (DE), related tags (RTA), related themes (RTH), transcript (TRA), speaker (SP) and TED event (TE). The segments over the bars represent the standard deviations when averaging over 5-fold cross-validation and five methods with all tested parameters.

3.6 Parameter and Attribute Selection for Content-Based Methods

We determine the optimal parameters and attributesas of the content-based methods for personalized recommendation using 5-fold cross-validation over the training set $\mathscr{M}$, which includes 80% of the ratings for each of the 2,427 TED users that have made 12 or more ratings. The remaining 20% of the ratings from these users are kept as an unseen test set $\mathscr{T}$, which is used in Section 3.7.

The CB methods use lexical features (vectors of words) extracted from one or more fields/attributes of each TED talk, represented schematically in Figure 3.4, and several meta-parameters for each of the semantic representations (TF-IDF, LSI, RP, LDA, and ESA) as described in Section 3.4. Exploring all possible combinations of attributes to find out which subset performs best is not tractable. Therefore, we grouped individual attributes into four groups: title and description (TIDE), related tags and themes (RTT), transcript (TRA), and speaker plus TED event (TESP). Along with all individual attributes, we tested these sets, and all their combinations, organized as in Figure 3.4.

For LSI and RP we optimized the values of the parameter t (number of topics) by varying it from 10 to 5,000 and for LDA from 10 to 200 only, for tractability reasons. Additionally, for LDA, we varied the α and β parameters from 0 to 1, and the optimal ones were found to be $\alpha = 1$ and $\beta = 0.002$. We fixed the neighborhood size at $k = 3$, which is a trade-off between computational cost and expected prediction accuracy (Koren and Bell, 2011).

Figure 3.5 displays the ranking of attributes and their combinations, ordered by the average F-measure (F@5) over *all* the tested types of semantic VSMs (i.e. TF-IDF, LSI, RP, LDA, and ESA) and all the parameters of methods stated in the previous paragraph. These results thus indicate which attributes perform well over *all* methods, as opposed to attributes that are optimal for *each* method, which will be shown below. As seen on the standard deviations obtained from cross-validation and averaging over the five methods (segments over the bars in Fig. 3.5), the non-overlapping segments indicate important differences between single or composite attributes. For instance, the four top-level attributes are clearly better than the four bottom ones.

The results show that the human-made description of talks (DE), the title (TI), and their combinations with other attributes (TIDE, TIDE.RTT, and TIDE.TESP.RTT) are the most useful attributes on average for content-based personalized recommendation. In addition, knowledge of the speaker (SP) is useful too (ranked sixth). However, these metadata fields come to a cost because they must be entered by the editors of the lecture repository. The description, in particular, requires a significant human effort, hence likely the TED speakers write their own descriptions.

The lowest performing attributes were the name of the TED event (TE) and the related themes assigned by TED editors (RTH), which presumably lack specificity for recommendation. In fact, the related tags and themes have been removed since we gathered the metadata in 2012 removed from the TED website. Somewhat surprisingly, the transcript (TRA) decreases the performance of all methods and most of the combinations that include it are in the middle of the ranking. One possible explanation is that the huge size of the transcript's vocabulary introduces a lot of noise.

Table 3.3 shows the optimal attributes and parameters for each semantic representation used with CB, together with the scores (precision, recall and F-measure at 5) that they enable the recommender system to reach (5-fold cross-validation on the development data). All the semantic-based methods except LDA outperform significantly the TF-IDF baseline (pairwise t-statistic, $p < 0.05$): 11% improvement for LSI, 7.6% for RP and up to 64% by ESA, which reaches the best score. While two semantic-based methods (LSI and RP) perform without significant differences, ESA is significantly above them (pairwise t-tests, $p < 0.05$). The performance of ESA shows that the external-knowledge-based representation of the items is significantly more useful to our task than the domain knowledge captured intrinsically by the other methods.

3.7 Performance of Personalized Lecture Recommendation

In this section, we compare recommendation performance of CB, CF and combined methods over the held-out test set $\mathscr{T}$, considering two different settings: (i) a cold-start setting where the collaborative rating information for the items is not available and (ii) a non-cold-start setting where it is. Note that when testing, we only hide the rating information for the user currently tested, but use the information from the other users to make our recommendations, following current practice in the field.

Method	Optimal Attributes	Performance (%)		
		P@5	R@5	F@5
LDA (*t*=200)	Title, description, TED event, speaker (TIDE.TESP)	1.63	1.96	1.78
TF-IDF	Title (TI)	1.70	2.00	1.83
RP (*t*=5000)	Description (DE)	**1.83**	**2.25**	**2.01**
LSI (*t*=3000)	Title (TI)	**1.86**	**2.27**	**2.04**
ESA	Title, description (TIDE)	**2.79**	**3.46**	**3.08**

Table 3.3: Optimal attributes for content-based methods found using 5-fold cross-validation on the training set. Scores in bold are significantly higher than TF-IDF ones. Moreover, the scores of ESA are significantly above those of RP and LSI (pairwise t-statistic, $p < 0.05$).

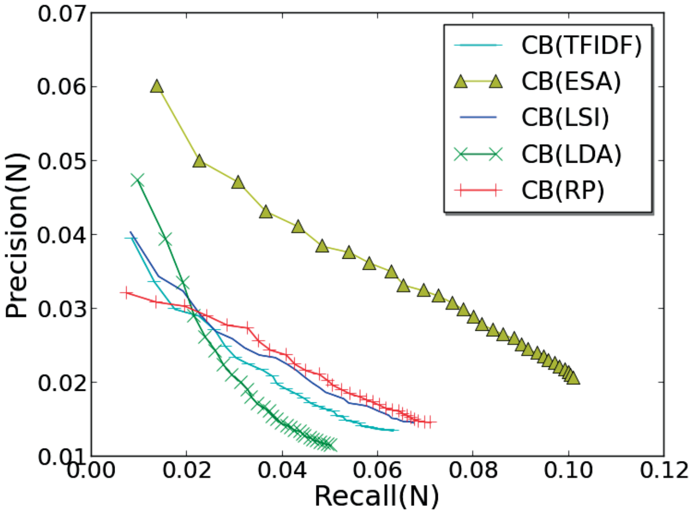


Figure 3.6: Scores of content-based methods in a cold-start setting, in terms of precision and recall at N ($1 \leq N \leq 30$) on the held-out set $\mathcal{T}$. The ESA-based distance outperforms by far all the others, while LSI and RP outperform TF-IDF for most values of N, and LDA outperforms TF-IDF for low values of N (1 to 4).

3.7.1 Cold-start Recommendations (CB Methods Only)

The cold-start setting is characterized by sparse user ratings, with many items not having been rated at all, which makes it impossible for CF methods to recommend these items (e.g. new TED lectures). In such a situation, only content-based methods can help making recommendations. In Figure 3.6, we show the performance of our CB methods in terms of precision and recall over the held-out set $\mathcal{T}$. Most of the semantic-based representations perform significantly better (t-statistic, $p < 0.05$) than TF-IDF, with +62% for ESA, +7% for LSI and +8% for RP. As for LDA, this does not improve over TF-IDF (as also seen in Table 3.3) except at the top 1 to 4 recommendations; it was also the most difficult method to tune.

The scores obtained appear to be overall quite small, though in line with previous work – see (Cremonesi et al., 2010; Pan et al., 2008) and Section 2.1.4. These scores must be interpreted in the light of the following two facts. Firstly, the probability of having the correct item ranked by chance first (P@1) among 1,149 candidates is only 0.08%, while our *lowest* score (for Random Projections) was 40 times higher at 3.20% (Figure 3.6). Moreover, the precision of random guessing decreases dramatically at higher ranks (e.g. P@5). Secondly, we consider here only the positive ratings (favorites) to calculate precision, and discard the scores of unseen items, which would have a much higher baseline.

The improvement brought by ESA appears to be again (as in Section 3.6) much greater than that of LSI and RP, allowing us to conclude that similarity based on concept spaces from external knowledge captures more effectively the content similarity and, consequently, the user preferences than the other semantic spaces and the baseline TF-IDF. Semantic-based approaches are thus more effective than keyword-based ones for cold-start personalized recommendations.

3.7.2 Non-Cold-Start Recommendations (All Methods)

In a non-cold-start setting, where the items have been rated by many users, the collaborative filtering information and the bias introduced by the popularity of items can be specifically exploited. As the CB methods do not have such information, their performance was found to be lower than that of CF methods, and will not be reported here. However, the combinations of CB and CF proposed in Section 3.5.3 (noted *uCF(·)* with '·' indicating the similarity method) allow content-based similarity to take into account the bias estimate, and their results are close to pure CF methods in the non-cold-start scenario, while being operational both in cold-start and non-cold-start settings.

Figure 3.7 displays the performance of two neighborhood models used for collaborative filtering: the normalized one using Pearson Correlation (*CF(PC)*) and the unnormalized one using cosine similarity (*uCF(COS)*). We also represent the two best performing combined methods, unnormalized, using TF-IDF and LSI distances (*uCF(TF-IDF)* and *uCF(LSI)*), as well as the *TopPopular* baseline. The best performance is achieved by the non-normalized

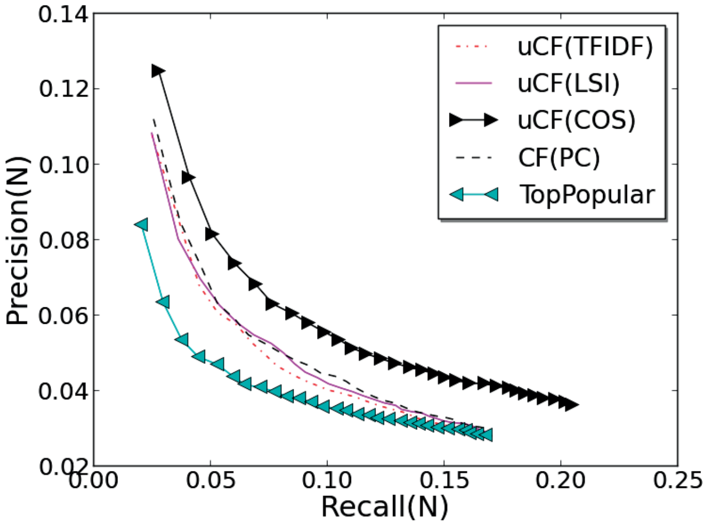


Figure 3.7: Lecture recommendation scores for two collaborative filtering methods, *CF(PC)* and *uCF(COS)*, and two combined methods, namely *uCF(TF-IDF)* and *uCF(LSI)* (neighborhood with TF-IDF and LSI distances), in a non-cold-start setting. Precision and recall at $1 \leq N \leq 30$ are computed on the held-out test set $\mathscr{T}$. Collaborative filtering using cosine similarity in a neighborhood model scores highest, but the combined model using neighborhoods and TF-IDF is not far behind.

neighborhood model with cosine similarity, *uCF(COS)* (+34% on average with respect to *TopPopular* over all data points in Figure 3.7). The *CF(PC)* model is in the same range, and significantly better than *TopPopular* (+15%). The CB methods have insignificant differences with each other and with *uCF(PC)*. All these comparisons are based on pairwise t-tests over the values of the P-R curves from 1 to 30.

The combined models, *uCF(TF-IDF)* and *uCF(LSI)*, perform similarly to *CF(PC)* and are also significantly better (t-statistic, $p < 0.05$) than *TopPopular*, respectively +10.5% and +13% above it. The other content-based similarities (RP, LDA, ESA) perform slightly below TF-IDF, but the difference is not statistically significant. Using the bias introduced by the item popularity thus decreases the difference in performance between the content-based similarity models, i.e. *uCF(LSI)* and *uCF(TF-IDF)*, compared to their differences in the cold-start setting.

3.8 Performance of Generic Recommendations

The goal of generic or user-independent recommendation is to predict items that are related to a given one, without any knowledge of user profiles. We use here unsupervised methods, namely rankers based on content similarities, defined in Section 3.5. As a ground-truth, we use the human-made lists of related videos that are available in the TED data set. In most of the cases (76.4% of the talks), TED editors have indicated three related talks for each talk, or sometimes fewer than three talks (for 23.6% of the talks).

Methods	TED	TopPopular	TF-IDF	LSI	RP	LDA	ESA
TED	1.000	0.006	0.129	0.156	0.143	0.091	0.124
TopPopular	-	1.000	0.003	0.003	0.004	0.004	0.006
TF-IDF	-	-	1.000	0.510	0.323	0.195	0.523
LSI	-	-	-	1.000	0.419	0.220	0.442
RP	-	-	-	-	1.000	0.200	0.299
LDA	-	-	-	-	-	1.000	0.193
ESA	-	-	-	-	-	-	1.000

Table 3.4: Evaluation of unsupervised methods (content-based rankers) for generic recommendation, in terms of overlap with the related talks recommended by TED editors on first line. In the remaining lines, the matrix provides the overlap values between all pairs of methods, showing for instance that ESA and LSI provide the most similar recommendations to TF-IDF (0.523 and 0.510). The metric is the F-measure computed over pairs of recommendation lists produced by the respective methods, and underlined scores are significantly higher than TF-IDF ones (pairwise t-statistic: $p < 0.05$).

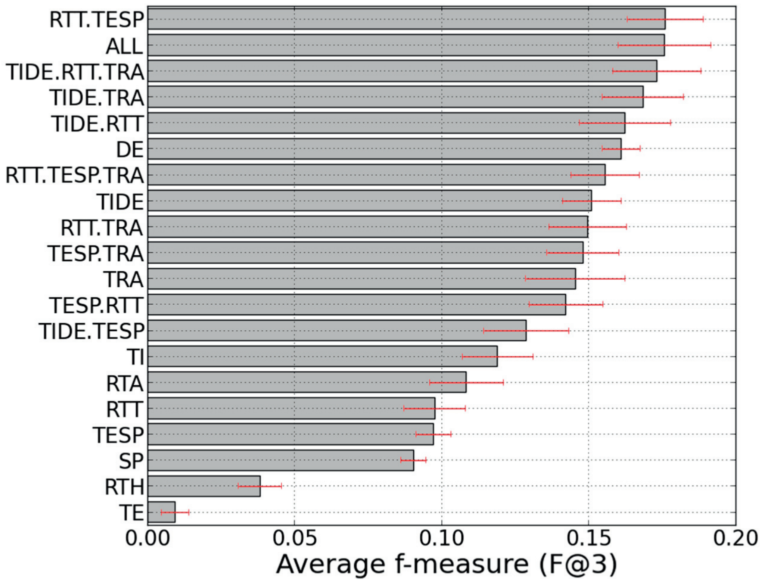


Figure 3.8: Ranking of atomic and combined features (see combinations in Figure 3.5) based on the decreasing average F-measure for TF-IDF similarities. The atomic features are: title (TI), description (DE), related tags (RTA), related themes (RTH), transcript (TRA), speaker (SP) and TED event (TE). The segments over the bars are the standard deviations from 5-fold cross-validation, with non-overlapping intervals indicating important differences.

Using classification accuracy metrics (F-measure), we evaluate various content-based rankers, namely semantic-based and keyword-based ones, in terms of their overlap with the ground-truth ranking. Table 3.4 shows that, similarly to personalized recommendations, the LSI and RP semantic-based methods significantly outperform the keyword-based one using TF-IDF and the other methods as well (pairwise t-test on 5-fold c-v., $p < 0.05$). However, the difference between LSI and RP is not significant. The parameters of the methods were set to the optimal values found for the personalized recommendation task in Section 7, which means the results that are obtained from these rankers might be even improved if we optimize them for the generic task. Results might also improve when supervised methods (rather than unsupervised ones) are used for learning to rank, such as SVM-Rank (Joachims, 2006). The main conclusion at this stage is that the semantic information is beneficial over keyword-based only methods for generic recommendation, as it was for personalized recommendation.

Figure 3.8 displays the ranking of features and their combinations, ordered by the average F-measure (F@3) obtained when using them in the TF-IDF content-based ranker. The ranking of the features for this task is quite different from the one for personalized recommendations (displayed in Figure 3.5 above). For generic recommendations, the combination of all features appears to be the second best performing subset of features, while the subset that actually performs best is RTT.TESP, which includes the related tags and themes, the speaker and the TED event. These sets were ranked in the middle for the personalized recommendation task. When considered independently, the related themes (RTH) and the TED event (TE) fields rank very low (respectively 19th and 20th), while the other two features, namely the related tags (RTA) and the speaker (SP) have have relatively low rank as well (respectively 15th and 17th).

We presume that when put together, these features capture complementary properties, because their combination leads to the best recommendation performance. Note that the combination of some other fields does not lead to improvement, implying that they capture overlapping properties, for example description (DE) compared to title plus description (TIDE). A possible explanation for these differences is that individual user preferences in the personalized task are more difficult to capture than the preferences of the TED editors who defined the related talks used as ground-truth for generic recommendations.

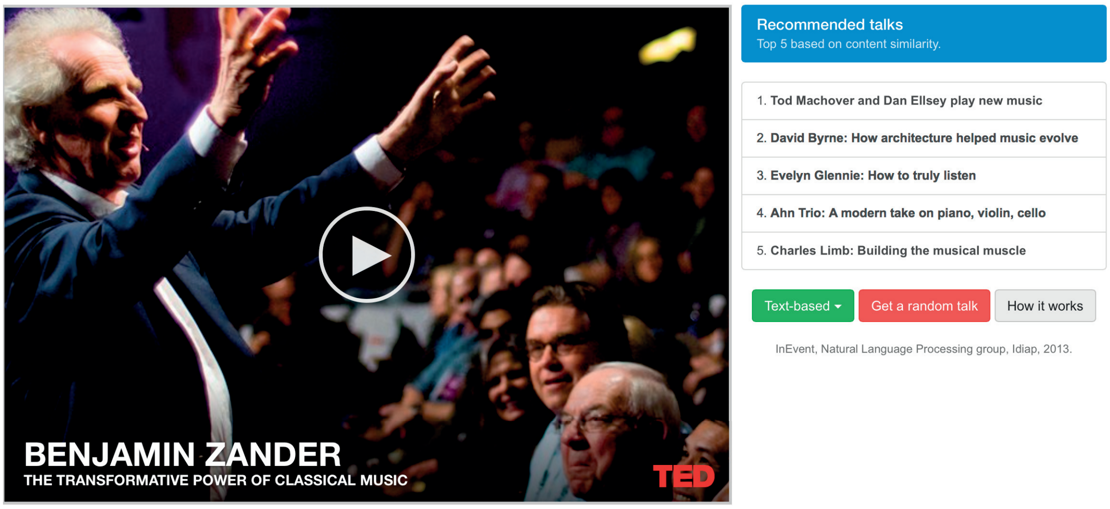
3.9 Applications

To demonstrate the generality of the proposed content representations, we apply them to several scenarios of lecture and snippet recommendation. Such recommendations require the linking of lectures according to their main content or based on their segments. The links can be found through a similarity function in the content representation space, as described above. For this purpose, we implemented and made available a content-based recommendation library[9], which is used in the two applications presented below.

[9] http://github.com/idiap/cbrec/.

3.9.1 The InEvent Portal: Content-Based Lecture Recommendation

The InEvent Portal[10] is purposed for accessing and navigating large repositories of networked multimedia. One of its basic functions is to provide generic (user-independent) recommendations over its repository which contains talks from Klewel[11] plus metadata imported from TED. As discussed above, in such a generic recommendation task, the goal is to predict lectures that are related to a given lecture that is being viewed, without any knowledge of a user's profile. For each of the talks in the repository we recommend the five most similar talks based on text content from all metadata fields. Figure 3.9 shows a snapshot from our online demonstration[12] which displays on its right side the most related talks to the one that is being viewed.



Recommended talks
Top 5 based on content similarity.

1. Tod Machover and Dan Ellsey play new music
2. David Byrne: How architecture helped music evolve
3. Evelyn Glennie: How to truly listen
4. Ahn Trio: A modern take on piano, violin, cello
5. Charles Limb: Building the musical muscle

InEvent, Natural Language Processing group, Idiap, 2013.

Figure 3.9: Generic recommendations computed using semantic similarities of items over the InEvent repository. Example of top five content-based recommendations for a talk that is currently being viewed, presented as a list of clickable titles on the right side.

The proposed solution uses the semantic vector space techniques described in Section 3.4 to represent lectures based on their content. Specifically, we make use of all available item meta-data such as titles, speaker names, descriptions, subtitles, and slide titles with start time, in addition to words extracted through speech recognition, or manual transcripts when they are available. Initially, a vector is built for each lecture (each position corresponding to a word of the vocabulary), with weights computed using TF-IDF coefficients. Then the feature vectors are projected to a low-dimensional semantic space with LSI and finally, a talk similarity matrix is computed with cosine similarity as in Eq. 3.5.

[10] http://www.inevent-project.eu/demos/inevent-portal/.
[11] http://portal.klewel.com/watch/webcasts/.
[12] http://www.inevent-project.eu/demos/automatic-recommendation-of-lectures/.

3.9.2 The MUST-VIS System: Content-Based Snippet Recommendation

The MUST-VIS system allows users to visualize a lecture as a series of segments which are represented by keyword clouds, with relations to other similar lectures and segments. This allows users to understand the main content of a lecture at a glance without entirely watching it, but also to navigate conveniently through the lecture graph generated by the system i.e the navigation graph. The system was designed by the Idiap NLP group (Bhatt et al., 2013b) for the MediaMixer/VideoLectures.NET Temporal Segmentation and Annotation Grand Challenge at ACM Multimedia 2013, and was declared the winner of this challenge. The MUST-VIS system makes use of multimodal processing of audios, videos and texts to build the navigation graph. Initially, the ASR transcripts or the subtitles of each lecture of the dataset are segmented using the TextTiling algorithm implemented in the NLTK toolkit (Loper and Bird, 2002). Then the words in each topic segment are ranked using the diverse keyword extraction technique proposed by Habibi and Popescu-Belis (2013, 2015), which selects keywords so that they cover the maximum number of topics mentioned in each segment with an additional diversity constraint. Words ranked higher are graphically emphasized in the word cloud. Our content-based methods are used then to compute the similarity between the lecture segments which defines the links between segments and between lectures in an interactive navigation graph (color-coded in the GUI). Figure 3.10 shows a snapshot of the graph for a lecture that is currently being viewed, and around it the most similar lectures. The user can quickly inspect the links between the segments and the keyword clouds to decide which lecture to watch next.

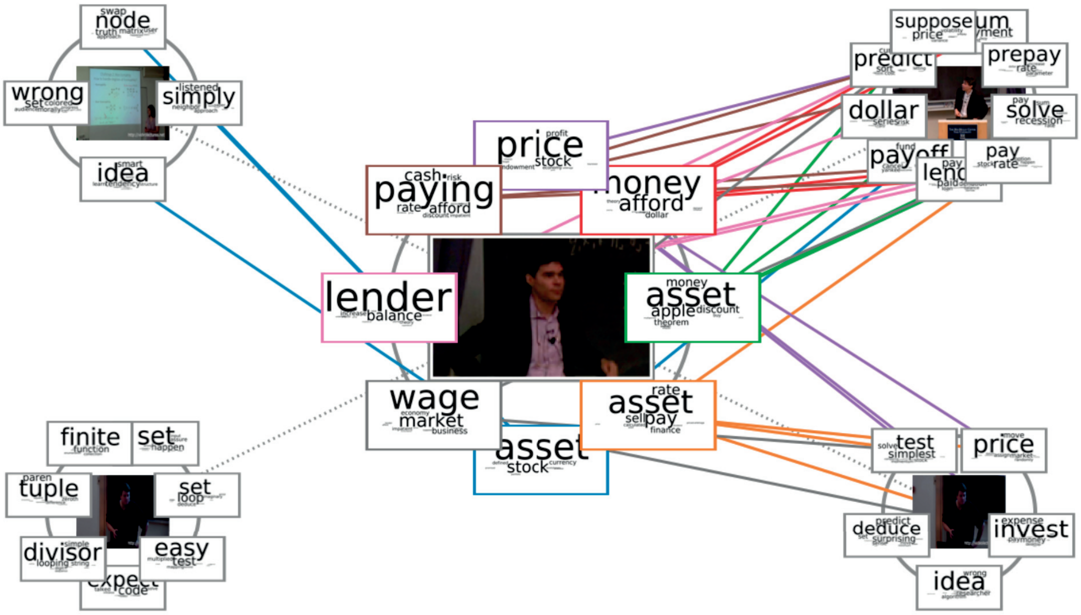


Figure 3.10: Navigation graph of the MUST-VIS system. Each lecture is represented with a keyframe and keyword clouds for each segment around it. The lecture in focus is surrounded by other lectures with related segments. The screenshot was taken from (Bhatt et al., 2013b).

3.10 Conclusion

In this chapter, we introduced a new dataset with metadata from the TED lectures and formulated two benchmark tasks for non-fiction multimedia recommendation utilizing the ground truth available from the TED website. Feature selection experiments over the most active TED users indicated that the most informative data fields for CB methods are the description and the title of each lecture. Using cross-validation, CB using ESA was found to outperform all other CB methods.

We compared in detail content-based, collaborative-filtering, and combined recommendation methods over the test set in two different settings: a cold-start one and a non-cold-start one. The benchmark scores obtained for lecture recommendation are comparable to similar studies on other tasks, e.g. movie recommendation (Cremonesi et al., 2010). We showed that the semantic-based methods (ESA, RP and LSI) were able to make more relevant recommendations than keyword-based ones (TF-IDF) in a cold-start setting, making them particularly applicable to multimedia datasets into which new items are inserted frequently. Even though we focused on the text modality, the proposed similarities can be potentially used for audio and visual modalities as well. However, the CB methods were outperformed by CF ones in a non-cold-start setting, although a combined method using a neighborhood model, user/item biases and TF-IDF similarity achieved performance close to pure CF by utilizing only the popularity bias. The proposed method can be used when newly-added and older items are both present, as it does not rely entirely on collaborative rating similarities.

According to our knowledge, no other dataset with non-fiction audio-visual material contains both content metadata and explicit user feedback (favorites) – a fact that points to the value of the TED dataset for multimedia recommendation. In the future, if other audio-visual collections with explicit feedback such as favorites are made available, they will allow additional evaluations of the proposed algorithms. However, if little or no explicit feedback is available, we show in the next chapter how to leverage other user-generated information such as comments. Lastly, to demonstrate the generality of the proposed content-based methods beyond our generic and personalized recommendation scenarios over the TED talks, we applied them to two other scenarios of lecture and snippet recommendation, namely within the InEvent portal and MUST-VIS system. These systems could be also used for obtaining additional user feedback which will help to further improve them.

4 Adaptive Sentiment-Aware Neighborhood Models for Recommendation

Predicting user preferences based on the content is beneficial when feedback is scarce, but is less effective than CF when feedback is dense, as shown in Chapter 3. Content information itself is not as accurate as collaborative information, likely because it is missing what users may think of the items. This lacking information, however, can be found in comments, in which users often express their opinions. In this chapter, we answer the question of whether the prediction of one-class user preferences (e.g. likes) improves based on sentiment expressed in free-form user comments. We combine the unary feedback with the sentiment of comments through fixed or learned mappings within a sentiment-aware nearest neighbor model, which serves as an effective personalized ranker of items according to their hypothesized relevance to users.

We evaluate the proposed models on top-N recommendation over three real-world datasets: TED lectures, Vimeo videos and Flickr images. The proposed models outperform several competing alternatives in a majority of cases, thus demonstrating the generality of the approach. In particular, the superiority of the adaptive sentiment-aware models validates our hypothesis that there are inherent relationships between sentiments expressed in comments and unary feedback, both at community and individual levels. The improvements due to our models are consistent across all three datasets, they are observed over three different assumptions on the negative class (i.e. items that are not seen or not liked), and they increase as comments become more abundant.

4.1 Introduction

Collaborative filtering methods aim to predict the preferences of an individual user based on items that have been previously rated by other similar users, often in the form of explicit numerical ratings, e.g. on a 1 to 5 scale. However, in many cases, ratings are only expressed through the users' behavior, such as bookmarking, marking as favorite, or liking, i.e. more generally in terms of 'action' or lack thereof, i.e. 'inaction'. This kind of feedback is common in social media and is easier to obtain since it requires considerably less effort from the user side compared with numerical ratings. The main drawback of the unary ratings is the lack of a negative class: it is inherently unsure whether user inaction means that an item was not seen or was seen but not liked. Another drawback, presented in Section 1.1, is that the reasons why a user has liked an item remain unclear; however, understanding them would be useful to explain the recommendations made to the users.

Dealing only with positive explicit feedback is usually referred to as the one-class collaborative filtering problem (Pan et al., 2008). There are several strategies to handle this problem. Hand-labeling negative instances, for example, converts the problem to a standard two-class collaborative filtering one, but is a time-consuming strategy. Alternatively, it is possible to make certain assumptions on the negative class, for example that the missing instances are all unknown, or all negative, but these assumptions bias the recommendation process. More sophisticated assumptions attempt to balance the solution and improve over the two extreme assumptions. In the previous chapter, we dealt with a one-class CF problem using only the most basic assumption that all missing instances are unknown, while here we investigate two additional assumptions on the negative class. Moreover, unlike focusing on the importance of item content for recommendation, here we investigate the importance of sentiment of free-form user comments for recommendation.

Our goal is to extract sentiment information from free-form user comments, which are available in abundance on social media websites, to improve one-class collaborative filtering. Previous studies of sentiment analysis for collaborative filtering have mostly focused on user reviews composed of text and numerical ratings, however, to the best of our knowledge, the study of free-form user comments to complement unary ratings remains largely unexplored. Here, the sentiment information is integrated with a nearest neighbors model similar to the one used in the previous chapter (see Section 3, Eq. 3.5–3.8) into a *sentiment-aware nearest neighbor model (SANN)* by mapping the sentiment scores to user ratings. We investigate several mappings, either direct ones using the output of a sentiment classifier, or adaptive ones, which adapt this output to user ratings through a learning algorithm. We evaluate our proposals against competitive recommendation models, over *three real-world multimedia datasets* – lectures from TED, videos from Vimeo, and images from Flickr – demonstrating consistent improvements that are independent from the negative class assumption and increase with the number of comments.

The chapter is organized as follows. Section 4.2 introduces and analyzes the datasets used in our experiments. Section 4.3 presents our sentiment analysis component, including evaluation results and sentiment-level statistics of the datasets. Section 4.4 formally defines the one-class collaborative filtering problem and describes the models we propose. Section 4.5 presents the experimental setup and evaluation protocol, while Sections 4.6–4.8 describe in detail our empirical studies and analyze their results.

4.2 Multimedia Collections

Unlike the previous chapter, in which we used only on a single dataset from TED, here we use three real-world multimedia datasets that contain both user comments and indications of favorite items, namely TED, Vimeo and Flickr (see Table 4.1). These are popular online repositories of talks, videos and images respectively which contain explicit user feedback of action or inaction, i.e. users mark certain items as favorites, while leaving all the others unmarked. Therefore, the problem of recommendation over these datasets is a *one-class* CF problem. The datasets have different user rating behaviors, comment densities and correlations between the two user-action variables (favorites and comments), as we show below.

4.2.1 Description of the Datasets

The TED dataset that we use in this chapter was already presented in Section 3.2 of the previous chapter. However, we use here a different snapshot of the TED dataset, created in in September 2012 (6 months after our initial crawl), which we also made publicly available by permission from TED owners, along with April 2012 version, under the same Creative Commons license[1]. The reason for re-crawling the dataset was that TED repository was growing quickly and the September 2012 version of the TED dataset contained more data than the April version, namely 1,203 talks (+5% more), 74,760 user profiles (+8.3% more), 129,633 indications of favorite talks (+20% more) and 209,566 comments on talks (+4.2% more). In comparison to the other two datasets used in this chapter – Vimeo and Flickr, presented in the next paragraph – TED users tend to make the longest and most elaborate comments, since they contain on average about 5 sentences and 95 words, compared to about 2 sentences and 20 words for the Vimeo and Flickr datasets.

Vimeo (www.vimeo.com) is an online video sharing repository that allows users to upload, share and view videos. The metadata are accessible in machine-readable format through an API provided by Vimeo. Using this API, in January 2013, we collected 2,000 videos, 255,144 user profiles, 722,474 indications of favorites ("likes") and 278,563 comments from the *nature*, *science*, *art*, *politics* and *music* categories. Flickr is another large online image and video sharing repository (www.flickr.com), which also provides an API giving access to their data. We collected a similar number of items as for Vimeo, namely 1,994 images, 246,272 user profiles, 477,184 indications of favorites ("likes") and 690,798 comments from the *macro* category. As the owners of the Vimeo and Flickr repositories forbid the redistribution of the data obtained through their APIs, we cannot provide these sets along with our distribution of TED metadata.

4.2.2 Statistics of the Datasets

To evaluate the utility of comments for recommendation, we consider from now on the *active users* of TED, Vimeo and Flickr, defined as those who indicated more than five favorites and

[1] http://www.idiap.ch/dataset/ted/.

	All Items and Users						Active Users		
Datasets	**Items**	**Users**	**Favorites**	**Comments**	***cpi***	***wpc***	**Users**	**Favorites**	**Comments**
TED	1,203	74,760	129,633	209,566	174	95.45	4,961	113,241	35,229
Vimeo	2,000	255,144	722,474	278,563	139	18.75	7,071	155,207	32,639
Flickr	1,994	246,272	477,184	690,798	346	22.31	9,963	161,398	304,564

Table 4.1: Statistics of TED (Sep. 2012), Vimeo and Flickr datasets: number of items, users, favorites, comments, average comments per item (*cpi*) and average words per comment (*wpc*). We will use only the *active users* in our experiments, who are defined as those who have indicated more than five favorites and have made at least one comment.

made at least one comment. Note that for the analysis of this section, in contrast to Chapter 3, we adopt a stricter definition for the active users (the minimum threshold of favorites is 5 instead of 1) in order to reduce the long-tail effect, i.e. having too many users with very few favorites (compared to the April 2012 dataset). However, later in this chapter (Sections 3.6 and 4.5) we use the same minimum number of 12 favorites for training methods only on the particularly active users.

Statistics about the active users are given in the rightmost three columns of Table 4.1. Figure 4.1 displays the distributions of favorites and comments per active user, ordered by decreasing number of favorites. The following differences are observed between datasets. In Flickr, users are more likely to make a comment than to mark an item as favorite, while in Vimeo the reverse is true: indeed, in Figure 4.1, red spikes stay mostly below the blue line in (b) and mostly above it in (c). In the TED dataset, the two behaviors can be observed: large and small spikes alternate in Figure 4.1 (a). We can also observe that the long-tail distribution of favorites is approximated more closely by comments in Flickr, followed by Vimeo, and then by TED.

To quantify the differences between the distributions of favorites and comments across users, we measured by Pearson's r coefficient. It turns out that our visual observations match the observations from the correlation between favorites and comments: r coefficient is weak in TED (0.11), moderate in Vimeo (0.33) and strong in Flickr (0.61). Finally, comment density, i.e. the ratio of comments over favorites, is 0.24 for TED, 0.18 for Vimeo and 1.88 for Flickr. Flickr is thus the densest dataset in terms of comments, while TED and Vimeo are much sparser. The variety of these three real-world multimedia datasets will thus allow us to test our proposal over different user behavior patterns and comment densities.

4.3 Sentiment Analysis

The first stage of our proposal for using comments in a one-class CF task is the sentiment analysis of user comments. Given the lack of ground-truth labels to use for training, we use a dictionary-based approach. Specifically, we extend the rule-based (RB) sentiment classifier designed by Wilson et al. (2005), as explained in Pappas et al. (2013), and make this

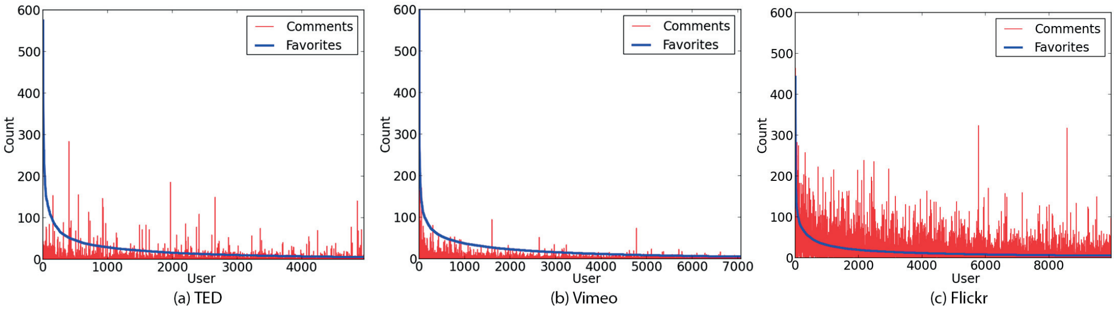


Figure 4.1: Numbers of comments (spikes) and favorites (curve) per active user, ordered by decreasing number of favorites.

implementation freely available.[2]

The RB algorithm first determines whether an expression is neutral or polar and then hypothesizes the polarity of the polar expressions by using a set of contextual rules accounting for phenomena such as negations, modifiers, intensifiers, and polarity shifters. The algorithm relies on the MPQA polarity lexicon[3] for identifying subjective and polar words in a given text. It proceeds through the following steps: (i) text pre-processing, (ii) feature extraction, (iii) polar expression marking, (iv) negation modeling, (iv) intensifier marking, (v) heuristic weighting, and (vi) calculation of the total polarity score. This score is not bounded since its range depends on the size and content of the input texts.

Since we build our sentiment-aware neighborhood models (in subsection 4.4.2) on top of the sentiment classification output, other rule-based classifiers could as well be used instead of the RB one, such as the one from the Pattern[4] or the TextBlob[5] libraries, as well as corpus-based classifiers trained on domain data such as the one from the LingPipe[6] or the Stanford[7] toolkits. However, for the corpus-based classifiers, the ground-truth labels for in-domain free-form comments needed for training are in general costly to generate.

In the rest of this section, we describe how the sentence-level and comment-level polarities are obtained from the RB classifier, we report the results of the sentiment labeling performed by humans, we evaluate the RB classifier, and lastly, we provide sentiment statistics over the three datasets.

[2] http://github.com/nik0spapp/unsupervised_sentiment/.
[3] http://mpqa.cs.pitt.edu/.
[4] http://www.clips.ua.ac.be/pattern/.
[5] https://textblob.readthedocs.org/en/dev/quickstart.html#sentiment-analysis.
[6] http://alias-i.com/lingpipe/demos/tutorial/sentiment/read-me.html.
[7] http://nlp.stanford.edu/sentiment/code.html.

Label	Value		Sentence
positive	+6	+0.50	She is very true in saying that mistakes are part of learning.
negative	−1	−0.05	The problem with the statement: ’the institutions determine work ethics’ is the point of correlation does not equal causation.
neutral	0	0	For years scientists have puzzled over how the sea surface temperature around Antarctica has risen, but sea ice there has been increasing at the same time.

Table 4.2: Examples of sentences with the three possible sentence-level labels from the RB sentiment classifier and their respective polarity values, first non-normalized and then normalized by the length of the sentence.

4.3.1 Sentence-level and Comment-level Polarity Estimation

Given a set of sentences from a user comment c, the RB classifier hypothesizes the polarity of each sentence $s \in c$ as a signed numerical value, noted $pol_{RB}(s)$ (non-normalized). If needed, the sentiment label of the sentence, *positive* or *negative*, is determined from the sign of $pol_{RB}(s)$, with *neutral* if $pol_{RB}(s) = 0$. In Table 4.2, we show examples of sentences with the three possible labels and their polarity values; here, it appears that the labels were correctly determined by the RB classifier. Having assigned polarity values to each sentence, the total polarity value and the label for each user comment are computed. Among the various possibilities for computing the total polarity value of a comment, we we will use here the sum of the polarities of each sentence normalized by the length of the sentence in terms of words, i.e. $pol_{RB}(c) = \sum_{s\in c}(pol_{RB}(s)/|s|)$.

4.3.2 Ground-truth Labeling

To evaluate the RB sentiment analysis component we focused on the binary classification of polarized comments, namely positive or negative, and performed two studies. In Study 1, we performed ground-truth labeling of a subset of the TED comments with three labels: positive, negative or undecided. Six human judges, who were recruited among our English-speaking colleagues, annotated 320 sentences and 160 comments that had been randomly selected from the TED data, with an overlap of about 20% in both cases to assess agreement. Agreement over the shared subset (61 sentences and 29 comments) was found to be $\kappa = 0.83$ for sentences and $\kappa = 0.65$ for comments using Fleiss’ kappa. As agreement was substantial, we subsequently used the entire set as ground truth. After excluding the undecided cases, we obtained 260 labels for sentences and 135 for comments.

To obtain additional ground-truth data, we performed Study 2, a larger-scale study using a crowdsourcing platform.[8] We submitted 1,200 randomly selected comments from TED for annotation on a sentiment scale from 1 to 5, by at least 3 and at most 7 annotators per comment. Crowdflower computes a trust-aware inter-annotator agreement score by testing the annotators’ trust randomly during the annotation process based on majority agreement

[8] http://crowdflower.com/

	Sentences				**Comments**							
	Study 1 (260 labels)				**Study 1** (135 labels)				**Study 2** (526 labels)			
Methods	**P**	**R**	**F**	κ	**P**	**R**	**F**	κ	**P**	**R**	**F**	κ
RB	73.4	76.4	74.9	0.53	75.7	69.7	72.6	0.43	78.6	67.3	72.5	0.48
Rand	47.3	49.9	48.5	-0.01	56.3	50.1	52.9	-0.02	50.0	50.0	50.0	0.00

Table 4.3: Performance of the RB and Rand sentiment classifiers measured with percentage precision, recall, F-measure and Fleiss' kappa.

over a subset of the comments. The agreement between the annotators was found to be 0.74 on a 0–1 scale. We thus obtained 623 positive, 314 neutral and 263 negative ground-truth labels. Furthermore, we created a balanced set for classification containing 263 positive and 263 negative comments, as balanced sets are often used in the literature (Pang et al., 2002; Turney, 2002; Pang and Lee, 2008). Below, we will measure the performance of the RB classifier on the balanced set as well as on the full set of positive and negative labels.

4.3.3 Evaluation of the RB Classifier

The binary classification results of our RB classifier and those of a random baseline (Rand) against the ground truth obtained in Studies 1 and 2 are shown in Table 4.3. When measured by the same kappa score as inter-annotator agreement, our system reaches $\kappa = 0.53$ on sentences (dataset of Study 1) and $\kappa = 0.43$ and 0.48 on comments (dataset of Study 1 and balanced subset of Study 2). As expected, Rand has close to zero κ values. The agreement between the RB classifier and the annotators is thus consistently moderate in both studies. Moreover, the RB classifier reaches a classification accuracy score (F-measure) of 72.5% on comments and 74.9% on sentences. When measured over the full set of labels obtained in Study 2, i.e. the unbalanced sentiment distribution of the comments, the RB classifier reaches higher classification scores (P, R and F) and slightly lower kappa (κ): 89.9% precision, 69.2% recall, 78.2% F-measure, and 0.43 kappa.

The quality of the labels assigned by the RB classifier is comparable to scores found in previous works on binary sentiment classification (Pang et al., 2002; Turney, 2002), in which classification performance reached about 75% F-measure. We will show below that this level of performance is sufficient to improve significantly the one-class CF task. Nevertheless, supervised methods for sentiment classification have been also used and shown useful for our task. Specifically, the study by Sun et al. (2014) on the TED dataset used supervised methods over a balanced set of 600 comments, and reached 87% accuracy using ensemble learning, while our RB classifier reaches 75% accuracy on a comparable set (balanced set of Study 2). The improvement of sentiment classification obtained by Sun et al. (2014) is conditioned on the availability of ground-truth annotation for learning, which limits the portability of their method, unlike general purpose dictionary-based methods such as the above one with the MPQA lexicon. Moreover, supervised learning has a risk of overfitting, which cannot be ruled out from the experiment by Sun et al. (2014) given the small size of their data set (600

	Count			Percentage			Average per Item		
Datasets	***pos***	***neg***	***neu***	***pos***	***neg***	***neu***	***pos***	***neg***	***neu***
TED	130,260	58,171	21,121	62.1%	27.7%	10.0%	108.2	48.3	17.5
Vimeo	195,397	21,726	61,375	70.1%	7.8%	22.0%	97.6	10.8	30.6
Flickr	530,787	22,924	137,087	76.8%	3.3%	19.8%	266.1	11.4	68.7

Table 4.4: Statistics about the sentiment of user comments, as estimated by the RB classifier on the three multimedia datasets.

comments out of 209,566).

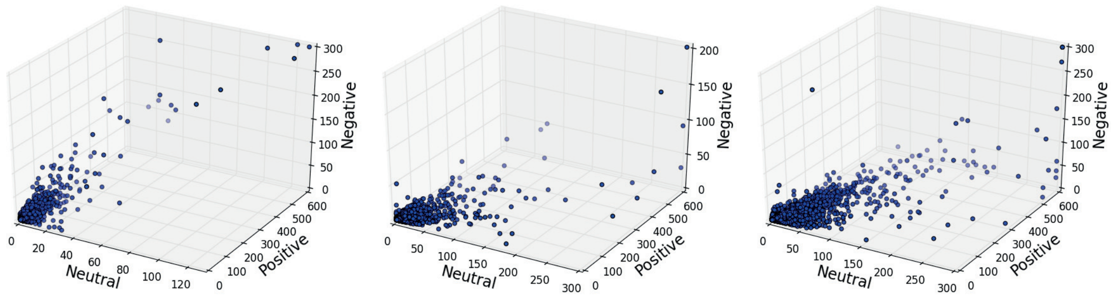
4.3.4 Sentiment Statistics

We labeled all the TED, Vimeo and Flickr comments using the RB classifier with the positive, negative or neutral labels. Statistics about the results of automatic labeling are given in Table 4.4. Based on this classification, the TED dataset appears to have more positive comments than negative ones (62% vs. 27%) and a small percentage of neutral comments (10%). The Vimeo and Flickr comments have an even more skewed distribution of positive vs. negative comments: 70.1% positive vs. 7.8% negative, and 76.8% positive vs. 3.3% negative, respectively.

Figure 4.2 displays the number of positive, negative and neutral labels per user assigned by the RB classifier (for all users, not only active ones). This plot allows us to inspect the distributions of the expressed sentiments per user which are not visible from the table, and to observe how skewed they are. We can observe the similarity in the concentrations of sentiments expressed by the users across datasets, the different shapes of the distributions, but also spot users who deviate from the average. In all the datasets, the average commenting behavior is roughly concentrated in the range of 0 to 200 comments on the positive axis, 0 to 100 on the neutral axis, and 0 to 50 on the negative axis. However, TED users appear to be distributed along the negative axis more than those of the other two datasets, which are flatter. Lastly, we can always observe a number of outliers. For instance, in all datasets, some users have a very large number of negative comments compared to their peers.

4.4 One-class Collaborative Filtering Models

Several applications such as the recommendation of news, bookmarks, images, or videos can be viewed as a one-class CF problem, with training data consisting of binary values expressing the user action or inaction, e.g. bookmarking or marking as liked (Pan et al., 2008). Inaction can mean that an item was either not seen or that it was seen but not liked. This ambiguity of the negative class makes the problem particularly difficult to solve. In the previous chapter, we had assumed that all the missing actions mean that the given item was not seen, while here we investigate two more negative class assumptions. In addition, we propose a method for leveraging comments for one-class CF by mapping their polarities to a format that is usable with neighborhood models.



(a) TED (b) Vimeo (c) Flickr

Figure 4.2: Three dimensional scatter plot of the number of positive, negative and neutral comments per user computed by the RB sentiment classifier. The opacity of the points indicates their depth in space. We can observe the dense areas and the spread over the axes of the sentiment distributions for each dataset. The dense areas for all datasets are concentrated in 0 to 200 of the positive axis, 0 to 100 of the neutral axis, and 0 to 50 of the negative axis; however, TED is more spread on the negative axis than Vimeo and Flickr.

The one-class CF problem is formalized as follows. Let U be the set of users of size $|U| = N_U$ and I the set of items of size $|I| = N_I$. The matrix of user-item ratings is $R = \{r_{ui}\}_{N_I}^{N_U}$ of size $N_U \times N_I$, with $r_{ui} = 1$ indicating a positive rating of item i by user u (e.g. i is a favorite of u) and $r_{ui} = ?$ an absent rating (i was not seen or not liked by u). If one assumes that some of the negative examples have been seen but not liked (or not marked as favorites), then the corresponding ratings become $r_{ui} = 0$. Our goal is to predict the preference of the users in the future, therefore, to evaluate our system, we hide a certain proportion of '1' values per user and measure how well we predict them, as often performed in previous studies.

Next, we present the baseline neighborhood model for one-class collaborative filtering, inspired by Cremonesi et al. (2010), which is the one described and used in Section 3.5.2 of the previous chapter. Here, we will use this model as the starting point for our sentiment-aware neighborhood model.

4.4.1 Neighborhood Models

Neighborhood or Nearest Neighbor (NN) models are often used for CF and have been proven to be quite effective despite their simplicity (Cremonesi et al., 2010). There are several versions of such models, including similarity-based interpolation, jointly-derived interpolation and generalized neighborhood models with parameters computed from the data (Koren and Bell, 2011). We will adopt here the first approach, based on similarities, and focus on item-based neighborhood models as defined in Eq. 4.1 below, with a prediction function $\hat{r}_{ui}$ that estimates the rating of a user u for an unseen item i.

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in D^k(u;i)} (r_{uj} - b_{uj}) d_{ij}}{\sum_{j \in D^k(u;i)} d_{ij}} \tag{4.1}$$

The prediction $\hat{r}_{ui}$, following Cremonesi et al. (2010), is the sum of the bias estimate b_{ui} of a user u towards an item i (defined in Eq. 4.2) and of a similarity score computed using the k most similar items to i that the user u has already rated, i.e. the neighborhood of item i, denoted by $D^k(u;i)$. The similarity score relies on a similarity metric such as cosine distance, as specified at the end of this subsection. The value of k limits the number of items to be taken into account, for efficiency purposes. The coefficient d_{ij} expresses the similarity between items i and j, computed as in Eq. 4.3 below. The denominator in Eq. 4.1 ensures that the predicted values fall in the same range as the known ones, although it is optional for top-N recommendation because we are interested in the ranking of top items rather than their rating.

The bias estimate b_{ui} is defined in Eq. 4.2 as the sum of the average rating μ, the bias estimate b_u of the user u, and the bias estimate b_i of the item i. The bias b_u is computed as the difference between the average rating of a user u, noted $\bar{r_u}$, and the mean μ. Similarly, the bias b_i is the difference between the average rating of an item i, noted $\bar{r_i}$, and the mean μ. Given that the ratings are not real-valued in one-class CF, the biases b_u and b_i are normalized by the total number of ratings of the most rated item, noted r_{max}.

$$b_{ui} = \mu + b_u + b_i, \text{ with: } b_u = \bar{r_u} - \mu \text{ and } b_i = \bar{r_i} - \mu, \text{ where}$$

$$\bar{r_u} = \frac{\sum_{i\in I} r_{ui}}{r_{max}},\ \bar{r_i} = \frac{\sum_{u\in U} r_{ui}}{r_{max}},\ \mu = \frac{\sum_{i\in I} \bar{r_i}}{N_I} \tag{4.2}$$

The coefficient d_{ij} is defined in Eq. 4.3 as the similarity s_{ij} between items i and j multiplied by a coefficient involving the number of common raters n_{ij} and a shrinking factor λ, following Cremonesi et al. (2010). The choice of the optimal value of λ and the optimal size of the neighborhood k used in $D^k(u;i)$ will be determined by cross-fold validation in Section 4.6.

$$d_{ij} = s_{ij}\frac{n_{ij}}{n_{ij} + \lambda} \tag{4.3}$$

The similarity s_{ij} between items i and j can be defined, as in Eq. 4.4, either as the cosine similarity, denoted by COS, or as Pearson's correlation, denoted by PC, following Cremonesi et al. (2010). The vectors for items i and j of size $|U|$ are obtained for each item after creating the co-rating matrix of size $N \times N$ that contains the number of times that two items have been co-rated by pairs of users. Given the vectors of two items $\vec{i}$ and $\vec{j}$, their expected values μ'_i and μ'_j, and their standard deviations σ_i and σ_j, the similarities with COS or PC are computed as follows:

$$s_{ij} = COS(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{||\vec{i}||_2 \times ||\vec{j}||_2} \text{ or } s_{ij} = PC(\vec{i},\vec{j}) = \frac{E[(\vec{i}-\mu'_i)(\vec{j}-\mu'_j)]}{\sigma_i\sigma_j} \tag{4.4}$$

4.4.2 Sentiment-Aware Neighborhood Models

We extend the neighborhood model defined above by proposing a sentiment-aware nearest neighbor model (SANN) with the main purpose of using, in addition to the explicit ratings, the preferences of the users that are implicitly expressed in user-generated texts such as comments. The polarities of the comments are computed by the RB sentiment classifier, and then combined with explicit ratings using a mapping function. Several proposals for such a function are made in this section.

The model in Eq. 4.1 is modified as follows.[9] Firstly, we use a new neighborhood $D'^k(u;i)$ to account for the additional data, and secondly, we define a new rating function r'_{uj} that combines the numerical output of the sentiment classifier and the explicit rating values. Moreover, the additional data from commented items is considered for the creation of the co-rating matrix used for the similarity s_{ij} in Eq. 4.4. Thus, we modify the Eq. 4.1 of the traditional neighborhood model as follows:

$$\hat{r}_{ui} = b_{ui} + \sum_{j \in D'^k(u;i)} d_{ij}(r'_{uj} - b_{uj}) \tag{4.5}$$

In this new definition, $\hat{r}_{ui}$ no longer represents a proper prediction of the rating, but serves only as a ranking function associating user u with item i. The goal of such function is to rank items according to user preference, for example, in the one-class case, higher ranked items are more likely to belong to the positive class than the lower ranked ones. $D'^k(u;i)$ is the neighborhood of the k most similar items that the user has already rated *or commented* and r'_{uj} is the rating function for item j that accounts both for explicit ratings and for those inferred from comments.

We propose the following model for r'_{uj}: if the explicit unary feedback of user u for item j is available (favorite mark, $r_{uj} = 1$) then $r'_{uj} = 1$, but if this unary feedback is not available ($r_{uj} \neq 1$, i.e. it is zero or unknown), then r'_{uj} takes the value of a mapping function m_{uj}. This is a function of the polarity scores of user's u comment(s) to item j, C_{uj}, for which several alternatives are proposed and studied below. Thus, r'_{uj} can be defined as the following piecewise function:

$$r'_{uj} = \begin{cases} 1, & \text{if } r_{uj} = 1 \\ m_{uj}, & \text{if } r_{uj} \neq 1 \end{cases} \tag{4.6}$$

This function augments the standard neighborhood model which makes use of explicit ratings when available (first part) with a sentiment mapping function based on user comments when explicit ratings are absent (second part). It should be noted that, when explicit ratings are available, the user's comments on the item are not considered at all. This is because the explicit rating is the ground truth which represents the actual preference of the user, while

[9] Originally proposed in our 2013 paper (Pappas and Popescu-Belis, 2013b), the model has also been adopted by others (Sun et al., 2014).

Type			Mapping function	Notation
Random	Discrete	-	$m_{uj} = sign_{rand}$	randSANN
Fixed	Discrete	-	$m_{uj} = sign_{RB}(C_{uj})$	sigSANN
	Continuous	-	$m_{uj} = 1 + z_{uj} \cdot pol_{RB}(C_{uj})$	polSANN
Learned	Discrete	Global	$m_{uj} = \begin{cases} \theta, & \text{if } sign_{RB}(C_{uj}) = 1 \\ \upsilon, & \text{if } sign_{RB}(C_{uj}) = 0 \\ \alpha, & \text{if } sign_{RB}(C_{uj}) = -1 \end{cases}$	ltmSANN(global)
		Per user	$m_{uj} = \begin{cases} \theta_u, & \text{if } sign_{RB}(C_{uj}) = 1 \\ \upsilon_u, & \text{if } sign_{RB}(C_{uj}) = 0 \\ \alpha_u, & \text{if } sign_{RB}(C_{uj}) = -1 \end{cases}$	ltmSANN(user)
	Continuous	Global	$m_{uj} = \eta + \zeta \cdot pol_{RB}(C_{uj})$	ltmpolSANN(global)
		Per user	$m_{uj} = \eta + \zeta_u \cdot pol_{RB}(C_{uj})$	ltmpolSANN(user)

Table 4.5: Random, fixed and learned mapping functions m_{uj} from discrete or continuous sentiment scores to ratings, for use within the SANN models.

the mapping function only makes an assumption on how the sentiment of comments from a user might correspond to her preference. Although, for instance, positive comments could consistently accompany a favorite item and vice-versa, empirical observations (see under 'Learned Mappings' in Section 4.4.2) show that this is not always the case. For these reasons, in our model, explicit ratings always have precedence over those inferred from comments. In what follows, we define three types of mapping functions m_{uj} from sentiment scores to ratings, which are summarized in Table 4.5.

Random Mapping

As a baseline, we compare a random sentiment classifier with the RB sentiment classifier, which will be used in the fixed and learned mapping functions. The random mapping, noted as randSANN, simply assigns a random class value $sign_{rand}$ (either 1, 0 or -1) to the sentiment of a user comment. Hence, this baseline does not extract any actual preference information from text.

Fixed Mappings

We first propose two different mapping functions that rely on the output of the RB classifier (polarity score), one based on the discretized output, represented in Table 4.5 as "fixed → discrete", and the other using the actual real-valued output, represented as "fixed → continuous". Let C_{uj} be the set of all comments made by a user u on an item j. The first function, denoted by sigSANN (for 'sign'), assigns a rating value for a user-item pair according to the sign of the average polarity score of the comments: $sign_{RB}(C_{uj}) = sign(mean(\{pol_{RB}(c) \mid c \in C_{uj}\}))$, where $pol_{RB}(c)$ is the polarity of a comment defined in Section 4.3.1.

The second mapping function, polSANN, uses the real-valued output of the RB classifier with a normalization factor and an offset. The polarity score of a given user u for a particular item j is $pol_{RB}(C_{uj}) = mean(\{pol_{RB}(c) \mid c \in C_{uj}\})$ and the normalization factor is $z_{uj} = 1/(1 + |C_u| \cdot |\{c \text{ s.t. } c \in C_u \wedge sign_{RB}(c) = sign_{RB}(C_{uj})\}|)$. This normalization penalizes the impact of the polarity score in proportion to the total number of a user's comments times the number of the user's comments of the same class as the predicted one. In other words, without normalization, polSANN estimates that users who always comment positively are biased towards positive feedback, and similarly for negative comments. The normalization z_{uj} aims to reduce these effects on the rating prediction $\hat{r}_{ui}$.

Learned Mappings

The mapping functions described above combine sentiment scores with ratings based on the intuition that positive scores imply positive preferences and negative scores imply negative preferences. However, this intuition may not be accurate in all cases, for example, a user could write positive comments about non-favorite items. To support this claim, we have examined the number of times ratings (i.e. marking as favorites) appear with either positive, negative or neutral comments, as labeled by the RB classifier. The results, listed in Table 4.6, show that in a majority of cases when a favorite is accompanied by a comment the latter is a positive one: 76.5% of the times on TED, 77.8% on Vimeo and 83.1% on Flickr. However, ratings can also be followed by neutral or negative comments, which motivate the need to employ learning methods to handle such cases by learning either global or individual behavior patterns from the data, as defined hereafter.

1. **Mapping discrete scores globally**: We introduce three parameters θ, v and α respectively for positive, neutral and negative comments, which define the mapping according to the piecewise function presented in Table 4.5 under "learned → discrete → global". This mapping is denoted by ltmSANN(global), with 'ltm' standing for "learning to map". Inspired by the global neighborhood models used by Koren and Bell (2011), we propose to learn the three parameters by minimizing the following regularized least squares objective on the training set:

   $$\min_{\theta,v,\alpha} \sum_{(u,i)\in R_{known}} (r_{ui} - \hat{r}_{ui}(\theta,v,\alpha))^2 + \epsilon(\theta^2 + v^2 + \alpha^2) \tag{4.7}$$

   where R_{known} is the set of all the user-item pairs (u,i) with known ratings, $\hat{r}_{ui}(\theta,v,\alpha)$ is the prediction made by the sentiment-aware rating predictor from Eq. 4.5 which now depends on θ, v and α (due to r'_{uj} and m_{uj}), and ϵ is the regularization hyper-parameter. Intuitively, the above objective, which is influenced by the user rating behavior, will learn the optimal parameters of the mapping function in order to make $\hat{r}_{ui} \approx 1$ if the actual rating r_{ui} is equal to 1, or close to 0 otherwise.

2. **Mapping discrete scores per user**: A similar mapping can also be learned for each user, considering that the discrete scores of the sentiment classifier may have a different

	Favorites accompanied by comment(s) from the same user				
	Total		**Positive**	**Neutral**	**Negative**
TED	7,053	(6.2% of the total)	5,385 (76.5%)	548 (7.7%)	1,120 (15.8%)
Vimeo	11,883	(7.7% of the total)	9,246 (77.8%)	1,898 (16.0%)	739 (6.2%)
Flickr	84,119	(52.1% of the total)	69,910 (83.1%)	12,208 (14.5%)	2,001 (2.4%)

Table 4.6: Total number of ratings (favorites) from active users which are accompanied by at least one comment from the same user, and the proportion of positive, neutral and negative comments among them, as labeled by the RB classifier.

impact on recommendation depending on the user. We introduce three vectors of user parameters, $\theta^* = \{\theta_u\}^{N_u}$, $\upsilon^* = \{\upsilon_u\}^{N_u}$ and $\alpha^* = \{\alpha_u\}^{N_u}$ respectively for positive, neutral and negative comments. These vectors define a user-specific mapping through the piecewise function shown in Table 4.5 under "learned → discrete → per user", denoted by ltmSANN(user). Similarly to ltmSANN(global), the parameters are computed by minimizing the following objective:

$$\underset{\theta^*,\upsilon^*,\alpha^*}{min} \sum_{(u,i)\in R_{known}} (r_{ui} - \hat{r}_{ui}(\theta_u, \upsilon_u, \alpha_u))^2 + \epsilon(\theta_u^2 + \upsilon_u^2 + \alpha_u^2) \tag{4.8}$$

3. **Mapping continuous scores globally**: We introduce two parameters η and ζ, respectively for the offset and slope of the linear relationship between the continuous score (polarity) of the RB classifier and the ratings, as defined in Table 4.5 under "learned → continuous → global", denoted by ltmpolSANN(global). This model is a generalized version of polSANN, with the η and ζ parameters being identical for all users, and learned from the data. The parameters are computed by minimizing the following objective:

$$\underset{\eta,\zeta}{min} \sum_{(u,i)\in R_{known}} (r_{ui} - \hat{r}_{ui}(\eta, \zeta)^2 + \epsilon(\eta^2 + \zeta^2) \tag{4.9}$$

4. **Mapping continuous scores per user**: Finally, we also define user-specific linear relationships between the continuous score (polarity) of the RB classifier and the ratings, denoted by ltmpolSANN(user). We introduce a parameter η and a vector of user parameters $\zeta^* = \{\zeta_u\}^{N_U}$ respectively for the offset and the user-specific slope of the linear relationships. The function is defined in Table 4.5 under "learned → continuous → per user", and is denoted by ltmpolSANN(user). This model is a generalized version of polSANN too, by adopting the user-specific normalization z_{uj} to the data. The objective to minimize is now:

$$\underset{\eta,\zeta^*}{min} \sum_{(u,i)\in R_{known}} (r_{ui} - \hat{r}_{ui}(\eta, \zeta_u)^2 + \epsilon(\eta^2 + \zeta_u^2) \tag{4.10}$$

ALGORITHM 1: Learning the mapping function globally (i.e. across all users). The algorithm which currently corresponds in Eq. 4.7 can be adapted to user-specific mapping scores by replacing the parameters θ, υ, α with the user parameter vectors $\theta^*, \upsilon^*, \alpha^*$.

```
Data: User ratings $R = \{r_{ui}\}_{N_I}^{N_U}$, User comments $C = \{C_{uj}\}_{N_I}^{N_U}$
Result: Parameters $\theta, \upsilon, \alpha$
$set(max\_iter, \gamma, \epsilon)$  # Set maximum number of iterations and hyper-parameters
$initialize(\theta, \upsilon, \alpha)$  # Initialize model parameters
while not converged do
    for $(u, i) \in R_{known}$ do
        for $j \in D'^k(u; i)$ do
            # Compute error for gradient steps
            $e_{ui} = r_{ui} - \hat{r}_{ui}(\theta, \upsilon, \alpha)$
            # Perform gradient steps for $\theta, \upsilon, \alpha$
            if $sign_{RB}(C_{uj}) = 1$ then
                $\theta = \theta + \gamma \cdot e_{ui} - \epsilon \cdot \theta$
            else if $sign_{RB}(C_{uj}) = 0$ then
                $\upsilon = \upsilon + \gamma \cdot e_{ui} - \epsilon \cdot \upsilon$
            else if $sign_{RB}(C_{uj}) = -1$ then
                $\alpha = \alpha + \gamma \cdot e_{ui} - \epsilon \cdot \alpha$
            end
        end
    end
    # Check for convergence based on tolerance and $max\_iter$
    $e = \frac{1}{R_{known}} \sum_{(u,i) \in R_{known}} e_{ui}$ #  Mean absolute error
    if $e_{prev} - e < tolerance \;\; or \;\; iter > max\_iter$ then
        converged = True
    end
    $e_{prev} = e$
    $iter{+}{+}$
end
```

Algorithm for Learning the Mappings

To minimize the above objectives (Eq. 4.7–4.10), we define a simple stochastic gradient descent solver inspired by the parameter estimation for global neighborhood models proposed by Koren and Bell (2011), although other optimization techniques can be used as well. The algorithm loops through all known ratings in R_{known} and for each (u, i) pair it modifies the parameter values in the opposite direction of the gradient of the prediction error $r_{ui} - \hat{r}_{ui}$. The algorithm designed to learn the parameters of Eq. 4.7 is shown as Algorithm 1. The algorithm is easily adapted to Eq. 4.8, on the one hand, and to Eq. 4.9 and Eq. 4.10 on the other hand.

The best values of the γ and ϵ hyper-parameters of the algorithm (respectively the step size and the regularization factor) will be determined empirically using cross-validation in Section 4.6. Likewise, the θ, υ, and α parameters can be initialized with random values or with the discrete class values of the sentiment classifier (1, 0, -1 as in the first fixed mapping, SANN); the best option will again be determined empirically. The overall complexity of Algorithm 1 is $\mathcal{O}(k \cdot |\{(u,i) \in R_{known}\}|)$, which is linear with respect to the input size, given that the size of the neighborhood k is usually considerably smaller than the number of non-empty elements in the user-item matrix R.

4.4.3 Negative Class Assumptions

The inherent problem of one-class CF is the lack of explicit negative feedback, in other words the uncertainty of the class to which an unknown rating belongs. An approach that is commonly used for one-class CF problems (see also Section 2.1.4) is to make an assumption about the distribution of the negative class. We describe here two intuitive assumptions used in previous studies and we propose an additional one that is a trade-off between the two. In Section 4.8.2 below, we will show that exploiting user comments for recommendation improves results for all three assumptions.

1. **All Missing as Unknown (AMAU):** All missing ratings are ignored, and only positive ones are used, with CF algorithms that model only non-missing data (Nati and Jaakkola, 2003). A direct consequence is that these models can only predict positive examples but not negative ones.
2. **All Missing as Negative (AMAN):** All missing ratings are treated as negative examples. This assumption has been shown empirically to perform quite well (Pan and Scholz, 2009), even if it introduces a potentially large imbalance between classes. The main drawback is that a classifier trained using this assumption will likely be biased towards the negative class.
3. **Equal-to-positive Missing as Negative (EMAN):** We propose this more nuanced approach, which treats as negative instances a random sample of the missing instances, equal in size to the number of positive instances per user. In this way, the model can be trained with equal numbers of examples from both classes.[10]

4.4.4 Baseline Models

We will compare the SANN models with several baselines in order to show that the additional information included in the SANN models, and not captured by the baselines, improves performance of one-class CF.

1. **TopPopular**: A user-independent method which recommends a fixed list of the most

[10] More sophisticated negative class assumptions have been studied by Pan et al. (2008) and Pan and Scholz (2009), who showed that they can marginally improve over the two extreme ones (AMAU and AMAN), but at the expense of a much greater complexity. Therefore, we limit our exploration of negative class assumptions to the three listed in the text.

popular items, i.e. those that received the most ratings across users, as used also in Chapter 3 (e.g. Figure 3.7).

2. **Nearest Neighbors (NN)**: The standard neighborhood model presented in Section 4.4.1 above. This model will be optimized with respect to the number of nearest neighbors k and the shrinking factor λ. We will test each of the three negative class assumptions, either with normalization, denoted by normNN, or without it, denoted by NN.

3. **Singular Value Decomposition (SVD)**: A common matrix factorization method, where the SVD of a user-item matrix R is a factorization of the form: $R = U\Sigma V^T$, where U is a unitary matrix $(M \times M)$, Σ is a diagonal matrix with non-negative real numbers on the diagonal and V^T is the transpose of the unitary matrix V $(N \times N)$. For the SVD algorithm we use the AMAN assumption (all unknown examples set to 0), because the matrices need to be filled with numeric values. The model will be optimized with respect to the low-rank dimensionality hyper-parameter l, i.e. the number of values to be considered from the diagonal matrix Σ. For our experiments we use the implementation of SVD provided in the *Python-recsys* library.[11]

4. **Non-negative Matrix Factorization (NMF)**: This is another common low-rank matrix approximation method, which decomposes a non-negative matrix R into two non-negative matrix factors W $(N \times l)$ and H $(l \times M)$ such that $R \approx WH$. Again, l is the low-rank dimensionality of the approximation, generally chosen to be smaller than N or M, so that W and H are smaller than R. To find the approximate factorization, we experimented with three different cost functions (Euclidean distance, generalized Kullback-Leibler (KL) divergence, or connectivity matrix convergence) and selected KL as the best performing one (see Section 4.6.3 for the actual scores). For the same reason as above, we will test here the AMAN assumption only. The model will be optimized with respect to the low-rank approximation hyper-parameter l. We use the implementation from the *nimfa* library (Zitnik and Zupan, 2012).[12]

5. **Sparse Non-negative Matrix Factorization (SNMF)**: A low-rank matrix approximation method which enforces sparsity on the learned factors. It uses an alternating least squares optimization objective with non-negativity constraints to approximate $R \approx WH$ (Kim and Park, 2007). Sparseness can be enforced either on the left factor, noted as SNMF/L, or on the right factor, noted as SNMF/R, by using the ℓ_1-norm. The model will be optimized with respect to the low-rank approximation hyper-parameter l and the ℓ_1 regularization hyper-parameter ϵ. Similarly to SVD and NMF, we will make the AMAN assumption, and use the implementation provided in the *nimfa* library.

[11] http://recsyswiki.com/wiki/python-recsys/.

[12] http://nimfa.biolab.si/.

Data	Set	Favorites	Comments	Users
TED	Training	92,560	22,259	4,961
	Testing: sparse	18,027	15,108	2,809
	Testing: dense	8,351	12,918	1,090
Vimeo	Training	126,954	22,303	7,071
	Testing: sparse	24,628	16,338	4,150
	Testing: dense	8,879	11,640	1,111
Flickr	Training	132,937	198,098	9,963
	Testing: sparse	21,540	133,074	4,182
	Testing: dense	9,807	86,792	1,100

Table 4.7: Numbers of favorites, comments and users per training/testing sets in the TED, Vimeo and Flickr datasets.

4.5 Evaluation Protocol and Metrics

For each of the three datasets, 80% of each *active user*'s[13] positive ratings (values of '1') are used for training and the remaining 20% are held out for testing (in the experiments from Sections 4.7.2 and 4.8). We will use two specific subsets of each test set, which include only users among active users only those who are particularly active, so that enough training and testing items are available for each of them. For the *dense* sets, we filter out from the entire testing sets users with fewer than 12 ratings and fewer comments than, respectively, 2 for TED, 3 for Vimeo and 39 for Flickr, so that enough users will be included (about 1100 for each set). For the *sparse* sets, we filter out the users with fewer than 12 ratings and 1 comment. These sets contain respectively around 7% (dense sets) and 16% (sparse sets) of all active users' ratings for all three datasets. Additional statistics are shown in Table 4.7. As a matter of fact, the sparse set is close to the entire test set of 20% of the active users. The optimization of the hyper-parameters is made on the training set using 5-fold cross-validation. Similarly to the sparse set, we also filter out from the test folds used in cross-validation (Section 4.6 and 4.7.1) the users with fewer than 12 ratings and 1 comment.

We evaluate all methods for one-class CF using the framework of top-N personalized recommendation, i.e. measuring how many items selected by each method in a set of N items actually match the user favorites hidden in the test set, for varying values of N. For this task, the error metrics such as RMSE are not the most appropriate ones to be used, since a top-N recommender does not need to infer item ratings (Cremonesi et al., 2010). Instead, it is more informative to apply the classification accuracy metrics of precision, recall and F-measure (Shani and Gunawardana, 2011). The average precision at N (noted AP) and the mean average precision at N (noted MAP) are respectively given in the following equations:

$$\mathrm{AP}(N) = \frac{1}{|U|}\sum_{u\in U}\frac{|\mathcal{T}_u \cap \mathcal{R}_{u@N}|}{N}\ ;\ \mathrm{MAP}(N) = \frac{1}{|U|}\sum_{u\in U}\left(\frac{1}{N}\sum_{1\le v\le N}\frac{|\mathcal{T}_u \cap \mathcal{R}_{u@v}|}{v}\right) \tag{4.11}$$

In both equations, N is the bound of top recommendations, $|U|$ is the total number of users

[13] As a reminder 'active users' are the users with at least five favorites and one comment (Section 4.2.2).

Hyper-params	$\boldsymbol{k}$	$\boldsymbol{\lambda}$	$\boldsymbol{l}$	$\boldsymbol{\epsilon}$	$\boldsymbol{\gamma}$
Range	1 – 50	1 – 50	5 – 100	4e-5 – 4e+5	4e-5 – 4e+5
Step	+2	+5	+5	×10	×10
normNN(AMAU)	19(T)/7(V)/23(F)	25(T)/40(V)/20(F)			
NN(AMAU)	27(TF)/49(V)	10(TF)/5(V)			
NN(AMAN)	7(TF)/9(V)	50(TVF)			
NN(EMAN)	17(TF)/27(V)	15(TF)/5(V)			
sigSANN(AMAU)	5(TF)/49(V)	10(TF)/5(V)			
sigSANN(AMAN)	5(TVF)	50(TVF)			
sigSANN(EMAN)	1(TF)/9(V)	5(TF)/15(V)			
SVD			5(TVF)		
NMF			5(TVF)		
SNMF			10(VF)/75(T)	1e-2(VF)/1e-3(T)	
ltmSANN (global)				4e-5(T)/4e-2(V)/ 4e-4(F)	4e-4(VF)/4e-1(T)
ltmSANN (user)				4e-3(T)/4e-2(V)/ 4e-5(F)	4e-3(VF)/4e-2(T)
ltmpolSANN (global)				4e-5(TV)/4e-3(F)	4e-1(TVF)
ltmpolSANN (user)				4e-4(T)/4e-5(V)/ 4e-2(F)	4e-1(T)/4e-0(VF)

Table 4.8: Optimal values of the hyper-parameters of each model found by grid searches (see also examples in Figure 4.3) over the TED (T)/Vimeo (V)/Flickr (F) datasets. The ranges of explored values and steps are shown in the second and third lines respectively. k is the number of nearest neighbors, λ is the similarity shrinking factor, l is the latent factor of SVD and NMF models, ϵ is the regularization hyper-parameter for ltmSANN and SNMF and γ is the step of gradient descent for ltmSANN.

in U, $\mathcal{T}_u$ is the set of items that a user u has marked as favorites and $\mathcal{R}_{u@N}$ is the set of top-N recommendations of the model for the user u. To compute average recall (AR), we divide by the number of items that a user u has marked as favorites, $|\mathcal{T}_u|$, instead of N for AP. Similarly, mean average recall (MAR) is computed by dividing by $|\mathcal{T}_u|$ instead of v for MAP. The average F-measure (AF) and mean average F-measure (MAF) are respectively computed as the harmonic means of the previous two metrics.

$$\mathrm{AF}(N) = 2 \cdot \frac{AP(N) \cdot AR(N)}{AP(N) + AR(N)}; \;\; \mathrm{MAF}(N) = 2 \cdot \frac{MAP(N) \cdot MAR(N)}{MAP(N) + MAR(N)} \tag{4.12}$$

In the following sections, we experiment with variable values of N, from 1 to 50, and base most of our conclusions on the top 50 recommendations.

4.6 Optimizing the Hyper-Parameters and Selecting the Models

In this section, we discuss the optimization of the hyper-parameters and the selection of the models by cross-validation over each training set. In Section 4.7, we provide a comparison of the scores of all models on the training sets with cross-validation, and then on the held-out sparse and dense test sets. In Section 4.8, we discuss the results, explaining the effectiveness of sentiment-aware nearest neighbor models under various configurations.

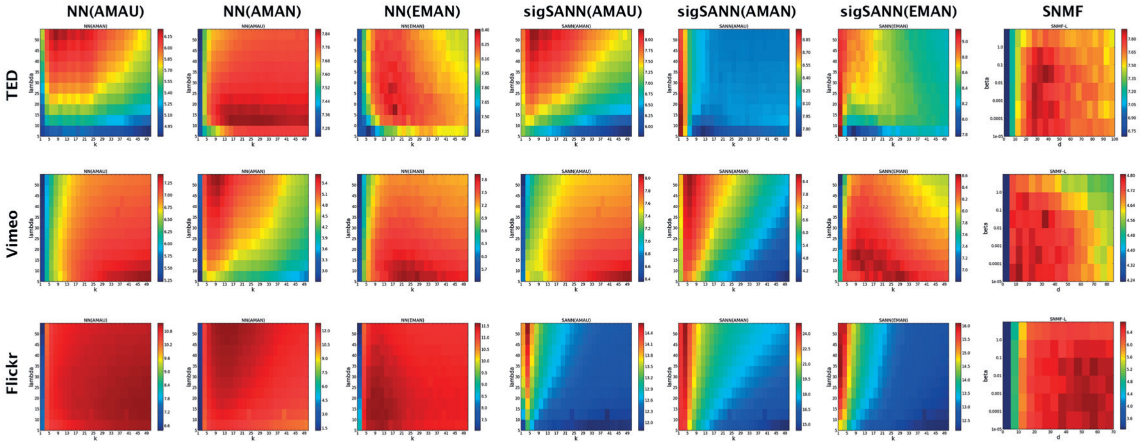


Figure 4.3: Performance heatmaps of the MAP score (at 50) from the grid search for NN, sigSANN and SNMF models over TED, Vimeo and Flickr training sets, with 5-fold cross-validation. The size of the neighborhood k is on the x-axis and the shrinking factor λ for the similarity d_{ij} between items is on the y-axis. Higher scores are in red and lower ones in blue, on scales adapted to each heatmap. The heatmaps show the concentration of regions with the highest performance for each dataset and model.

4.6.1 Selection Method and Comparison of Values

Table 4.8 lists the hyper-parameters on which each model relies and the ranges of values that we searched for each parameter with the incrementation steps. The optimal values per model were obtained from grid search, i.e. a complete search over all the combinations of values, with 5-fold cross-validation. These time-consuming computations were carried out using Idiap's computation grid with about 400 processor cores. The optimal values led to the cross-validation scores over the training sets presented in Tables 4.9 and 4.10 and to the scores over the held-out sets presented in Tables 4.11 and 4.12, which are discussed below.

The results of grid searches for optimal hyper-parameter values for the NN, sigSANN and SNMF models are represented using heatmaps in Figure 4.3. Each point represents the MAP at 50 score obtained through 5-fold cross-validation with the corresponding hyper-parameter values. In most of the cases, the optimal values (red colors) are well inside each range of values, indicating that the ranges were sufficiently large to ensure that a global optimum was found. When this was not the case, we extended the ranges to ascertain this fact.

The sigSANN model appears to reach its best performance on smaller values of the size of the neighborhood k than standard NN in most cases. This happens presumably because the additional sentiment information incorporated into sigSANN allows the neighborhood model to find fewer but more relevant neighbors compared to NN, which requires more neighbors but which are likely less relevant. In cases where the heatmap patterns for NN and sigSANN are similar, namely on TED under the AMAU assumption, and under the AMAU and EMAN assumptions on Vimeo, both models reach their best values for similar values of k.

The sigSANN model outperforms or is comparable with NN over the full spectrum of k: for instance, the lowest MAP scores of sigSANN(AMAU) (6%, 6.4% and 12% for TED, Vimeo and Flickr respectively), are comparable to the best scores for NN(AMAU) (6.1%, 7,2% and 10.8%), and similar observations can be made for the other negative class assumptions. The differences between NN and sigSANN are larger on Flickr, likely because this dataset contains far more comments to be exploited by SANN than TED or Vimeo (Table 4.1). Lastly, the NN and sigSANN models appear to have a stable performance across the three negative class assumptions over Flickr – i.e. the optimal values appear in the same areas of the heatmap – while on Vimeo and TED they are less stable.

4.6.2 Hyper-Parameters of Neighborhood Models

The neighborhood models rely on two hyper-parameters, namely the size of the neighborhood k and the shrinking factor λ for the similarity d_{ij} between items i and j (see Section 4.4.1). The grid search examined 250 different models for each possible negative class assumptions, ending up with 750 NN models and 750 SANN models (see Table 4.8). In addition, for the s_{ij} similarity included in d_{ij}, we experimented with two proximity measures, Cosine Similarity and Pearson's Correlation. The latter performed significantly better, thus from here on all models will use it. For the learned SANN models, namely ltmSANN(global), ltmSANN(user), ltmpolSANN(global), and ltmpolSANN(user), the optimization of their γ and ϵ hyper-parameters was performed after setting k and λ to their optimal values for the fixed SANN models.

Turning now to the differences between discrete and continuous SANN models, as well as global or per user ones, we observed that on TED, the best performance was achieved by ltmpolSANN(user), followed by ltmSANN(user), and then ltmSANN(global) and ltmpolSANN(global). On the Vimeo and Flickr datasets, the ltmSANN(global) performed the best, followed by ltmSANN(user), ltmpolSANN(user), and ltmpolSANN(global). Therefore, while the use of actual polarity scores (as opposed to their signs only) in a user-dependent way is optimal for TED, this is not the case for the other two datasets. Possible causes for this include the lack of variability in the scores, or its weak effect on the rating behavior of the users, or the smaller reliability of polarity values from the RB classifier compared to their signs. In what follows, we will report the results of the best-performing model for each dataset, and unify their notation, for simplicity, as ltmSANN.

4.6.3 Hyper-Parameters of Low-rank Factorization Models

The SVD models rely on a single hyper-parameter which is the low-rank dimensionality l. We performed a linear search to find the best performance among 20 SVD models which were obtained by uniformly varying l from 5 to 100. The NMF and SNMF models also rely on l, and SNMF moreover relies on the regularization hyper-parameter ϵ. Similarly to SVD, for NMF we performed a linear search to find the best performance among 20 NMF models (l from 5 to

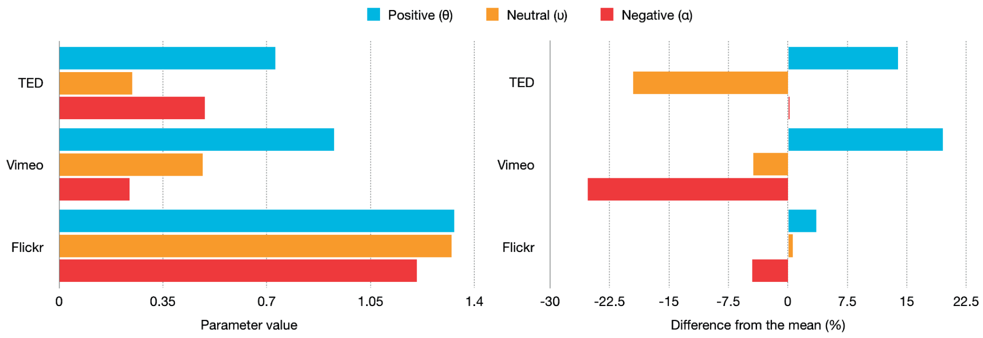


Figure 4.4: Example of learned weights of positive (θ), neutral (υ) and negative (α) comments for the ltmSANN (global) model over TED, Vimeo and Flickr. The left side displays the values of the parameters while the right side shows their difference, as a percentage, from the mean value of each dataset.

100). For SNMF, we performed grid search over a range of values for l and ϵ (5 to 100 and 10^{-6} to 10^{6} respectively) ending up with 260 different SNMF models. This was repeated for two sparsity options, namely over the left factor W (SNMF/L) or over the right factor H (SNMF/R), ending up with 520 models. The highest performance was obtained by applying sparsity on the left factor (SNMF/L) for Vimeo and Flickr datasets, and on the right factor for TED dataset.

The optimal values of their parameters, i.e. l and ϵ which lead to the highest performance are shown in Table 4.8, namely $l = 10$, $\epsilon = 10^{-2}$ for Vimeo and Flickr, and $l = 75$, $\epsilon = 10^{-3}$ for TED.

In addition, we experimented with the three different cost functions mentioned in Section 4.4.4 and found out that Kullback-Leibler (KL) divergence performed best. Therefore, all our NMF and SNMF models use it. For simplicity reasons, as in the case of ltmSANN, we will use a common name (SNMF) for the best performing SNMF model per dataset.

4.6.4 Examples of Globally Learned Parameters for Discrete Sentiment Output

Figure 4.4 shows examples of learned θ, υ and α parameters for the ltmSANN(global) model.[14] The values of these parameters indicate the importance of, respectively, positive, negative or neutral comments for the recommendation task: the greater the value the more important the sentiment class. The optimal values of the parameters found by learning (Algorithm 1) are similarly ordered for each dataset, with the θ parameter (weight of positive comments) having the greatest value in all cases. The α parameter (weight of negative comments) has the smallest absolute value on Vimeo and the highest value on TED. This means that the negative comments on Vimeo are more rarely followed by positive feedback (i.e. rating as favorite), while on TED, users tend to leave negative comments even though they liked a talk, possibly as a result of disagreements with other TED users in a discussion thread.

[14] These are not hyper-parameters optimized through grid search, but parameters learned by the optimization method in Algorithm 1.

	TED (5-fold c-v)			Vimeo (5-fold c-v)			Flickr (5-fold c-v)		
Model	**MAP**	**MAR**	**MAF**	**MAP**	**MAR**	**MAF**	**MAP**	**MAR**	**MAF**
TopPopular	3.77	12.07	5.75	2.92	7.86	4.26	1.95	6.13	2.96
normNN(AMAU)	4.59	13.76	6.88	3.59	9.18	5.16	2.26	6.77	3.39
NN(AMAU)	5.22	15.88	7.86	5.20	12.97	7.42	7.69	20.25	11.15
NN(AMAN)	4.23	11.96	6.24	4.05	9.28	5.64	8.88	22.98	12.81
NN(EMAN)	5.59	16.86	8.40	5.57	13.68	7.92	8.01	21.09	11.61
SVD	4.45	13.30	6.67	3.32	8.64	4.80	2.16	6.63	3.26
NMF	5.08	15.37	7.64	3.86	9.49	5.49	4.45	12.03	6.50
SNMF	5.33	15.87	7.98	3.91	9.59	5.56	5.07	13.29	7.34
sigSANN(AMAU)	6.03	17.51	8.97	5.65	14.04	8.06	10.05	27.09	14.66
sigSANN(AMAN)	5.63	15.32	8.24	**6.46**	**13.49**	**8.73**	**17.21**	**46.01**	**25.05**
sigSANN(EMAN)	**6.16**	**17.84**	**9.15**	6.05	14.98	8.62	10.94	29.51	15.96
sigSANN vs. best (%)	+10.1	+5.8	+8.9	+15.9	-1.8	+10.2	+93.8	+100.21	+95.55
randSANN	4.67	13.93	7.00	4.08	9.42	5.69	8.18	21.13	11.80
polSANN	6.41	18.42	9.51	7.06	14.71	9.54	18.39	49.79	26.86
ltmSANN(global)	6.50	18.74	9.65	**7.11**	**14.75**	**9.59**	**18.45**	**50.33**	**27.00**
ltmSANN(user)	6.52	18.86	9.69	7.07	14.71	9.55	18.43	49.90	26.92
ltmpolSANN(global)	6.36	18.42	9.46	6.94	14.54	9.40	18.07	49.06	26.42
ltmpolSANN(user)	**6.56**	**19.03**	**9.76**	7.05	14.71	9.53	18.41	49.89	26.90
ltmSANN vs. sigSANN (%)	+6.8	+6.6	+6.6	+10.0	+9.3	+9.8	+7.2	+9.3	+7.7

Table 4.9: Performance of each recommendation model using MAP, MAR and MAF metrics at 50 with 5-fold cross-validation on the training set. The percentage of improvement of the best SANN model (indicated in bold) over the best baseline (underlined) is displayed in the sixth row from the bottom. The last row displays the additional improvement obtained with ltmSANN (again in bold) compared to the best SANN model for each dataset. All improvements are significant among the three datasets (pairwise t-statistic, $p < 0.01$).

Regarding the neutral parameter (v), the smallest value is on TED while the highest one is on Flickr. It appears that neutral comments on TED imply absence of feedback, while on Flickr they are more likely to be followed by positive feedback. Furthermore, parameters learned on Flickr have higher values than those learned on the other datasets, showing that comments are more important for recommendation on this dataset. This finding matches the observation that the Flickr comments have the highest correlation with the one-class ratings, as shown in Section 4.2.2.

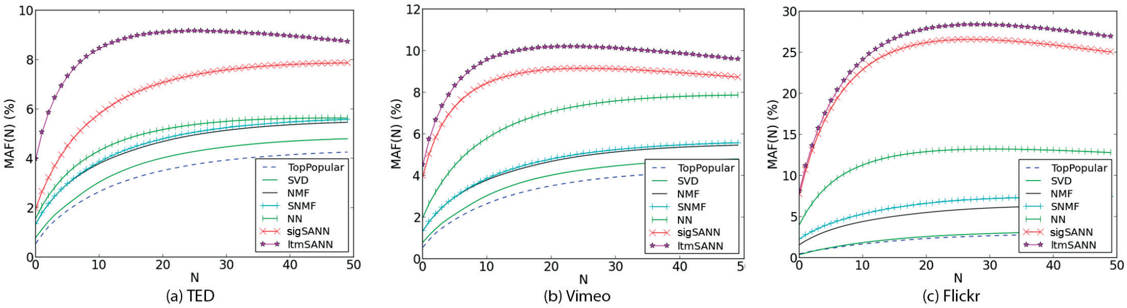
4.7 Personalized Multimedia Recommendation

4.7.1 Results over the Training Sets with Cross-Validation

In Table 4.9 we present the results of 5-fold cross-validation over the TED, Vimeo and Flickr training sets. The sigSANN model performed significantly better (pairwise t-statistic, $p < 0.01$) than all the other models not using comments, namely about 9% improvement for TED, 10% for Vimeo and even 95% for Flickr using MAF at 50. The adaptive SANN models, i.e. ltmSANN and its variants, further improved over sigSANN (bottom part of Table 4.9), namely about 6% improvement on TED, 10% improvement on Vimeo and 8% on Flickr. To examine the effect of N on the reported improvements, Table 4.10 displays (for the best-performing methods) the

	TED (5-fold cv)				Vimeo (5-fold cv)				Flickr (5-fold cv)			
	MAF at N				MAF at N				MAF at N			
Methods	**10**	**20**	**30**	**40**	**10**	**20**	**30**	**40**	**10**	**20**	**30**	**40**
TopPopular	4.69	5.46	5.72	5.77	2.53	3.45	3.90	4.13	1.68	2.35	2.68	2.87
SVD	5.55	6.40	6.62	6.64	2.89	3.96	4.44	4.68	1.85	2.59	2.95	3.16
NMF	6.12	7.03	7.30	7.35	3.67	4.62	5.11	5.35	4.42	5.51	6.07	6.30
SNMF	6.22	7.28	7.62	7.70	3.71	4.73	5.21	5.46	5.33	6.62	7.19	7.38
NN	6.92	8.06	8.04	8.46	5.60	7.00	7.56	7.79	11.31	12.93	13.24	13.10
sigSANN	8.70	9.58	9.61	9.44	8.30	9.08	9.14	8.97	22.93	26.20	26.53	25.91
ltmSANN	**9.31**	**10.19**	**10.18**	**9.97**	**9.40**	**10.16**	**10.14**	**9.90**	**24.13**	**27.85**	**28.35**	**27.77**
ltmSANN vs. best (+%)	34.5	26.4	26.6	17.8	67.8	45.1	34.1	27.0	113.8	115.3	114.1	111.9

Table 4.10: Performance of models with optimal settings using MAF at N when N varies from 10 to 40 on the training sets. The scores for $N = 50$ are in Table 4.9. The last row displays the improvement of the ltmSANN model over the best baseline (here NN, underlined).



(a) TED (b) Vimeo (c) Flickr

Figure 4.5: Comparison of models in terms of average MAF at N, for $1 \le N \le 50$, using cross-validation on the training set.

additional values of MAF at N for N lower than 50, namely 10, 20, 30 and 40, while Figure 4.5 plots all the values of MAF at N, for N from 1 to 50. It appears that the differences between the proposed models (ltmSANN, sigSANN) and each of the other ones remain constant when N varies, or even increase for smaller values of N, especially below 20. These values may even be considered as more important for a top-N recommender system than larger ones, because a user can find more quickly a relevant entry in a short recommendation list.

Among the low-rank factorization models, SVD performed similarly to the standard NMF and both of them performed best with low values of l. However, NMF was consistently better than SVD in all cases. The lowest scores for SVD are the ones obtained on the Flickr dataset, on which SVD was outperformed by all other methods except the TopPopular baseline. SNMF, on the other hand, was the best performing model among the low-rank factorization ones and it also performed better than NN models on the TED dataset, which confirms the validity of the sparsity assumption in this data, especially when the SNMF scores are compared to those of SVD and NMF. One reason for the lower scores of SNMF models compared to NN over Vimeo and Flickr might be that these datasets have 40% more items (about 800 more), and 30% and 50% respectively more users (about 2,000 and 5,000 more) than the TED dataset, resulting in

	TED (Sparse held-out)			**Vimeo (Sparse held-out)**			**Flickr (Sparse held-out)**		
Methods	**MAP**	**MAR**	**MAF**	**MAP**	**MAR**	**MAF**	**MAP**	**MAR**	**MAF**
TopPopular	3.10	13.42	5.04	2.12	9.34	3.46	1.32	6.37	2.19
SVD	4.38	16.48	6.92	2.91	11.06	4.60	3.64	15.03	5.86
NMF	4.34	16.46	6.87	2.82	10.80	4.48	3.48	13.38	5.53
SNMF	4.70	18.19	7.47	2.93	11.09	4.64	3.94	14.31	6.18
NN	5.10	19.32	8.07	4.11	15.67	6.51	6.79	24.83	10.67
sigSANN	5.77	21.45	9.09	4.83	15.19	7.33	13.79	50.54	21.67
ltmSANN	**6.10**	**22.73**	**9.63**	**5.48**	**16.92**	**8.28**	**15.26**	**56.75**	**24.05**
ltmSANN vs. best (+%)	19.6	17.6	19.3	33.3	7.9	27.1	124.7	124.5	125.3

Table 4.11: Performance of models with optimal settings using MAP, MAR and MAF at 50 on the *sparse held-out sets*. The last row displays the improvement of the ltmSANN model over the best baseline (NN, which is underlined).

much sparser user-item matrices despite the similar number of ratings. Such cases appear to be more difficult to model with latent factors than with local models such as NN.

4.7.2 Results over the Held-Out Sets

In Tables 4.11 and 4.12 we report results on the sparse and dense held-out sets respectively. Similarly to the results on the training sets with 5-fold cross-validation, the sentiment-aware models outperformed all the other ones. The ltmSANN model was the best performing one, with 19% improvement for TED, 27% for Vimeo and 125% for Flickr on the sparse held-out sets, and even higher improvements on the dense held-out sets: 43%, 180% and 106% respectively. On Flickr, which is the densest among the three datasets, the improvements of ltmSANN with respect to the other models were higher than on TED or Vimeo in all cases (i.e. the scores were more than doubled), namely cross-validation over the training set and testing over the two held-out sets. These results indicate that the denser a dataset with respect to user comments, the better the performance of the sentiment-aware models.[15]

In Figures 4.6 and 4.7 we display the performance of the models on the sparse and dense held-out sets by plotting the average precision (AP) against the average recall (AR) at N, varying N from 1 to 50. The SANN models have better performance compared to the baselines over all values of N, except for the largest values on the sparse Vimeo dataset (Fig. 4.6 (b)). Similarly to the observations in Figure 4.5, here the sentiment-aware models outperform the baselines by a larger margin for the smaller values of N (typically 1 to 20). Moreover, for Flickr, which has the highest density of comments, the difference is large over the entire range of N. The sentiment-aware models (fixed and adaptive ones) consistently outperform the other models (NMF, SNMF and NN) on the six sets in Figures 4.6 and 4.7. These results strongly indicate that sentiment information extracted from user comments is predictive for one-class CF, and

[15] One relative exception was the fact that the ltmSANN model had a smaller relative improvement (with respect to the baseline models) on the dense held-out set of Flickr than on the sparse one (106% vs. 125%). Still, its absolute MAF improvement on the sparse held-out set was 14% (10.67 for NN vs. 24.05 for ltmSANN) and in the dense held-out set it was larger, at 18% (16.53 for NN vs. 34.11 for ltmSANN).

	TED (Dense held-out)			**Vimeo (Dense held-out)**			**Flickr (Dense held-out)**		
Methods	**MAP**	**MAR**	**MAF**	**MAP**	**MAR**	**MAF**	**MAP**	**MAR**	**MAF**
TopPopular	3.42	12.64	5.39	2.24	7.41	3.45	2.13	6.82	3.25
SVD	5.03	15.80	7.63	3.32	9.39	4.91	3.93	17.49	10.03
NMF	4.78	15.17	7.27	3.20	9.16	4.74	6.70	15.42	9.34
SNMF	5.25	17.63	8.09	4.17	11.04	6.05	7.82	17.08	10.72
NN	5.65	17.97	8.60	5.33	15.31	7.91	12.10	26.07	16.53
sigSANN	7.45	23.94	11.37	8.58	22.54	12.44	22.99	53.07	32.09
ltmSANN	**8.05**	**26.43**	**12.35**	**9.92**	**25.71**	**14.32**	**24.42**	**56.56**	**34.11**
ltmSANN vs. best (+%)	42.4	47.0	43.6	186.1	167.9	181.0	101.8	116.9	106.3

Table 4.12: Performance of models with optimal settings using MAP, MAR and MAF at 50 on the *dense held-out sets*. The last row displays the improvement of the ltmSANN model over the best baseline (NN, which is underlined).

that adapting the sentiment scores to the user ratings further improves performance.

4.8 Analysis of the Results

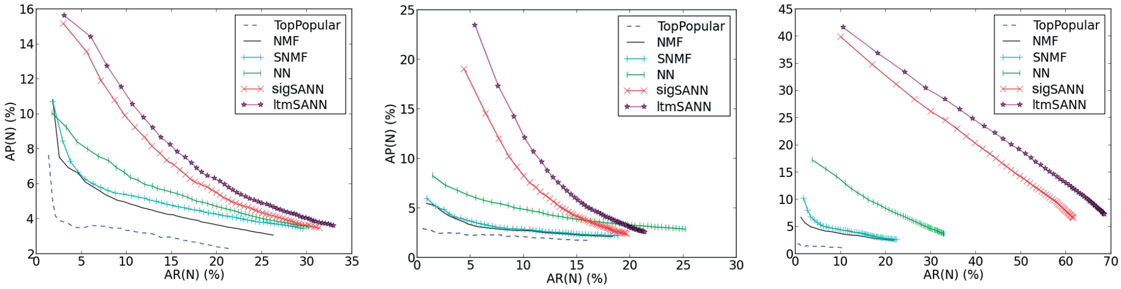
In the previous section we demonstrated the effectiveness of the proposed sentiment-aware neighborhood models, first by using cross-validation over the training sets and then by testing on held-out sets. In this section, we quantify the respective impacts of sentiment analysis, negative class assumptions, sentiment mapping functions, and quantity of comments on the performance of recommendation.

4.8.1 Importance of Sentiment Analysis

To assess the impact of sentiment analysis, we compare the recommendation results when using a random classifier (randSANN) with those obtained using our state-of-the-art rule-based one (sigSANN). These results, shown in Table 4.9, show that the sigSANN model outperforms randSANN over all datasets, with about 30% MAF improvement on TED, 53% on Vimeo and 112% on Flickr. The performance of randSANN is similar to the performance of the neighborhood model (NN) under the AMAN assumption. This means that when the quality of the sentiment classifier is poor, the additional information that is enclosed in user comments cannot be reliably exploited and it is the actual ratings that predict the user preference. All other things being equal, a more accurate sentiment classifier than the RB one could achieve further improvements, as suggested in a recent study (Sun et al., 2014).

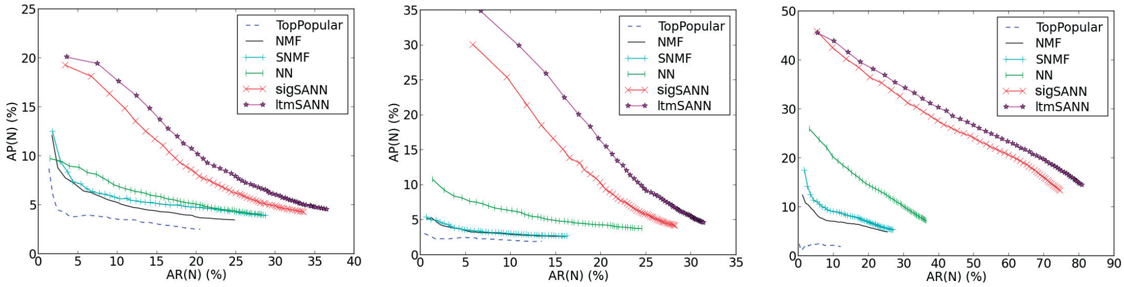
4.8.2 Independence from Negative Class Assumptions

We also studied whether the additional information captured from comments is always predictive for one-class CF regardless of the negative class assumption. We observe from Table 4.9 that under all assumptions the SANN model outperforms the neighborhood model (NN) significantly on all the three datasets using cross-validation. The greatest improvements of



(a) TED (b) Vimeo (c) Flickr

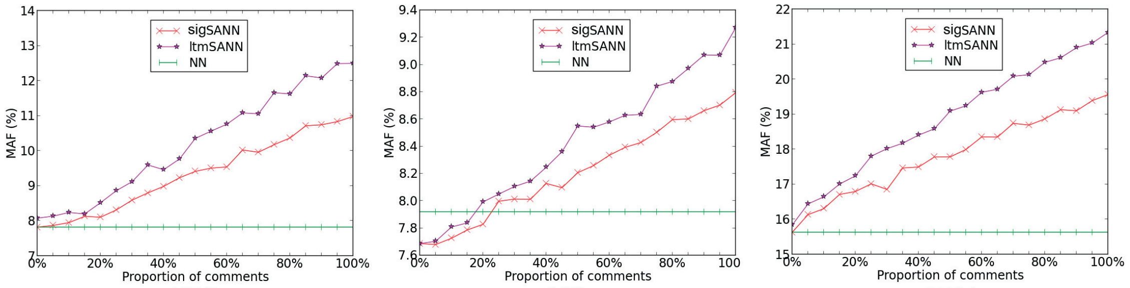
Figure 4.6: Model comparison in terms of average recall at N (horizontally) and average precision at N (vertically), for N from 1 to 50, on the *sparse held-out sets*. Data points with lower values of N have higher precision and lower recall. The top-scoring curves, upper-right, correspond to the ltmSANN model.



(a) TED (b) Vimeo (c) Flickr

Figure 4.7: Model comparison in terms of average recall at N (horizontally) and average precision at N (vertically), for N from 1 to 50, on the *dense held-out sets*. Data points with lower values of N have higher precision and lower recall. The top-scoring curves, upper-right, correspond to the ltmSANN model.

the sigSANN model are: 9% on TED under the EMAN assumption, 10% on Vimeo also with EMAN, and 95% on Flickr under the AMAN assumption. The best performing assumption for NN on TED and Vimeo was EMAN, and for Flickr it was AMAN. The same assumptions were the best performing ones for sigSANN over the three datasets, though for Vimeo there was no significant difference between EMAN and AMAN. Furthermore, the performance ordering of the different assumptions for NN and sigSANN is the same in most cases (combination of dataset and assumption in Table 4.9). From our experimental results, we conclude that the performances of the different assumptions depend mostly on the dataset, and then on the model. Overall, the additional information captured by SANN is valuable independently of the negative class assumption.



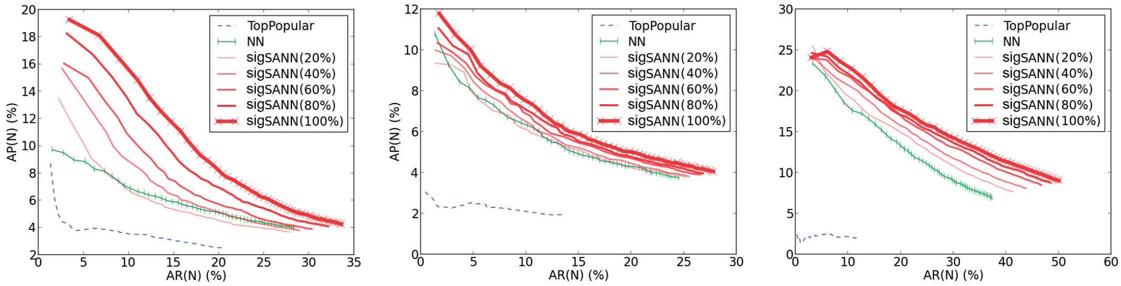
(a) TED (b) Vimeo (c) Flickr

Figure 4.8: Performance of baseline (NN) and sentiment-aware neighborhood models (sigSANN and ltmSANN) under the EMAN assumption, when varying the proportion of training comments, measured by MAF at 50 on the *dense held-out sets*. The performance of the proposed models increases with the number of comments.

4.8.3 Learning to Map Sentiment Scores to Ratings

Another question is: is it better to adapt our sentiment analysis scores to preference scores? To answer the question, we compare learned mappings for discrete sentiment values (ltmSANN) and for continuous ones (ltmpolSANN), learning either a global or a user-specific mapping, against fixed mappings (see scores in Table 4.9). Both ltmSANN models (per user and global) performed similarly with respect to each other but significantly better than the non-adaptive sigSANN model (6% improvement on TED, 10% on Vimeo and 8% on Flickr using 5-fold c.-v.). The user-specific mapping performs slightly better on the TED dataset, while the global mapping is optimal on Vimeo and Flickr. The reason is likely that in the Vimeo and Flickr communities, users have the tendency to follow shared textual norms to express their preferences through their texts, while in the TED community, the users have the tendency to follow more individual norms.

The ltmpolSANN method achieved the highest scores on TED, but it scored below ltmSANN over Vimeo and Flickr. Still, it always performed similarly to or better than polSANN. When considering the sentiments of each user individually, for long elaborate comments like in TED, it is more reliable to treat two comments of the same sentiment type differently as in ltmpolSANN(user), while for short comments like in Vimeo and Flickr it is better to treat them equally as in ltmSANN(user). When considering the sentiments of the users globally, the best option is to treat them equally for all types of sentiment, that is to use ltmSANN(global) instead of ltmpolSANN(global). The fixed mapping with normalization, polSANN, achieved only marginally lower performance compared to ltmSANN and ltmpolSANN(user). This an interesting result given that polSANN has a fixed and simple mapping. However, despite its complexity, ltmSANN is more flexible and can be applied to other datasets or predictors.



(a) TED (b) Vimeo (c) Flickr

Figure 4.9: Recall at N (horizontally) and precision at N (vertically) for N from 1 to 50 for two baselines (TopPopular and NN) and a sentiment-aware neighborhood model (sigSANN) under the EMAN assumption, on the *dense held-out sets*, when varying the proportion of comments that are used for training from 20% to 100% (color-coded as indicated). Lower values of N correspond to lower recall and higher precision in each curve.

4.8.4 Necessary Quantity of Comments

We now examine the quantity of comments that is necessary, when using the sentiment-aware models, to improve performance over the baselines. Figure 4.8 plots the MAP at 50 score of sigSANN and ltmSANN models under the EMAN assumption (the best one, see 4.8.2) when the proportion of comments varies from 0% to 100% of the total of available comments, on the dense held-out sets. Both models increase their performance as the number of comments increases and they outperform the NN baseline (under the same assumption, EMAN) already when only 5% of comments for TED and Flickr are used, or 20% for Vimeo. Similar results are obtained for the sparse held-out sets, except that the proportion of comments needed to outperform the NN baseline is slightly higher.

Using the same variation of the proportion of comments over the dense held-out sets, we plot in Figure 4.9 the precision and recall curves at N (for N from 1 to 50) of the sigSANN model compared to the NN baseline (under the EMAN assumption as well). On the TED dataset, the improvement of the sigSANN model is much higher for every additional fraction of comments than on Vimeo and Flickr, and the smallest improvements are on the Vimeo dataset. EMAN was the best assumption for the sigSANN model on TED, while for Vimeo and Flickr it was not optimal (see Tables 4.9 and 4.12). Hence, in the latter case, the improvement of sigSANN model over the baseline is in reality even higher than the one displayed in Figures 4.8 and 4.9.

4.8.5 Comparison with Other Models over the TED Dataset

We provide a brief comparison of the proposed fixed and learned mappings, SANN and ltmSANN, with the recommendation models proposed in a recent study (Sun et al., 2014) that also makes use of the TED dataset as we have distributed it (Section 4.2.1). Using 5-fold cross validation, in terms of average precision at 5 and 10, our SANN model outperforms the best sentiment-aware neighborhood models presented by Sun et al. by respectively 15% and 20%

Method	AP@5	AP@10
SA_UCF (Sun et al., 2014)	8.37	6.05
SA_ICF (Sun et al., 2014)	8.15	6.23
SANN	**9.93**	**7.79**
SA_AWAN_MF (Sun et al., 2014)	6.73	5.69
SA_wAWAN_MF (Sun et al., 2014)	10.07	7.98
ltmSANN	**10.60**	**8.23**
Fused model (Sun et al., 2014)	11.42	9.32

Table 4.13: Comparison of our SANN (fixed mapping) and ltmSANN (learned mapping) models with a recent study (Sun et al., 2014) on the TED dataset using 5-fold cross-validation (80% for training and 20% for testing). The scores of Sun at al. have been copied verbatim in the table, with the best one for each type (NN or matrix factorization) being underlined.

(top part of Table 4.13). Similarly, our ltmSANN model outperforms the best sophisticated models from Sun et al. (2014) based on matrix factorization by respectively 5% and 3% in terms of AP at 5 and 10 (middle part of Table 4.13). However, SANN and ltmSANN are outperformed by the combined model proposed by Sun et al., which exploits both frameworks, namely neighborhood models and matrix factorization (last line of the table). It is thus a topic for future research to explore the combination of our models as well, while avoiding over-fitting the TED dataset, but seeking progress on TED, Vimeo, Flickr and possibly other datasets at the same time.

4.8.6 Synthesis on the Influence of the Datasets on the Results

From the description of the datasets in Section 4.2 and the experimental results reported in Sections 4.7 and 4.8 we can infer the following relationships between the properties of the datasets and the performances of the methods:

- When the number of items increases (TED → Vimeo → Flickr), the performances of NN and SANN models increase, while the performances of latent factor models (SVD, NMF and SNMF) decrease, likely because NN models can cope with rating sparsity more effectively than the latent factor ones.
- When the density of comments increases, either from one dataset to another (Vimeo → TED → Flickr) or within each dataset (sparse vs. dense sets as in Section 4.8.4), the scores of SANN models increase, because there are more comments from which to extract sentiment information.
- When the correlation between comments and ratings increases (TED → Vimeo → Flickr), the percentages of improvement of SANN models compared to the best baseline increase, because SANN models are able to map appropriately the sentiment of comments to ratings and thus they benefit the most from the correlation between the two properties.

- When the test sets are dense, all the methods (NN, SANN and latent factor models) perform better than on sparse test sets because they contain a larger number of active users with many ratings in their profiles, hence more complete profiles than sparse sets.

4.9 Conclusion

In this chapter, we proposed sentiment-aware models to improve one-class CF. The models were evaluated on three real-world multimedia datasets, namely TED talks, Vimeo videos and Flickr images, demonstrating significant improvements over models that do not use sentiment information. For instance, on the Vimeo and Flickr test sets that have the most comments, scores are doubled with respect to not using the sentiment of comments. In addition, it was shown that the improvements of sentiment-aware models hold for all three negative class assumptions, meaning that the benefits gained from various strategies for balancing the negative class are likely to be preserved when combined with our model.

The results of extensive empirical studies showed that the adaptive sentiment-aware models (ltmSANN or ltmpolSANN) performed better than those with a fixed mapping (SANN or polSANN). This is likely because ltmSANN is able to adapt the sentiment scores to the user preferences, and in particular to model cases in which the output scores of the sentiment classifier do not exactly match actual preferences. This procedure can be considered as rating inference, although since we deal only with positive values (i.e. 1), the ratings that are inferred correspond to importance weights rather than commonly-used ratings (e.g. on a 1 to 5 scale). Still, these weights allow us to rank items for each user and to successfully recommend the top-N items in the list.

The proposed models are relevant to many real-world applications to communities where users interact both in terms of explicit feedback (favorites, likes) and in terms of textual feedback (comments, discussions). In datasets with a small amount of comments, the improvements brought by our models are likely to be less noticeable, although the improvements for individual users who comment frequently will still be noticeable. We have shown experimentally that our models perform well with three different types of content: lectures, general-purpose videos, and images. However, our models are not constrained by a domain, and can adapt to domain data through learning, so they are likely to perform well in domains with similar type of feedback, including traditional product recommendation, although their exact performance remains to be assessed experimentally in each case.

5 Weighted Multiple Instance Learning for Text Regression Tasks

Sentiment information from user comments can improve the modeling of user preferences, as we showed in the previous chapter. Although this approach enables systems to suggest new content that is relevant to users, there are also other factors which affect the quality of the recommendations. For instance, the lack of explanations of the recommendations that are made compromises the system's trustworthiness and transparency to users. Current methods for explaining recommendations are uninformed about why the users have particular preferences about items and their particular aspects. Such information can be found in reviews of products which rate them globally or according to specific aspects. Hence, recommender systems would benefit from the capability to understand the user preferences or opinions about items and how they are expressed in such text. The majority of methods for modeling aspect sentiment from text have focused on feature engineering or learning, assuming a supervised learning objective. However, the data often comprises labels assigned by users at the document (review) level only, which makes it hard to determine to which specific part of the text they refer. In this chapter, we cast the above sentiment modeling problem as a weakly supervised one. To solve this problem, we propose a weighted multiple-instance learning model applied to text regression tasks, namely the prediction of aspect, sentiment and emotion ratings from user-contributed texts such as product reviews and free-form comments.

5.1 Introduction

Sentiment analysis of texts provides a coarse-grained view of their overall attitude towards an item, either positive or negative. However, the overall sentiment of a text towards an item often results from the ratings of several specific aspects of the item. For instance, the author of a review might have a rather positive sentiment about a movie, having particularly liked the plot and the music, but not too much the actors. Determining the ratings of each aspect automatically is a challenging task, which may seem to require the engineering of a large number of features designed to capture each aspect. Here, we propose a new feature-agnostic solution for analyzing aspect-related ratings expressed in a text, thus aiming for a finer-grained, deeper analysis of text meaning than overall sentiment analysis.

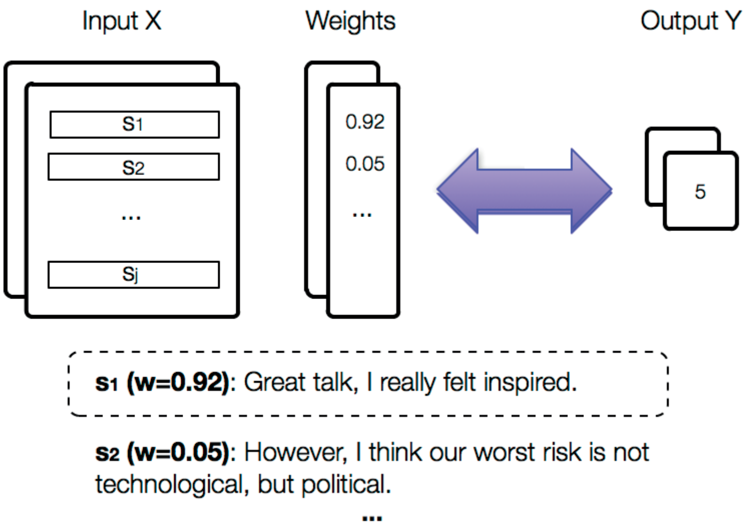

Input X
Weights
Output Y
S1
S2
...
Sj
0.92
0.05
...
5
s1 (w=0.92): Great talk, I really felt inspired.
s2 (w=0.05): However, I think our worst risk is not technological, but political.
...

Figure 5.1: Analysis of a comment (bag of sentences $\{s_1,...,s_j\}$) annotated by humans with the maximal positive sentiment score (5 stars). The weights assigned by MIR reveal that s_1 has the greatest relevance to the overall sentiment.

Current state-of-the-art methods to sentiment analysis including those dedicated to aspect-based sentiment analysis, attempt to go beyond word-level features, either by using higher-level linguistic features such as POS tags, parse trees, and semantic knowledge, or by learning features that capture syntactic and semantic dependencies between words. Once an appropriate feature space is found, the ratings are typically modeled using a linear model, such as Support Vector Regression (SVR) with ℓ_2 norm for regularization or Lasso Regression with ℓ_1 norm. By treating a text globally, these models ignore that the sentences of a text have diverse contributions to the overall sentiment or to the attitude towards a specific aspect of an item.

In this chapter, we propose a new learning model which answers the following question: “To what extent does each part of a text contribute to the prediction of its overall sentiment or the rating of a particular aspect?” The model uses multiple-instance regression (MIR), based on the assumption that not all the parts of a text have the same contribution to the prediction of the rating. Specifically, a text is seen as a bag of sentences (instances), each of them modeled as a word vector. The overall challenge is to learn which sentences refer to a given aspect, and how they contribute to the text’s attitude towards it, but the model applies to overall sentiment analysis as well. For instance, Figure 5.1 displays a positive global comment on a TED talk and the weights assigned to two of its sentences by MIR.

Using regularized least squares, we formulate an optimization objective to jointly assign instance weights and regression hyperplane weights. Then, by estimating instance relevance, we predict global or aspect ratings in previously unseen texts. The parameters of the model are learned using an alternating optimization procedure inspired by Wagstaff and Lane (2007). Our model requires only text with ratings for training, with no particular assumption on the word features to be extracted, and provides interpretable explanations of the predicted ratings through the relevance weights assigned to sentences. We also show that the model has reasonable computational demands.

The model is evaluated on aspect, sentiment and emotion rating prediction over seven datasets: five of them contain reviews with aspect labels about beers, audiobooks and toys (McAuley et al., 2012), and two contain TED talks with emotion labels, and comments on them with sentiment labels (Pappas and Popescu-Belis, 2013b). Our model outperforms previous MIR models and two strong linear models for rating prediction, namely SVR and Lasso, by more than 10% relative in terms of mean squared error. The improvement is observed even when the sophistication of the feature space increases, e.g. from BOW with counts to BOW with TF-IDF.

The rest of the chapter is organized as follows. Section 5.2 introduces our framework and formulates the problem, while Section 5.3 describes within this framework the main assumptions on the relations between instances and bags made in past studies. Section 5.4 presents our MIR model and learning procedure. Section 5.5 presents the datasets and evaluation methods. Section 5.7 reports our results on three text regression tasks, namely aspect, sentiment and emotion rating prediction, and provides examples of rating explanation, while Section 5.9 we apply the findings of this study in two real-world scenarios.

5.2 Problem Definition: Weakly Supervised Text Regression

Let us consider a set B of m bags accompanied by numerical labels Y as input data D. Therefore, D is noted as $D = \{(\{b_{1j}\}^d_{n_1}, y_1), ..., (\{b_{mj}\}^d_{n_m}, y_m)\}$, with $b_{ij} \in \mathbb{R}^d$ (for $1 \leq j \leq n_i$) and $y_i \in \mathbb{R}$. Each bag B_i consists of n_i data points (called ‘instances’), which are typically d-dimensional vectors, e.g. vectors of words. The challenge is to infer the label of new bags which may have a variable number of instances n_i. This requires finding a set of bag representations $X = \{x_1, \ldots, x_m\}$ of size m where $x_i \in \mathbb{R}^d$, to train our regression model, from which the class labels of new bags can then be inferred. The goal is then to find a mapping from this representation, noted $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}$, which is able to predict the label of a given bag. Ideally, assuming that X is the optimal bag representation for our task[1], we look for the optimal regression hyperplane Φ which minimizes a loss function $\mathscr{L}$ plus a regularization term Ω as follows:

$$\Phi = \underset{\Phi}{arg\,min} \left(\underbrace{\mathscr{L}(Y, X, \Phi)}_{\text{loss}} + \underbrace{\Omega(\Phi)}_{\text{reg.}} \right) \tag{5.1}$$

Since the best set of representations X for a task is generally unknown, one has to make assumptions to define it or compute it jointly with the regression hyperplane Φ. Thus, the main difficulty lies in finding a good assumption for X, as we will now discuss.

[1] Note that X is a function of B.

5.3 Previous Multiple Instance Assumptions

The assumption that is made about the input in multiple instance problems has a strong impact on the performance of a multiple instance learning algorithm and its appropriateness depends on the data at hand. We describe here three assumptions that have been frequently made in past MIR studies, to which we will later compare our model: namely, aggregating all instances, keeping them as separate examples, or choosing the most representative one (Wang et al., 2012). For each assumption, we will experiment with two state-of-the-art regression models (noted abstractly as f), namely SVR (Drucker et al., 1996) and Lasso (Tibshirani, 1996) with respectively the ℓ_2 and ℓ_1 norms for regularization.

5.3.1 Aggregated Instances

According to the *Aggregated* assumption, each bag is represented as a single d-dimensional vector, which is the average of its instances (hence $x_i \in \mathbb{R}^d$). Then, a regression model f is trained on pairs of vectors and class labels, $D_{agg} = \{(x_i, y_i) \mid i = 1, \ldots, m\}$, and the predicted class of an unlabeled bag $B'_i = \{b'_{ij} \mid j = 1, \ldots, n_i\}$ is computed as follows:

$$\hat{y}(B'_i) = f(mean(\{b'_{ij} \mid j = 1, \ldots, n_i\})) \tag{5.2}$$

In fact, a simple sum can also be used instead of the mean, and we observed in practice that with an appropriate regularization there is no difference on the prediction performance between these options. This baseline corresponds to the typical approach for text regression tasks, where each text sample is represented by a single vector in the feature space. For instance, bag-of-words (BOW) with counts or TF-IDF weights.

5.3.2 Instance as Example

According to the *Instance* assumption, each of the instances in a bag as separate examples, by labeling each of them with the bag's label. A regression model f is learned over the training set made of all vectors of all bags, $D_{ins} = \{(b_{ij}, y_i) \mid i = 1, \ldots, m; j = 1, \ldots, n_i\}$, assuming that there are m labeled bags. To label a new bag B'_i, given that there is no representation x_i, the method simply averages the predicted labels of its instances:

$$\hat{y}(B'_i) = mean(\{f(b'_{ij}) \mid j = 1, \ldots, n_i\}) \tag{5.3}$$

Instead of the average, the median value can also be used, which is more appropriate when the bags contain outlying instances.

5.3.3 Primary Instance

According to the *Prime* assumption, a single instance in each bag is responsible for its label (Ray and Page, 2001). This instance is called the primary or prime one. The method is similar to the previous one, except that only one instance per bag is used as training data: $D_{pri} = \{(b_i^p, y_i) \mid i = 1, \ldots, m\}$, where b_i^p is the prime instance of the i^{th} bag B_i and m is the number of bags. The prime instances are discovered through an iterative algorithm which refines the regression model f. Given an initial model f, in each iteration the algorithm selects from each bag a prime candidate which is the instance with the lowest prediction error. Then, a new model is trained over the selected prime candidates, until convergence. For a new bag, the target class is computed as in Equation 5.3 above.

The main drawback of *Prime* assumption is that is able to pinpoint the primary instance only when the bag is already labeled, thus it is not applicable on unlabeled bags. *Aggregated* and *Instance* assumptions are unable to pinpoint the most relevant instances with respect to the label of each bag. Therefore, all the above methods are applicable only to the prediction of labels of unseen bags. Instead of using a pre-defined method to aggregate instances, our model learns an optimal method directly from in-domain training data which allows more degrees of freedom in the regression model than previous ones.

5.3.4 Instance Relavance

According to the *Instance Relevance* assumption, a weighted combination of the instances is responsible for the label of each bag, as in (Wagstaff and Lane, 2007; Wagstaff et al., 2008; Wang et al., 2011). This assumption can be seen as a generalization of the *Aggregated* and *Prime* assumptions. For instance, when we set equal instance weights for each bag, the *Instance Relevance* assumption is equivalent to taking the simple average of the instances as in the *Aggregated* one, while when we set only one non-zero weight for each bag it is equivalent to the *Primary* assumption. The main drawbacks of the previous instance relevance methods that we intend to address (see Section 5.4) are the following ones: (i) prohibitive complexity for high dimensional feature spaces which are common in text regression tasks, as in Wagstaff and Lane (2007); Wagstaff et al. (2008); Wang et al. (2011); (ii) inefficiency or inability to estimate the importance of the instances of unseen bags, as in (Wang et al., 2011) and (Wagstaff and Lane, 2007) respectively; and (iii) lack of modeling of weights with sparsity which can be adapted to different types of data. Our proposal to overcome these issues is described in detail hereafter.

5.4 Proposed Model: Weighted Multiple Instance Regression

The new MIR model that we propose assigns individual relevance values (weights) for each instance of a bag to model the input structure, similarly to the assumption made by Wagstaff and Lane (2007), but unlike the three first simplifying assumptions. To jointly learn instance weights and target labels efficiently, we minimize a regularized least squares loss (RLS) objective instead of solving normal equations, which allows to support high-dimensional feature spaces, as required for text regression tasks. In addition, the model is able to predict both the class label and the contribution of each instance of the bag to the bag's label of previously unseen (hence unlabeled) bags. Our model learns an optimal method directly from in-domain training data to aggregate instances, rather than a pre-defined one, and allows more degrees of freedom in the regression model than previous ones. Lastly, the weight of an instance is interpreted as its relevance both in training and prediction.

5.4.1 Instance Relevance Assumption

Each bag defines a bounded region of a hyperplane orthogonal to the y-axis (the envelope of all its points). The goal is to find a regression hyperplane that passes through each bag B_i and to predict its label by using at least one data point x_i within that bounded region. Thus, the point x_i is a convex combination of the points in the bag, in other words B_i is represented by the weighted average of its instances b_{ij}:

$$x_i = \sum_{j=1}^{n_i} \psi_{ij} b_{ij}, \psi_{ij} \geq 0 \text{ and } \sum_{j=1}^{n_i} \psi_{ij} = 1 \tag{5.4}$$

where ψ_{ij} is the weight of the j^{th} instance of the i^{th} bag. Each weight ψ_{ij} indicates the saliency or relevance of an instance j to the prediction of the class y_i of the i^{th} bag. The constraint forces x_i to fall within the bounded region of the points in bag i and guarantees that the i^{th} bag will influence the regressor.

5.4.2 Modeling Bag Structure and Labels

We propose here a method for jointly learning to predict the instance weights and the target labels of bags over a given dataset. Let us consider a set of m bags, where each bag B_i is represented by its n_i d-dimensional instances, i.e. $B_i = \{b_{ij}\}^d_{n_i}$, along with the set of k target class labels for each bag, $Y = \{y_i\}^k_m, y_i \in \mathbb{R}$. The set of representation of all B_i in the (same) feature space $X = \{x_1, \ldots, x_m\}$ with $x_i \in \mathbb{R}^d$, is obtained using the n_i instance weights associated to each bag B_i, $\psi_i = \{\psi_{ij}\}^1_{n_i}$ with $\psi_{ij} \in [0, 1]$, which are initially unknown. Thus, we look for a linear regression model f that is able to model each target value y_i for each bag i using the regression coefficients $\Phi \in \mathbb{R}^d$, that is, $Y = f(X) = \Phi^T X$, where X and Y are respectively the sets of training bags and their labels[2].

We define a loss function according to the least squares objective dependent on B, Y, Φ and

[2] Note that the coefficients Φ, the weights Ψ and representations X are different for each aspect. For brevity, we develop the formulation assuming only one aspect ($k = 1$).

the set of weight vectors $\Psi = \{\psi_1, \dots, \psi_m\}$ using Eq. 5.4 as follows:

$$\begin{aligned}\mathscr{L}(Y, B, \Psi, \Phi) &= ||Y - \Phi^T X||_2^2 \\ &\overset{(5.4)}{=} \sum_{i=1}^{N} \Big(y_i - \Phi^T \big(\sum_{j=1}^{n_i} \psi_{ij} b_{ij}\big)\Big)^2 \\ &= \sum_{i=1}^{N} \Big(y_i - \Phi^T (B_i \psi_i)\Big)^2 \end{aligned} \tag{5.5}$$

Using the above loss function, accounting for the constraints of our assumption in Eq. 5.4 and assuming ℓ_2-norm for regularization with ϵ_1 and ϵ_2 terms for each $\psi_i \in \Psi$ and Φ respectively, we obtain the following least squares objective from Eq. 5.1. The selection of the ℓ_2-norm was based on preliminary results which demonstrated that ℓ_2-norm has superior performance than ℓ_1-norm, i.e. it learned a more accurate function. Other combinations of p-norm regularizations can be explored for f_1 and f_2, e.g. to control the sparsity of instance weights and regression coefficients (amount of non-zero elements).

$$\begin{aligned}\psi_1, \dots, \psi_m, \Phi = \underset{\psi_1,\dots,\psi_m,\Phi}{arg\ min} \sum_{i=1}^{m} \underbrace{\Big(\underbrace{\Big(y_i - \Phi^T (B_i \psi_i)\Big)^2}_{f_1 \text{ loss}} + \underbrace{\epsilon_1 ||\psi_i||}_{f_1 \text{ reg.}}\Big)}_{f_2 \text{ loss}} + \underbrace{\epsilon_2 ||\Phi||^2}_{f_2 \text{ reg.}} \\ \text{subject to:} \quad \psi_{ij} \geq 0 \ \forall i, j \text{ and } \sum_{j=1}^{n_i} \psi_{ij} = 1 \ \forall i \end{aligned} \tag{5.6}$$

The above objective is non-convex and difficult to optimize because the minimization is with respect to all $\psi_1, \dots, \psi_m$ and Φ at the same time. As indicated in Eq. 5.6 above, we will note f_1 a model that is learned from the minimization only with respect to $\psi_1, \dots, \psi_m$ and f_2 a model obtained from the minimization with respect to Φ only. In Eq. 5.6, we observe that if one of the two functions is known or held fixed, then the other one is convex and can be learned with the well-known least squares solving techniques. In Section 5.4.3, we will describe an algorithm that exploits this observation.

Having computed $\psi_1, \dots, \psi_m$ and Φ, we can predict a label for an unlabeled bag using Eq. 5.3, but then we would not be able to compute the weights of its instances. Moreover, information that has been learned about the instances during the training phase would not be used during prediction. One way to be able to predict instance weights is to learn a set of regression coefficients $O \in \mathbb{R}^d$ which are able to map a given instance to a real-valued weighted in [0,1] interval which also satisfies the constraints of Eq. 5.4. To achieve this we have to replace ψ in Eq 5.6 with a linear regression model $f_3(B_i) = O^T B_i$; thus, creating a least squares objective based on a bilinear model with m non-negativity and unit constraints, as follows:

$$O, \Phi = \underset{O,\Phi}{arg\ min} \sum_{i=1}^{m} \Big(\Big(y_i - \Phi^T (B_i O^T B_i)\Big)^2 + \epsilon_1 ||O^T B_i||\Big) + \epsilon_2 ||\Phi||^2 \tag{5.7}$$

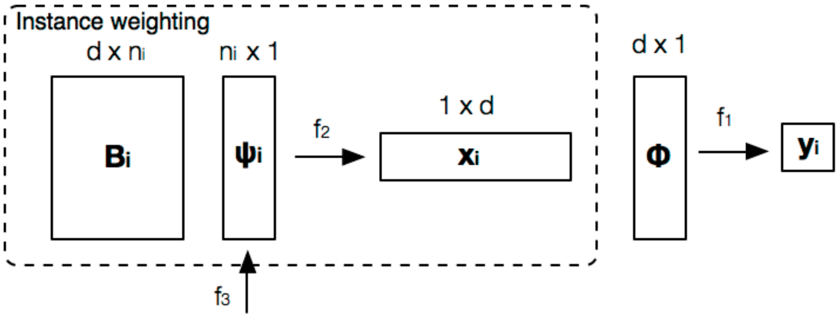


Figure 5.2: Visual representation for the training and testing procedure of Algorithm 2. The training of f_1 and f_2 is done with the alternating projections method (AP), and the f_3 model is trained on the weights that are obtained from the previous step. For testing, the weights of an unseen bag B'_i are estimated using the f_3 model, and the class label is predicted using the f_1 model with the estimated bag representation x_i.

subject to: $(O^T B_i)_{ij} \geq 0 \; \forall i,j$ and $\sum_{j=1}^{n_i} (O^T B_i)_{ij} = 1 \; \forall i$. We find this objective is intractable for learning in practice due to the numerous non-negativity and unit constraints, and therefore one has to make simplifications to make it tractable. For instance, by removing the unit constraints or by forcing f_3 output to be positive with unit length; however, this would mean that we deviate from our goal to have control over the sparsity of the instance weights.

For these reasons, we solve the problem sequentially instead of jointly by introducing a third regression model f_3, with regression coefficients $O \in \mathbb{R}^d$ assuming a ℓ_2-norm for the regularization with ϵ_3 term, which is trained on the relevance weights obtained from the Eq. 5.6, $D_w = \{(b_{ij}, \psi_{ij}) \mid i = 1,...,m; j = 1,...,n_i\}$. The optimization objective for the f_3 model is the following:

$$O = \underset{O}{arg\ min} \underbrace{\sum_{i=1}^{N} \sum_{j=1}^{n_i} \left(\psi_{ij} - O^T b_{ij}\right)^2}_{f_3 \text{ loss function}} + \underbrace{\epsilon_3 ||O||^2}_{f_3 \text{ reg.}} \tag{5.8}$$

Again, this minimization can be easily performed with the well-known least squares solving techniques. The learned model is able to estimate the weights of the instances of an unlabeled bag B'_i during prediction time as follows:

$$\hat{\psi}_i = f_3(B'_i) = O^T B'_i \tag{5.9}$$

The $\hat{\psi}_i$ weights are estimations which are influenced by the relevance weights learned in our minimization objective of Eq. 5.6 but their range of values is not constrained at prediction time. To obtain interpretable weights, we normalize the estimated scores to the $[0,1]$ interval as follows: $\hat{\psi}_i = \hat{\psi}_i / \sum(\hat{\psi}_i)$. Finally, the prediction of the label for the unsen bag B'_i using the estimated instance weights $\hat{\psi}_i$ is done as follows:

$$\hat{y}_i = f_2(B'_i) = \Phi^T B'_i \hat{\psi}_i \tag{5.10}$$

5.4.3 Learning with Alternating Projections

Algorithm 2 on page 93 solves the non-convex optimization problem of Eq. 5.6 by using a powerful class of methods for finding the intersection of convex sets, namely alternating projections (AP). The problem is firstly divided into two convex problems, namely f_1 loss function and f_2 loss function, which are then solved in an alternating fashion. Like EM algorithms, AP algorithms do not have general guarantees on their convergence rate, although, in practice, we found it acceptable, at generally fewer than 20 iterations. The convergence is achieved when the difference of the mean absolute error of the model between two consecutive iterations over the training set does not exceed a predefined threshold of 10^{-6}, noted as 'tolerance' in Algorithm 2.

Figure 5.2 displays a visual representation of the training and testing procedure of Algorithm 2. The algorithm takes as input the set of bags B_i with known labels y_i and the regularization terms $\epsilon_1, \epsilon_2, \epsilon_3$ and proceeds as follows. First, under a fixed regression model f_2, it proceeds with optimizing f_1, i.e. finding the optimal assignment of weights to the instances of each bag (can be seen as a projection of Φ vectors on the ψ_i space, which is a n_i-simplex), and computes the new representation set X of the bags. Second, given the fixed instance weights, the algorithm trains a new regression model f_2 using X (this can be seen as a projection back to the initial Φ space). This procedure repeats until convergence, i.e. when there is very small or zero decrease on the training error based on the predefined tolerance as mentioned above, or until a maximum number of iterations has been reached. Finally, the regression model f_3 is trained on the weights learned from the previous steps.

5.4.4 Complexity Analysis

We analyze the time complexity of the proposed algorithm in terms of the following input variables, noted $h = \{m, \hat{n}, d\}$, where m is the number of bags, $\hat{n}$ is the average size of the bags, and d is the dimension of the feature space (here, size of word vectors). The time complexity T_{AP} of the AP procedure in Algorithm 2 can be expressed as follows:

$$\begin{aligned} T_{AP}(h) &= \big(\mathcal{O}\big(m\big)\big(\mathcal{O}\big(\hat{n}\big)\mathcal{O}\big(f_1\big)\big)\big) + \mathcal{O}\big(f_2\big) \\ &= \big(\mathcal{O}\big(m\big)\big(\mathcal{O}\big(\hat{n}\big)\mathcal{O}\big(\hat{n}^2\big)\big)\big) + \mathcal{O}\big(md^2\big) \\ &= \mathcal{O}\big(m(\hat{n}^2 + d^2)\big) \end{aligned} \tag{5.11}$$

Thus, the overall time complexity T of Algorithm 2 in terms of the same input variables $h = \{m, \hat{n}, d\}$ is derived as follows:

$$\begin{aligned} T(h) &= T_{AP}(h) + T_{f_3}(h) \\ &= \mathcal{O}\big(m(\hat{n}^2 + d^2)\big) + \mathcal{O}\big(m\hat{n}d^2\big) \\ &= \mathcal{O}\big(m(\hat{n}^2 + d^2 + \hat{n}d^2)\big) \end{aligned} \tag{5.12}$$

ALGORITHM 2: Learning the parameters of the regularized least squares objective in Eq. 5.6 with alternating projections. The stopping criterion is based on the pre-defined error tolerance over MAE or MSE and a maximum number of iterations.

```
Data: Reviews $B = \{b_{ij}\}_{n_i}^{m}$, Ratings $Y = \{Y_i\}^m$
Result: Parameters $\psi_1, \ldots, \psi_N$, $\Phi$, $O$
set($max\_iter$, $tolerance$, $\epsilon_1$, $\epsilon_2$, $\epsilon_3$) #  Set max iterations, tolerance and hyper-parameters
initialize($iter$, $e_{prev}$) #  Initialize iteration counter and previous error
initialize($\psi_1, \ldots, \psi_N, \Phi$, $X$) #  Initialize model parameters
while not converged do
    for $B_i$ in $B$ do
        # Given $\Phi$ optimize for weights $\Psi_i$
        $P_i = \Phi^T B_i$ # Project to the instance space
        $\psi_i = cRLS(P_i, Y_i, \epsilon_1)$ # $f_1$ model
        $x_i = B_i \psi_i^T$ # Project back to the feature space
    end
    # Given $\Psi_1, \ldots, \Psi_m$ optimize for $\Phi$s
    $\Phi = RLS(X, Y, \epsilon_2)$ # $f_2$ model
    # Check for convergence based on tolerance and $max\_iter$
    $e = \frac{1}{m} \sum_i (Y_i - X_i^T \Phi)$ #  Mean absolute error
    if $e_{prev} - e < tolerance$ or $iter > max\_iter$ then
        $converged$ = True
    end
    $e_{prev} = e$
    $iter++$
end
# Given $\psi_1, \ldots, \psi_m$ optimize for $O$
$O = RLS(\{b_{ij} \forall i, j\}, \{\psi_{ij} \forall i, j\}, \epsilon_3)$ # $f_3$ model
```

where T_{AP} and T_{f_3} are respectively the time complexities of the AP procedure and of training the f_3 model. Eq. 5.12 shows that when $\hat{n} \ll m$, the model complexity is linear with the input bags m and always quadratic with the number of features d.

Lastly, we can see that T_{AP} is more efficient than a previous proposal (Wagstaff and Lane, 2007) that used normal equations for computing the f_1 and f_2 regression models:

$$T_{AP}(h) = \underbrace{\mathcal{O}\big(m(\hat{n}^2 + d^2)\big)}_{\text{with RLS}} \leq \underbrace{\mathcal{O}\big(m(\hat{n}^3 + d^3)\big)}_{\text{with normal equations}} \tag{5.13}$$

The time complexity of RLS is indeed quadratic with respect to d, while when using normal equations it is cubic, which makes computation prohibitive for large feature spaces such as those used for dealing with text data.

Previous works on relevance assignment for MIR have prohibitive complexity for high dimensional feature spaces or numerous bags and hence they are not most appropriate for text regression tasks. Wagstaff and Lane (2007) have cubic time complexity with the average bag

	Bags	**Instances**		**Dim.**	**Aspect ratings**
Dataset	**Type**	**Type**	**Count**	**Count**	**Classes**
BeerAdvocate	review	sentence	12,189	19,418	feel, look, smell, taste, overall
RateBeer (ES)			3,269	2,120	appearance, aroma, overall, palate, taste
RateBeer (FR)			4,472	903	appearance, aroma, overall, palate, taste
Audiobooks			4,886	3,971	performance, story, overall
Toys & Games			6,463	31,984	educational, durability, fun, overall
TED comments	comment	sentence	3,814	957	sentiment (polarity)
TED talks	comments per talk	comment	11,993	5,000	unconvincing, fascinating, persuasive, ingenious, longwinded, funny, inspiring, jaw-dropping, courageous, beautiful, confusing, obnoxious

Table 5.1: Description of the datasets used for aspect, sentiment and emotion rating prediction. 'Dim.' is the number of dimensions of each feature space.

size $\hat{n}$ and the number of features d; Zhou et al. (2009) use kernels, thus their complexity is quadratic with the number of bags m; and Wang et al. (2011) have cubic time with respect to d. Our formulation is thus competitive in terms of complexity and the benefits gained from our modeling do not come with excessive computational demands.

5.5 Description and Statistics of the Datasets

We describe here and provide statistics for the seven datasets, along with an in-depth analysis of the emotion attributes of the TED dataset.

5.5.1 Overview of Aspect, Sentiment and Emotion Data

We use seven datasets summarized in Table 5.1. Five publicly available datasets were built for aspect prediction by McAuley et al. (2012) – namely BeerAdvocate, Ratebeer (ES), RateBeer (FR), Audiobooks and Toys & Games – and have the aspect ratings that were assigned by their creators on the respective websites. On the set of comments on TED talks presented in Chapter 3, Section 3.2, which we gathered and distributed, we aim to predict two types of labels: talk-level emotion attributes assigned by the community users through voting[3], and comment polarity scores assigned by crowdsourcing as explained in Chapter 4, Section 4.3.

The distributions of ratings per dataset are shown in Figure 5.3 grouped in three categories, namely aspect rating datasets, sentiment datasets, and emotion datasets. Five datasets are in English, one in Spanish (Ratebeer) and one in French (RateBeer), so our results will also demonstrate the language-independence of our method.

From every dataset we keep 1,200 texts as bags of sentences, to ensure comparison across a similar number of bags. However, we also use three *full-size* datasets, namely Ratebeer ES (1,259 labeled reviews), Ratebeer FR (17,998) and Audiobooks (10,989). The features for

[3] The votes per talk express the aggregate emotion repsponses of the community users after they view a talk.

each of them are word vectors with binary attributes signaling word presence or absence, in a traditional bag-of-words model (BOW). The word vectors are provided with the first five datasets and we generate them for the latter two, after lowercasing and stopword removal. Moreover, for TED comments, we compute TF-IDF scores using the same dimensionality as with BOW, to experiment with a different feature space. The target class labels are normalized by the maximum rating in their scale, except for TED talks where the votes are normalized by the maximum number of votes over all the emotion classes for each talk, and two emotions, 'informative' and 'ok', are excluded as they are neutral ones.

5.6 Evaluation Protocol and Metrics

We compare the proposed model, noted *APWeights*, with following baselines: the *Aggregated*, *Instance* and *Prime* methods presented in Section 5.3, and the *Clustering* method, which is an instance relevance method proposed by Wagstaff et al. (2008)[4]. In addition, we report for comparison the scores of *AverageRating*, which always predicts the average rating over the training set. Lastly, we compare our proposal with state-of-the-art models on aspect rating prediction proposed by McAuley et al. (2012).

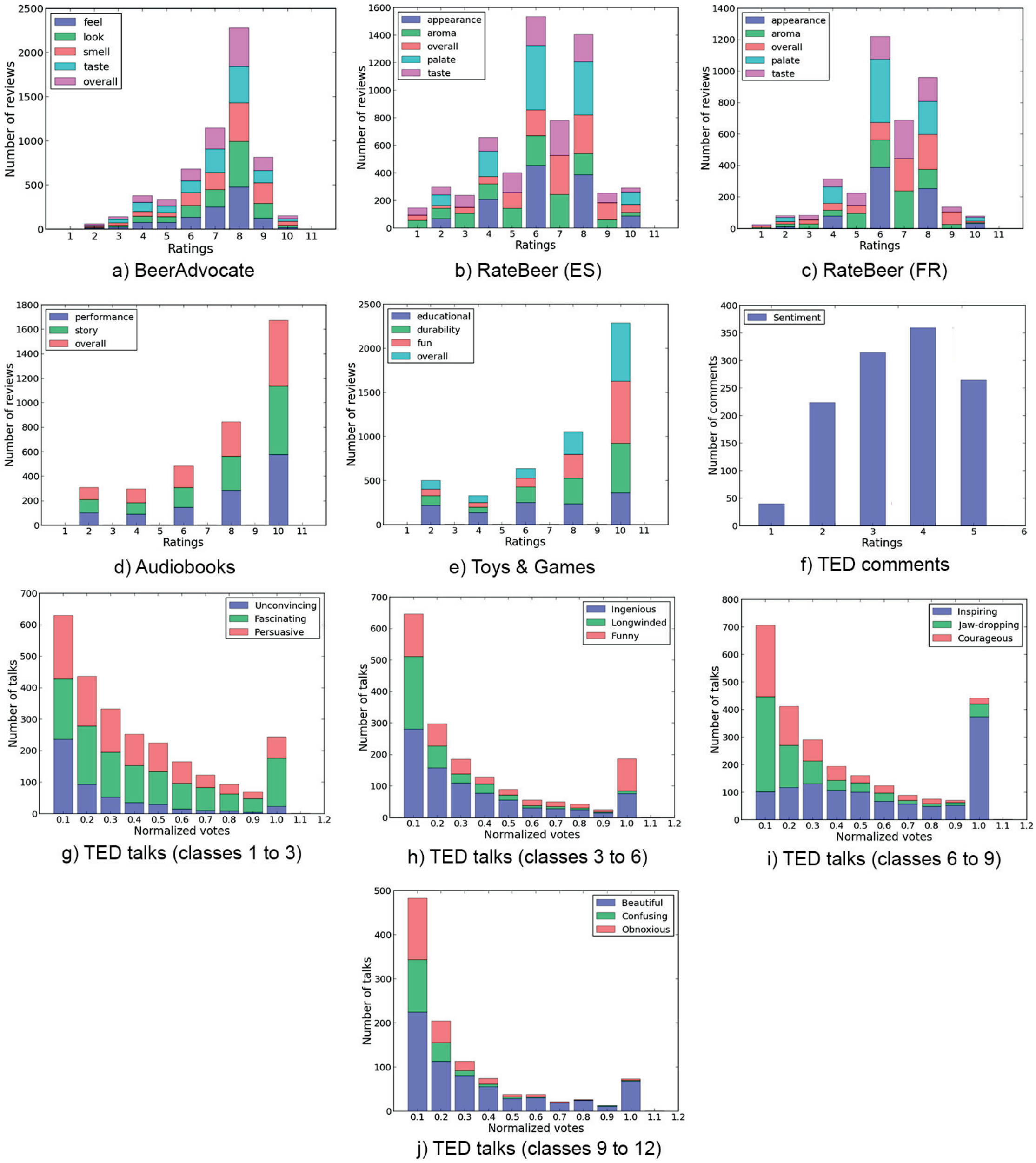
First, for each aspect class, we optimize all methods on a development set of 25% of the data, i.e. 300 randomly selected bags. Then, we perform 5-fold cross-validation for every aspect on each entire data set and report the average scores using the optimal hyper-parameters for each method. To compare with methods from McAuley et al. (2012), we replicate their experimental setup i.e. by splitting the dataset in half for training and testing.

We report standard error metrics for regression, namely the Mean Absolute Error (MAE) and the Mean Squared Error (MSE). The former measures the average magnitude of errors in a set of predictions while the latter measures the average of their squares. Both are defined over the test set of bags B'_i of size m' in Eq. 5.14. The cross-validation scores are obtained by averaging the MAE and MSE scores on each fold.

$$\text{MAE} = \frac{1}{m'}\sum_{i=1}^{m'} |f(B_i) - y_i| \text{ ; } \text{MSE} = \frac{1}{m'}\sum_{i=1}^{m'} (f(B_i) - y_i)^2 \tag{5.14}$$

To find the optimal hyper-parameters for each model, we perform 3-fold cross-validation on the development set using exhaustive grid-search over a range of possible values and select the ones that perform best in terms of MAE. The hyper-parameters to be optimized for the baselines (except for AverageRating) are the regularization terms λ_2, λ_1 of their regression model f, namely SVR which uses the ℓ_2 norm and Lasso which uses the ℓ_1 norm. As for APWeights, it relies on three regularization terms, namely $\epsilon_1, \epsilon_2, \epsilon_3$ of the ℓ_2-norm for the f_1, f_2 and f_3 regression models. Lastly, for the Clustering baseline, we use the f_2 regression model, which relies on ϵ_2 and the number of clusters k, optimized over the 5–50 range with step 5, for its clustering algorithm, here k-Means. All the regularization terms are optimized

[4] We use the implementation from https://github.com/garydoranjr/mcr/.



a) BeerAdvocate

b) RateBeer (ES)

c) RateBeer (FR)

d) Audiobooks

e) Toys & Games

f) TED comments

g) TED talks (classes 1 to 3)

h) TED talks (classes 3 to 6)

i) TED talks (classes 6 to 9)

j) TED talks (classes 9 to 12)

Figure 5.3: Distributions of rating values per rating class for seven datasets which are grouped in three categories: (1) aspect rating datasets, namely BeerAdvocate, RateBeer (ES), RateBeer(FR), Audiobooks and Toys & Games, from (a) to (e); (2) sentiment datasets, namely TED comments (f); (3) emotion attributes datasets, namely TED talks from (g) to (j). The emotion values of the TED talks were normalized per talk. Note that the ratings in Audiobooks and Toys & Games datasets have been projected from the 1–5 interval to the 1–10 interval (hence, the zero counts on the ratings with odd numbers).

	Review labels									
	BeerAdvocate		**RateBeer (ES)**		**RateBeer (FR)**		**Audiobooks**		**Toys & Games**	
Model \ Error	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
AverageRating	14.20	3.32	16.59	4.31	12.67	2.69	21.07	6.75	20.96	6.75
Aggregated (ℓ_1)	13.62	3.13	15.94	4.02	12.21	2.58	20.10	6.14	20.15	6.33
Aggregated (ℓ_2)	14.58	3.68	14.47	3.41	12.32	2.70	19.08	5.99	18.99	5.93
Instance (ℓ_1)	12.67	2.89	14.91	3.54	11.89	2.48	20.13	6.17	20.33	6.34
Instance (ℓ_2)	13.74	3.28	14.40	3.39	11.82	2.40	19.26	6.04	19.70	6.59
Prime (ℓ_1)	12.90	2.97	15.78	3.97	12.70	2.76	20.65	6.46	21.09	6.79
Prime (ℓ_2)	14.60	3.64	15.05	3.68	12.92	2.98	20.12	6.59	20.11	6.92
Clustering (ℓ_2)	13.95	3.26	15.06	3.64	12.23	2.60	20.50	6.48	20.59	6.52
APWeights (ℓ_2)	**12.24**	**2.66**	**14.18**	**3.28**	**11.37**	**2.27**	**18.89**	**5.71**	**18.50**	**5.57**
APW vs. SVR (%)	*+16.0*	*+27.7*	*+2.0*	*+3.8*	*+7.6*	*+15.6*	+1.0	*+4.5*	*+2.6*	*+6.0*
APW vs. Lasso (%)	*+10.1*	*+15.1*	*+11.0*	*+18.4*	*+6.8*	*+11.8*	*+6.0*	*+6.9*	*+8.1*	*+11.9*
APW vs. 2nd best (%)	+3.3	+7.8	+1.5	*+3.3*	*+3.7*	*+4.9*	+1.0	*+4.5*	*+2.6*	*+6.0*

Table 5.2: Performance of aspect rating prediction (the lower the better) in terms of MAE and MSE (× 100) with 5-fold cross-validation. All scores are averaged over all aspects in each dataset. The scores of the best method are in **bold** and those of the second best one are underlined. Significant improvements (paired t-test, $p < 0.05$) are in *italics*. Fig. 5.4 shows MSE scores per aspect for three methods on five datasets.

over the same range of possible values, noted $a \cdot 10^b$ with $a \in \{1,\dots,9\}$ and $b \in \{-4,\dots,+4\}$, hence 81 values per term. For the regression models, we use the *scikit-learn* machine learning library (Pedregosa et al., 2012)[5]. Our code and data are made available online[6].

5.7 Results on Text Regression Tasks

We report here our results on aspect, sentiment and emotion rating prediction. Moreover, we verify whether our contribution is independent of the selection of the feature space, and lastly we analyze qualitatively the parameters of the model to assess their interpretability.

5.7.1 Aspect Rating Prediction

The results for aspect rating prediction are given in Table 5.2. The proposed APWeights method outperforms all other baselines on each dataset. In particular, it outperforms the two models with the Aggregated Instance assumption, which correspond respectively to traditional SVR (ℓ_2 norm) and Lasso (ℓ_1 norm) which use BOW with counts. The SVR baseline has on average 11% lower performance than APWeights in terms of MSE and about 6% in terms of MAE. Similarly, the Lasso baseline has on average 13% lower MSE and 8% MAE than APWeights. As shown in Figure 5.4 on page 100, APWeights also outperforms them for each aspect in the five review datasets.

The Instance-as-Example assumption with ℓ_1 performed well on BeerAdvocate and Toys &

[5] http://scikit-learn.org/stable/.

[6] https://github.com/nik0spapp/weighted-MIL/.

	SENTIMENT LABELS		EMOTION LABELS	
	TED comments		**TED talks**	
Model \ Error	**MAE**	**MSE**	**MAE**	**MSE**
AverageRating	19.47	5.05	17.86	6.06
Aggregated (ℓ_1)	17.08	4.17	15.98	5.03
Aggregated (ℓ_2)	16.88	4.47	15.24	4.97
Instance (ℓ_1)	17.69	4.37	16.48	5.30
Instance (ℓ_2)	16.93	4.24	16.10	5.57
Prime (ℓ_1)	17.39	4.37	15.98	5.78
Prime (ℓ_2)	18.03	4.91	16.74	5.94
Clustering (ℓ_2)	17.64	4.34	17.71	6.02
APWeights (ℓ_2)	**15.91**	**3.95**	**15.02**	**4.89**
APW vs SVR (%)	*+5.7*	*+11.5*	+1.5	+1.6
APW vs Lasso (%)	*+6.8*	+5.3	*+6.0*	*+2.9*
APW vs 2^{nd} best (%)	*+5.7*	+5.3	+1.5	+1.6

Table 5.3: MAE and MSE (× 100) on sentiment and emotion prediction with 5-fold cross-validation. Scores on TED talks are averaged over the 12 emotions. The scores of the best method are in **bold** and those of the second best one are underlined. Significant improvements (paired t-test, $p < 0.05$) are in *italics*.

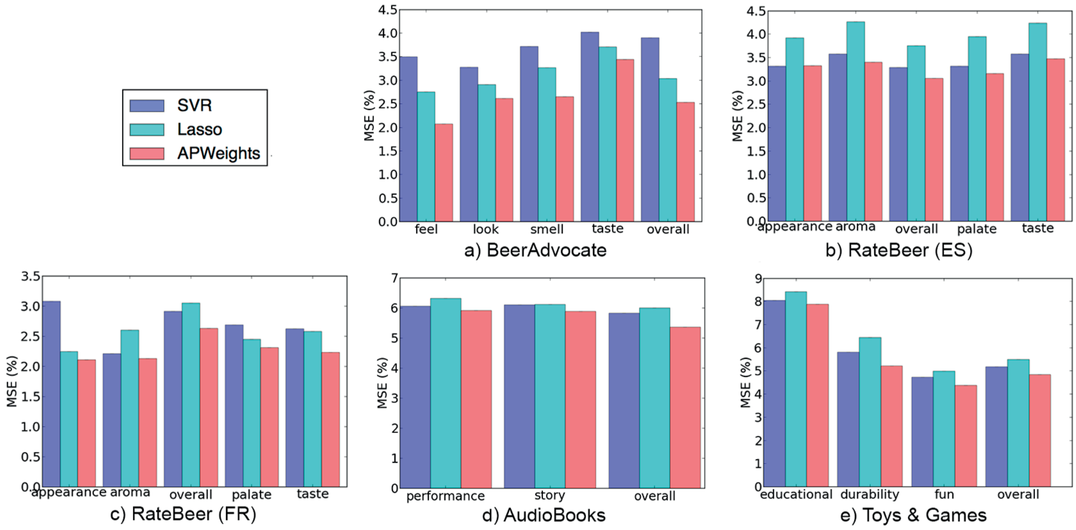
Games (for MSE), and with ℓ_2 it performed well on Ratebeer (ES), RateBeer (FR) and Toys & Games (for MAE). Therefore, this assumption is quite appropriate for this task, however it still scores below APWeights, by about 5% MAE and 8%–9% MSE. The Prime assumption with ℓ_1 performed well only on the BeerAdvocate dataset and with ℓ_2 only on the Toys & Games dataset, always with lower scores than APWeights, namely about 9% MAE and 15%–18% MSE. This suggests that the Prime assumption is not the most appropriate for this task. Lastly, even though Clustering is an instance relevance method, it has similar scores to Prime, presumably because the relevances are assigned according to the computed clusters and they are not directly influenced by the objective of the task.

To compare with the state-of-the-art results obtained by McAuley et al. (2012), we evaluated our proposal on three of their *full-size* datasets. Splitting each dataset in half for training vs. testing (as done by McAuley at al.) and using the optimal settings from our experiments above, we measured the average MSE over all aspects. APWeights improves over Lasso by 10%, 26% and 17% MSE respectively on each dataset: the absolute MSE scores are 3.8% for Lasso vs. 3.4% for APWeights on Ratebeer SP; 2.3% vs. 1.7% on Ratebeer FR; and 6.3% vs. 5.2% on Audiobooks. When compared to the best SVM baseline provided by McAuley et al., our method improves by 32%, 43% and 35% respectively on each dataset, though it did not use their rating model. McAuley et al. report MSE scores of 3%, 2% and 3% (respectively on the above data sets) for their best model, which uses a joint rating model and an aspect-specific text segmenter trained on hand-labeled data. These scores are comparable to those of our model (3.4%, 1.7% and 5.2%), which does not use these features, though it could benefit from them in the future.

Lastly, as mentioned by the same authors, predictors which use segmented text, for example

	BOW		TF-IDF	
Model \ Error	**MAE**	**MSE**	**MAE**	**MSE**
Aggregated (ℓ_1)	17.08	4.17	16.59	3.97
Aggregated (ℓ_2)	16.88	4.47	16.25	4.16
Instance (ℓ_1)	17.69	4.37	18.11	4.50
Instance (ℓ_2)	16.93	4.24	16.88	4.23
Prime (ℓ_1)	17.39	4.37	17.72	4.43
Prime (ℓ_2)	18.03	4.91	17.10	4.29
APWeights (ℓ_2)	**15.97**	**3.97**	**15.36**	**3.63**
APW vs SVR	*+5.3%*	*+11.2%*	*+5.5%*	*+12.5%*
APW vs Lasso	*+6.4%*	+5.0%	*+7.3%*	*+8.5%*
APW vs 2nd best	*+5.3%*	+5.0%	*+5.5%*	*+8.5%*

Table 5.4: MAE and MSE (× 100) on sentiment prediction with 5-fold cross-validation over TED comments, when changing the features space from BOW with counts to TF-IDF. Improvements in *italics* are significant with $p < 0.05$ (pairwise t-test).



a) BeerAdvocate

b) RateBeer (ES)

c) RateBeer (FR)

d) AudioBooks

e) Toys & Games

Figure 5.4: MSE scores of SVR, Lasso and APWeights per aspect over the review datasets.

with topic models as in Lu et al. (2011), do not necessarily outperform SVR baselines. Instead, they have marginal or even no improvements, which is why we did not further experiment with them. Interestingly, however, MIR algorithms under several assumptions go beyond SVR baselines with BOW and even more sophisticated features such as TF-IDF (see below).

5.7.2 Sentiment and Emotion Prediction

Our method is also competitive for sentiment prediction over comments on TED talks, as well as for talk-level emotion prediction with 12 dimensions from subsets of 10 comments on each talk (see Table 5.3). APWeights outperforms SVR and Lasso, and all other methods for each task. For sentiment prediction, SVR is outperformed by 11% MSE and Lasso by 5%. For emotion prediction (averaged over all 12 aspects), differences are smaller, at 1.6% and 2.9% respectively. The smaller differences could be explained by the fact that among the 10 most

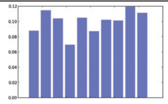
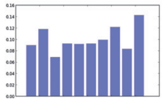
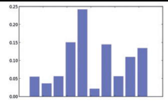
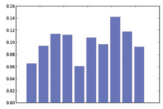
Class	Top comment per talk (according to weights ψ_i)	$\hat{\psi}_i$ distribution
inspiring	“It seems to me that the idea worth spreading of this TED Talk is inspiring and key for a full life. ‘No-one else is the authority on your potential. You’re the only person that decides how far you go and what you’re capable of.’ It seems to me that teens actually think that. As a child one is all knowing and all capable. How did we get to the (...)”	
beautiful	“The beauty of the nature. It would be more interesting just integrates his thought and idea into a mobile device, like a mobile, so we can just turn on the nature gallery in any time. The paintings don’t look incidental but genuinely thought out, random perhaps, but with a clear grand design behind the randomness. Drawing is an art where it doesn’t (...)”	
funny	“Funny story, but not as funny as a good ’knock, knock’ joke. My favorite knock-knock joke of all time is Cheech & Chong’s ‘Dave’s Not Here’ gag from the early 1970s. I’m still waiting for someone to top it after all these years. [Knock, knock] ‘Who is it?’ the voice of an obviously stoned male answers from the other side of a door, (...)”	
courageous	“I was a soldier in Iraq and part of the unit represented in this documentary. I would question anyone that told you we went over there to kill Iraqi people. I spent the better part of my time in Iraq protecting the Iraqi people from insurgents who came from countries outside of Iraq to kill Iraqi people. We protected families men, women, and (...)”	

Table 5.5: Four examples of top comments (according to weights ψ_i) for correctly predicted emotions in four TED talks (score 1.0) and the distribution of weights over the 10 most recent comments in each talk.

recent comments per talk, many are not related to the emotion that the system tries to predict.

5.7.3 Independence from Feature Space

As mentioned earlier, the proposed model does not make any assumption about the feature space. Thus, we examined whether the improvements it brings remain present even with a different feature space, for instance based on TF-IDF instead of BOW with counts. For sentiment prediction on TED comments, we found that by changing the feature space to TF-IDF, strong baselines such as Aggregated (ℓ_1) and (ℓ_2), i.e. SVR and Lasso, improve their performance, now reaching 16.25% and 16.59% MAE respectively (with 4.16% and 3.97% MSE). However, APWeights still outperforms them on both MAE and MSE scores, which reach 15.35% and 3.63%, thus improving over SVR by 5.5% on MAE and 12.5% on MSE, and over Lasso by 7.4% on MAE and 8.5% on MSE. These results suggest that improvements with APWeights could be observed also on even more sophisticated feature spaces.

5.8 Interpreting the Relevance Weights

Apart from predicting ratings, the MIR scores assigned by our model reflect the contribution of each sentence to these predictions. To illustrate the explanatory power of our model, we provide examples of predictions on test data taken from the cross-validation folds above. Table 5.5 displays the most relevant comment (based on $\hat{\psi}_i$) for several correctly predicted

Sentences per comment	$\hat{\psi}_i$	$\hat{y}_i$	y_i
“Very brilliant and witty, as well as great improvisation.”	0.64	5.0	5.0
“I enjoyed this one a lot.”	0.36		
“That’s great idea, I really like it!”	0.56	4.2	4.0
“I can’t wait to try it, but first thing, I need a house with big windows, next year, maybe I can do that.”	0.44		
“This dog and pony show is classic revisionist history.”	0.17	3.3	3.0
“Tesla, who was a true genius, wanted to give mankind free energy.”	0.26		
“This frightened those in power.”	0.18		
“This was the cause of his demise. ”	0.17		
“His secret is still out there waiting to be shown the light of day.”	0.22		
“Unfortunately countries are not led by gifted children.”	0.48	2.4	2.0
“They are either dictated by the most extreme personalities who crave nothing but power or managed by politicians who are voted in by a far from gifted population.”	0.52		
“I am very disappointed by this, smug, cliched and missing so much information as to be almost worthless.”’	0.43	1.8	1.0
“No mention of ship transport lets say 50% of all material transport, no mention of rail transport, no mention of agriculture with its vast use of petrochemical fertilizers, pesticides and diesel use.”	0.29		
“I am sorry to be so negative, this just sounds like a sales pitch that he has given too many times without any constructive criticism.”	0.28		

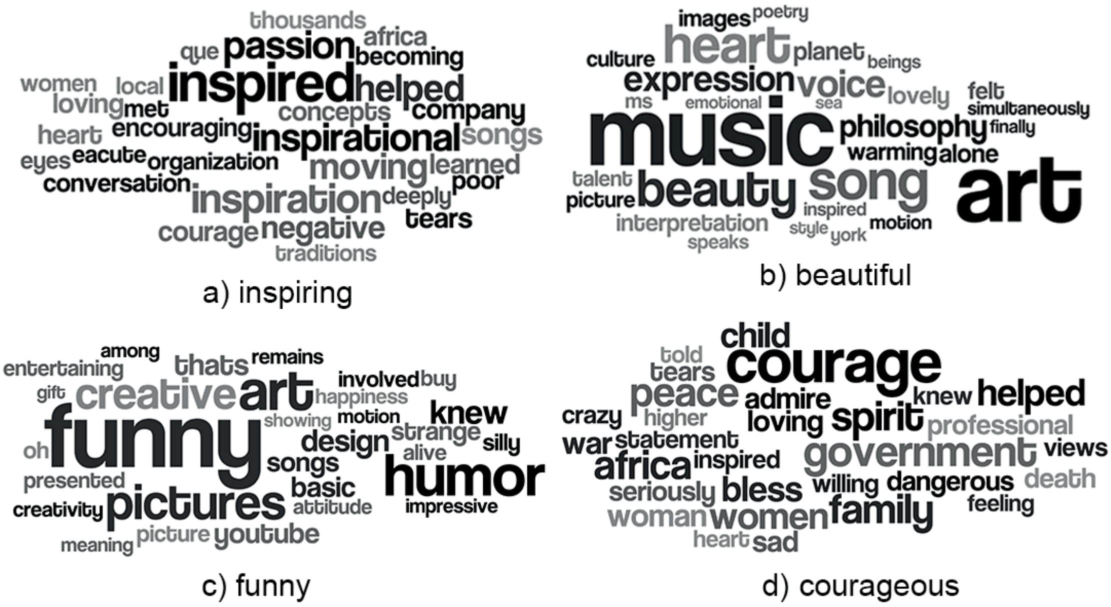
Table 5.6: Examples of predicted sentiment for TED comments: $\hat{\psi}_i$ is the estimated relevance of each sentence with respect to the sentiment, $\hat{y}_i$ is the predicted polarity score and y_i is the actual polarity score.

emotions of specific TED talks from the test sets, along with the distribution of $\hat{\psi}_i$ weights of the other comments of the talk. Four emotion classes are illustrated in the figure: ‘informative’, ‘beautiful’, ‘funny’ and ‘courageous’. The selected comments appear to reflect correctly the fact that the respective emotion is the majority one for each of the talks in the figure. As noted above, this task is particularly challenging because we use only the ten most recent comments for each talk.

Moreover, we represent in Figure 5.5 the top words in the vector space for the same sample of four emotions, as computed according to the Φ regression coefficients; they are displayed as tag clouds using freely available software[7]. Again, we observe that the most important keywords selected for each emotion match our intuitions about it, as they appear to be highly correlated with the emotion descriptor, e.g. ‘beautiful’ with various forms of art, or ‘inspiring’ with a range of synonyms such as ‘inspirational’ and ‘encouraging’. These clouds show the high precision of the Φ regression coefficients in selecting relevant keywords from the comments.

Table 5.6 displays TED comments selected from the test set of a given fold, for the comment-

[7] http://www.wordle.net/.



a) inspiring

b) beautiful

c) funny

d) courageous

Figure 5.5: Top words based on Φ for predicting four emotions from comments on TED talks, namely inspiring, beautiful, funny and courageous. The size of each word is proportional to its coefficient value in the regression hyperplane.

level sentiment prediction task. The table also shows the $\hat{\psi}_i$ relevance scores assigned to each of the composing sentences, the predicted polarity scores $\hat{y}_i$ and the actual ones y_i assigned by human raters. We observe that the sentences that convey the most sentiment are assigned higher scores than sentences with less sentiment, always with respect to the global polarity level. For example, the first sentence of the first comment (highly positive with a score of 5) is assigned a weight of 0.64, versus 0.36 for the second sentence: these values match our intuition regarding the contribution of each sentence to the overall sentiment of the comment. Similarly, for all the other polarity levels illustrated in the table, the highest ranked sentences appear to be the most relevant for the sentiment that the author wants to convey.

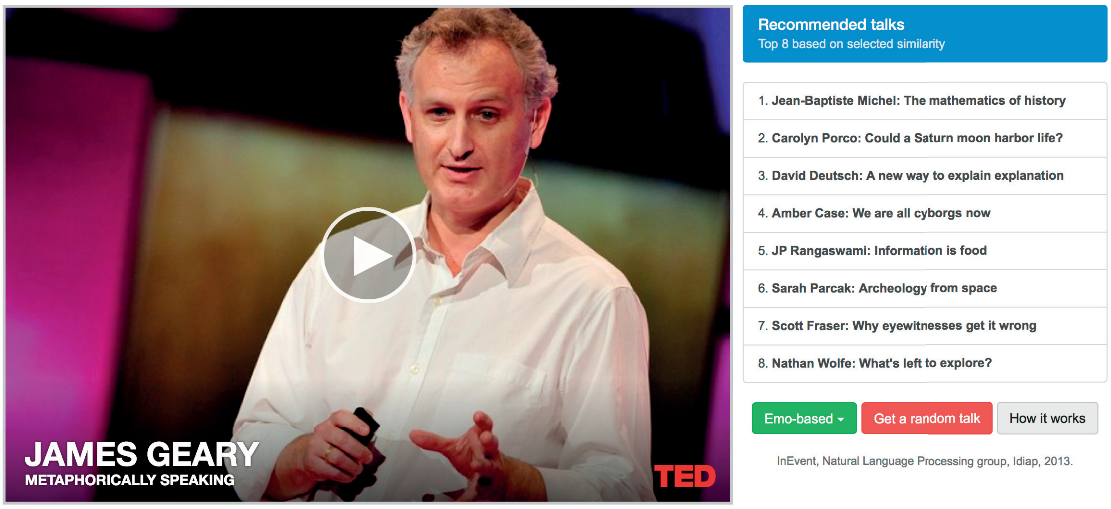
The examples presented here suggest that the APWeights model has more degrees of freedom for interpretation than the other models, since it is able to assign meaningful relevance weights to parts of a text (here, sentences) as well as words, while the state-of-the-art models to which we compare can only consider the relevance of words. Due to the lack of quantitative ground truth data for intrinsic evaluation, we illustrated the explanatory power only qualitatively with examples. However, it is possible to perform extrinsic evaluations of the assigned relevance weights on other tasks such as sentence ranking, segmentation or summarization – as we demonstrate in the next chapter.

5.9 Applications

In the previous section, we showed that the proposed MIR model is applicable to several types of text regression tasks, and that its results are interpretable as relevance weights. In this section, we demonstrate its versatility and practical value by applying it to two real-world scenarios, namely emotion-based lecture analysis for recommendation (5.9.1) and its explanation (5.9.2), and explanation of comment sentiment results (5.9.3).

5.9.1 Emotion-based Analysis and Recommendation

Within the context of the InEvent Portal[8], apart from the content-based recommendation functionality (presented in Section 3.9.1), we propose to recommend lectures based on emotion attributes. Using the MIR model, we analyzed 1000 TED talks over 12 real-valued emotion dimensions, and we generated recommendations for each talk based on emotions using cosine similarity. Figure 5.6 displays an example of top eight recommendations based on emotion for a TED talk. The MIR model is trained on the human-made transcripts and the community emotion labels which are obtained via the TED API. To learn from transcripts with known aspect ratings, the model assigns importance weights to each of the paragraphs, uncovering their contribution to the ratings per emotion. Then, the model is used to predict emotion attributes in previously unseen transcripts. Apart from its use for generic recommendation, this functionality might be also useful when designing and preparing a TED talk before presenting it to the general audience, to anticipate the emotions that could be perceived in it.



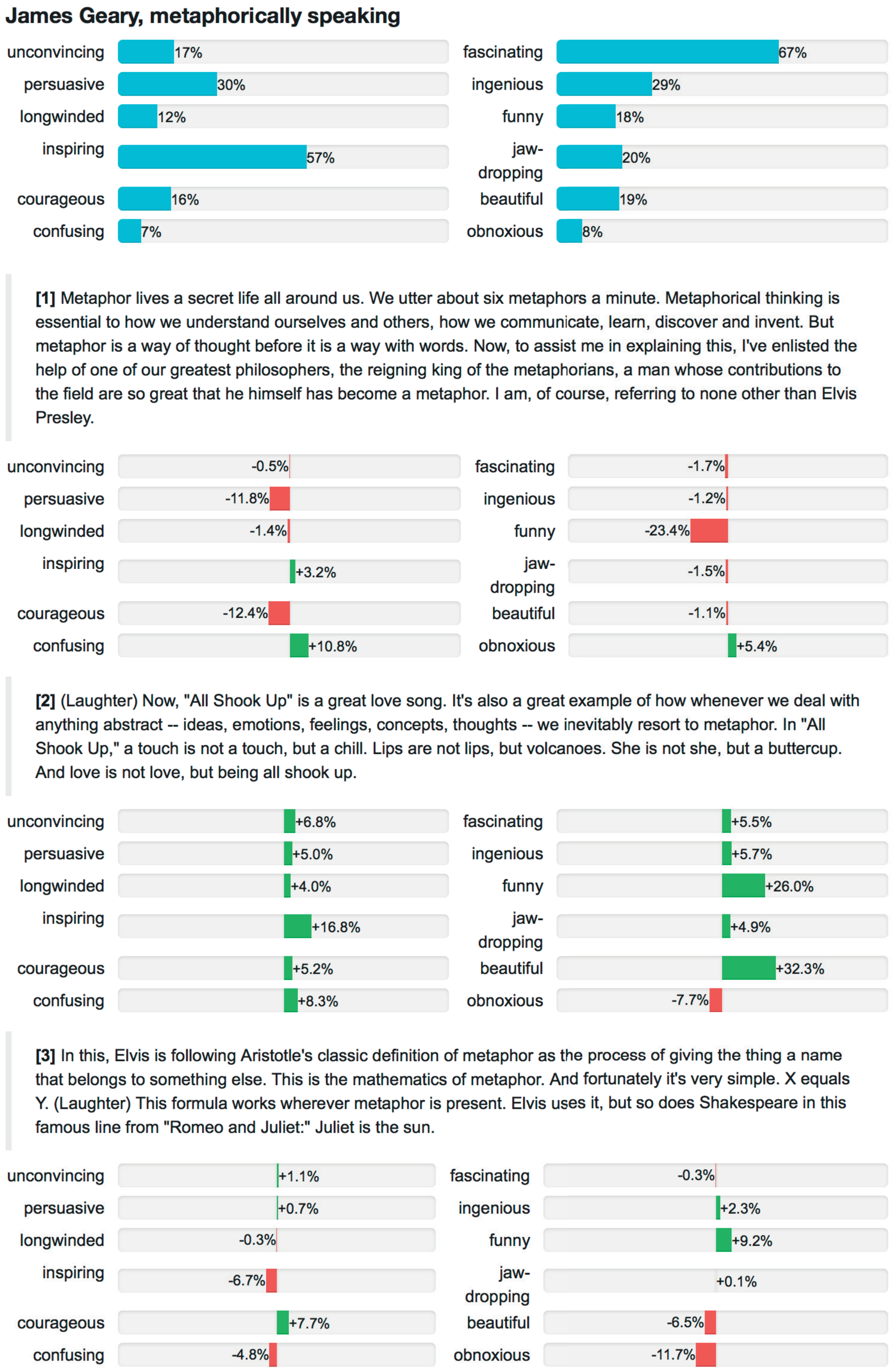
Recommended talks
Top 8 based on selected similarity

1. Jean-Baptiste Michel: The mathematics of history
2. Carolyn Porco: Could a Saturn moon harbor life?
3. David Deutsch: A new way to explain explanation
4. Amber Case: We are all cyborgs now
5. JP Rangaswami: Information is food
6. Sarah Parcak: Archeology from space
7. Scott Fraser: Why eyewitnesses get it wrong
8. Nathan Wolfe: What's left to explore?

InEvent, Natural Language Processing group, Idiap, 2013.

Figure 5.6: Example of top eight emotion-based recommendations for a TED talk ("Metaphorically speaking" by James Geary). The recommendations are based on emotion similarity (along 12 dimensions) hypothesized automatically for TED talks based on their transcripts.

[8] http://www.inevent-project.eu/demos/inevent-portal/.



James Geary, metaphorically speaking

[1] Metaphor lives a secret life all around us. We utter about six metaphors a minute. Metaphorical thinking is essential to how we understand ourselves and others, how we communicate, learn, discover and invent. But metaphor is a way of thought before it is a way with words. Now, to assist me in explaining this, I've enlisted the help of one of our greatest philosophers, the reigning king of the metaphorians, a man whose contributions to the field are so great that he himself has become a metaphor. I am, of course, referring to none other than Elvis Presley.

[2] (Laughter) Now, "All Shook Up" is a great love song. It's also a great example of how whenever we deal with anything abstract -- ideas, emotions, feelings, concepts, thoughts -- we inevitably resort to metaphor. In "All Shook Up," a touch is not a touch, but a chill. Lips are not lips, but volcanoes. She is not she, but a buttercup. And love is not love, but being all shook up.

[3] In this, Elvis is following Aristotle's classic definition of metaphor as the process of giving the thing a name that belongs to something else. This is the mathematics of metaphor. And fortunately it's very simple. X equals Y. (Laughter) This formula works wherever metaphor is present. Elvis uses it, but so does Shakespeare in this famous line from "Romeo and Juliet:" Juliet is the sun.

Figure 5.7: Example of emotion values (along 12 dimensions) hypothesized automatically for an entire lecture (blue bars) and, underneath, the relative relevance per emotion score of its first paragraphs (green and red bars). The recommendations resulting from emotion-based similarities are presented at top right of Fig. 5.6.

5.9.2 Explanation of Predicted Emotions of Talks

Figures 5.7 and 5.6 are snapshots of our online demonstration of emotion-based analysis and recommendation[9] related to the same TED talk. While browsing a particular TED talk, the user is able to view below the video player the estimated emotion values per dimension (blue bars). This allows users to inspect at a glance the emotional attributes of the talk. Below the emotion attributes, the user can view the transcript of the full TED talk segmented in paragraphs. For each paragraph the relative relevance (mean centered) to each emotion dimension based on MIR is displayed. This allows the user to inspect which paragraphs are more relevant to particular dimensions.

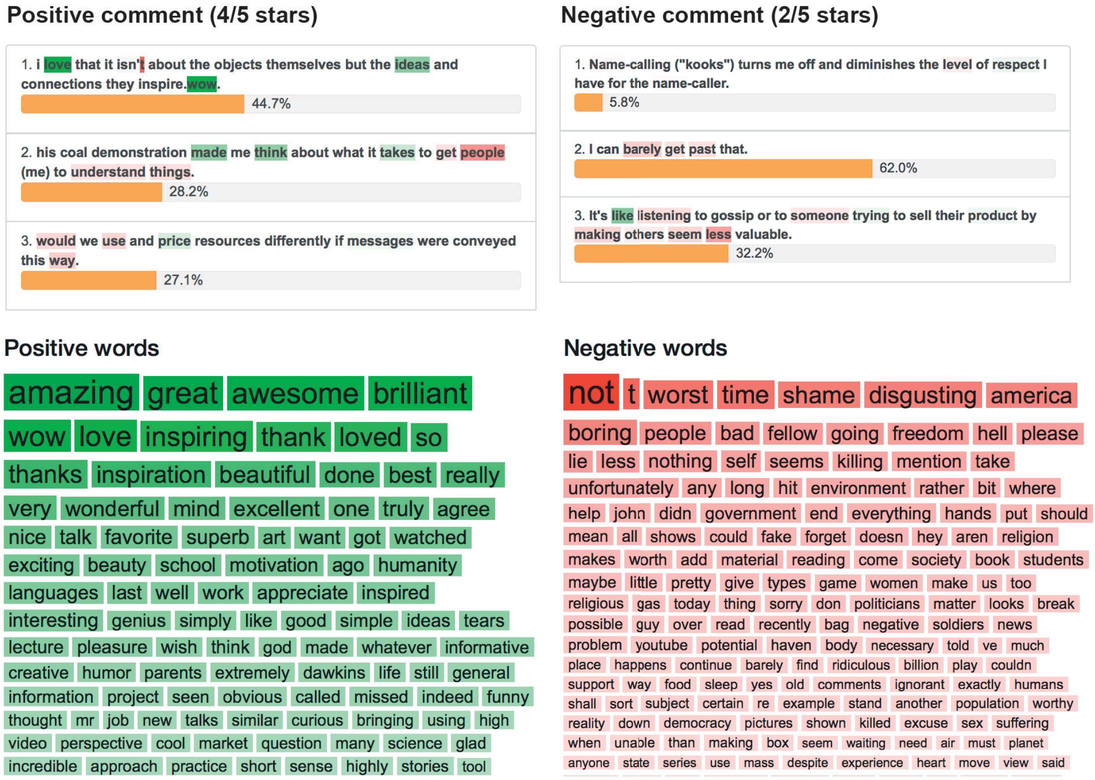
To further facilitate the browsing and searching within a TED talk, the system allows the user to click on the estimated emotion values (blue bars) of his/her preference to sort all the paragraphs of the transcript according to their relevance to the selected emotion dimension. For example, the user can click on the funny dimension and all the paragraphs which are most related to the funny emotion will appear on top. Essentially, this function serves as an explanatory tool of the estimated emotional attributes, since the user has a better idea of which parts of the TED talk contributed the most to the estimation of the emotions by the system. Similarly, when viewing the recommendations, it is possible to display excerpts of the most similar paragraphs to the paragraphs of the talk that is currently being viewed, in order to explain to the user why a particular talk has been recommended.

5.9.3 Explanation of Predicted Sentiments of Comments

Figure 5.8 displays the sentiment analysis of two sample user comments using the proposed MIR model trained on the TED comments dataset from Section 5.7. The orange bars show the contribution of each sentence to the overall sentiment value. The highlighted words belong to the most predictive words in positive comments (marked in green), and in negative ones (marked in red). On the left, a positive comment (4 out of 5 stars) comprised of three sentences is displayed. The first sentence of this comment is found to be the most relevant to its overall sentiment with 44.7%, and indeed it contains highly positive words such as ‘love’, ‘ideas’ and ‘wow’. The two other sentences follow with a lower relevance score, since both their content have fewer sentiment words than the first sentence.

On the right, a negative comment (2 out of 5 stars) comprising three sentences is displayed. The most relevant sentence to the comment’s negative sentiment is the second one with 62.0%, containing negative words such as ‘barely’, ‘get’ and ‘past’, followed by the third sentence with 32.2% and the first one with 5.8%. Even though the third sentence contains more slightly negative words and one highly negative word (‘less’), it still has a lower contribution than the second one, because its negative sentiment is compensated by a highly positive word (‘like’). The MIR analysis thus allows the summarization of the sentiment of comments by determining

[9] InEvent demonstration: Emotion-based analysis and recommendation, http://www.inevent-project.eu/demos/emotion-based-analysis-and-recommendation-of-lectures/demo.html.



Positive comment (4/5 stars)

1. i love that it isn't about the objects themselves but the ideas and connections they inspire.wow.

44.7%

2. his coal demonstration made me think about what it takes to get people (me) to understand things.

28.2%

3. would we use and price resources differently if messages were conveyed this way.

27.1%

Negative comment (2/5 stars)

1. Name-calling ("kooks") turns me off and diminishes the level of respect I have for the name-caller.

5.8%

2. I can barely get past that.

62.0%

3. It's like listening to gossip or to someone trying to sell their product by making others seem less valuable.

32.2%

Positive words

amazing great awesome brilliant
wow love inspiring thank loved so
thanks inspiration beautiful done best really
very wonderful mind excellent one truly agree
nice talk favorite superb art want got watched
exciting beauty school motivation ago humanity
languages last well work appreciate inspired
interesting genius simply like good simple ideas tears
lecture pleasure wish think god made whatever informative
creative humor parents extremely dawkins life still general
information project seen obvious called missed indeed funny
thought mr job new talks similar curious bringing using high
video perspective cool market question many science glad
incredible approach practice short sense highly stories tool

Negative words

not t worst time shame disgusting america
boring people bad fellow going freedom hell please
lie less nothing self seems killing mention take
unfortunately any long hit environment rather bit where
help john didn government end everything hands put should
mean all shows could fake forget doesn hey aren religion
makes worth add material reading come society book students
maybe little pretty give types game women make us too
religious gas today thing sorry don politicians matter looks break
possible guy over read recently bag negative soldiers news
problem youtube potential haven body necessary told ve much
place happens continue barely find ridiculous billion play couldn
support way food sleep yes old comments ignorant exactly humans
shall sort subject certain re example stand another population worthy
reality down democracy pictures shown killed excuse sex suffering
when unable than making box seem waiting need air must planet
anyone state series use mass despite experience heart move view said

Figure 5.8: Explaining sentiment analysis results with the proposed MIR model. One positive comment (left) and one negative comment (right) are analyzed and explained using MIR (upper part), with the orange bar showing the contribution of each sentence to the overall sentiment value. The most predictive words for positive class and negative class are displayed in the lower part.

the top relevant sentences. This enables users to have efficient access to meaningful sentences of comments, and brings to them deeper knowledge of how the sentiment results are obtained by the system.

5.10 Conclusion

In this chapter, we introduced a novel multiple-instance regression model for aspect rating prediction from text, which learns instance relevance weights together with target labels. To the best of our knowledge, this has not been considered before. Compared to previous work on relevance assignment for MIR, the proposed model is able to predict labels on unseen bags in addition to assigning relevance to instances of previously labeled bags, while remaining efficient in terms of complexity. When used for aspect rating prediction, our proposal outperforms state-of-the-art MIR and non-MIR methods on seven publicly available datasets. These results make it applicable on a large variety of available labeled texts with numerical target labels.

Compared to previous work on aspect rating and sentiment prediction, our model performs significantly better than BOW regression baselines (SVR, Lasso) without using additional knowledge or features. The improvement persists even when the sophistication of the feature space increases, suggesting that our contribution is to a certain extent independent from feature engineering or learning. Lastly, the qualitative evaluation on test examples demonstrates that the parameters learned by the model are also interpretable and have explanatory power, thus making a step further towards understandable models.

Our experimental results on three text regression tasks and seven different datasets validate our hypothesis that the labels in text regression tasks are only weakly connected to the various segments of a text, and that the appropriate framework for learning is therefore one which accounts for this uncertainty in the input, namely the multiple instance learning framework. This finding is the most important outcome of our study, and it differentiates it from previous methods which assumed a classical strongly supervised learning setting. Another important finding is that the models which are based on weak supervision are more interpretable than the ones that are not based on it. This has great potential impact in several applications such as the explanation of recommendations, sentiment or emotion attributes and the design of new text-based algorithms which are able to better justify their estimations.

6 Learning Aspect Saliency of Sentences for Review Understanding

Understanding the precise attitude of users toward the aspects that they discuss in reviews remains an open problem in NLP, with potential impacts on information filtering, behavior analysis, and computational psychology. The majority of methods for modeling aspect ratings have focused on feature engineering or learning, assuming a supervised learning objective. Nonetheless, the data often comprises weak labels assigned by users at the review level, which makes it hard to determine which specific part of the review motivates them. In the previous chapter, we have cast this problem as a weakly supervised one using weighted multiple-instance regression (MIR), with competitive results on aspect rating prediction. However, although we interpreted the instance weights as the saliency or relevance of each sentence to the targeted aspect, this interpretation was only analyzed qualitatively on a few examples. In this chapter, we will demonstrate quantitatively that the saliency weights estimated by MIR capture meaningful structural information from texts on review understanding tasks.

6.1 Introduction

Methods for multi-aspect sentiment analysis have been typically evaluated on three tasks pertaining to the understanding of reviews: (1) *segmentation*, i.e. identifying the sentences of a review which discuss each ratable aspect; (2) *summarization*, i.e. choosing the most likely sentence per aspect in a review; and (3) *rating prediction*, i.e. determining the review-level attitude of the user toward each aspect. These operations when applied on reviews that are not accompanied by explicit aspect ratings enable the extraction of preferences to be used by recommender systems. Moreover, displaying aspect-based summaries of reviews makes recommendations more understandable and trustable.

In this chapter, we propose an approach based on MIR that learns the connections between sentences and aspects based on the words of the sentences. For each aspect, the method learns the saliency or relevance weights of the sentences of a review in an unsupervised manner, as in the previous chapter, and uses them to predict the aspect rating, but also, in combination with a Conditional Random Field (CRF) model, to perform aspect-based segmentation and

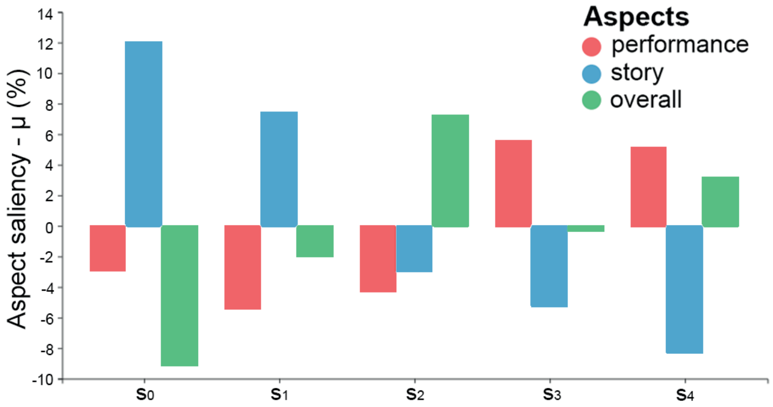


Figure 6.1: Aspect saliency weights estimated with MIR for an audiobook review of five sentences. The weights reveal the contribution of the sentence to the review's aspect ratings.

summarization. Figure 6.1 displays the aspect saliency weights estimated with MIR on an audiobook review. Our main goal is to evaluate the informativeness of such aspect saliency features from MIR for review segmentation and summarization, and thus to validate our previous findings of MIR effectiveness on aspect rating prediction over additional, larger datasets than in Chapter 5, by comparing them with state-of-the-art. Our contributions to review understanding can be summarized as follows: (i) we evaluate the usefulness of weighted MIR on six multi-aspect datasets with a total of two million reviews, for products such as beers or toys, with 3 to 6 aspects per product; (ii) we augment word or topic feature spaces with contextual feature templates bringing high-level semantics related to aspect saliency and sentiment; and (iii) we reach or exceed the state-of-the-art scores, thus validating the effectiveness of the proposed approach.

The rest of the chapter is organized as follows. In Section 6.2, we recap on how to use MIR for estimating aspect saliency weights and ratings, and in Section 6.3, we show how to combine them with CRF models for segmenting and summarizing reviews. In Section 6.4, we provide the description of the datasets, evaluation, and baselines. In Section 6.5, we compare the results across CRF models with and without employing MIR features on review segmentation and summarization, while in Section 6.6, we compare MIR-based methods with other state-of-the-art methods on three review understanding tasks. Lastly, in Section 6.7, we perform a qualitative analysis through examples for segmentation and summarization which have not been discussed and illustrated in the previous chapter.

6.2 Learning to Predict Aspect Ratings

In the previous chapter, we proposed a weighted MIR model which can be applied to various text regression tasks such as aspect rating prediction. As a reminder, the prediction of the

ID	Aspect	Sentence
s_0	story	This book was nearly as good as the first one in the series.
s_1	story	It seemed the ending was at least an hour or more too late.
s_2	story	When I thought it should be over, I checked how many minutes I had left and knew I was not even close.
s_3	performance	I liked the narration, I thought he did a good job.
s_4	overall	Still a 4 star rating: good story, good characters, and looking forward to the third in the series.

Table 6.1: Sentences and actual aspect labels for the audiobook review in Figure 6.1.

target label of an unseen bag B_i is done as follows:

$$\hat{y} = f_2(B_i) = \Phi^T B_i \hat{\psi}_i$$

The weighted MIR model (Section 5.4), apart from its ability to predict aspect ratings, can also estimate the saliency of sentences with respect to each aspect. The saliency is learned without using any knowledge of which aspect is being discussed in each sentence, i.o.w. they are learned in an unsupervised manner. Hence, it is beneficial to leverage them for improving the segmentation and summarization tasks. For the unsupervised case, the categorical aspect label of a sentence can be estimated by its maximum aspect saliency score as follows:

$$a = \underset{a}{\text{argmax}}\ \hat{\psi}_i^{(a)} \tag{6.1}$$

6.3 Learning to Segment Reviews

In this section, we describe how to use the learned aspect saliency weights for summarization and segmentation. To segment and summarize reviews in a supervised learning setting, we focus on Conditional Random Fields (CRFs), a class of structured conditional probabilistic models. These models are typically trained using bags of word-level features, from simpler ones (binary or numerical counts, TF-IDF weights) to more complex ones (POS tags, Wordnet synsets, and others). Here, in addition to BOW, we introduce in the CRF model new contextual features based on the sentence-level aspect saliency and sentiment features predicted by MIR. Although CRFs have been previously used for review segmentation, none of the past studies has modeled aspect saliency and sentiment information obtained without sentence-level supervision (as surveyed in Section 2.2.3).

6.3.1 Linear Chain CRF

Given a set of m reviews $B_i = \{b_{ij}\}_{n_i}^{d}$ (input variables) accompanied by a set of m aspect labels $Y_i = \{y_{ij}\}_{n_i}^{1}$ (output variables), with y_{ij} taking values from 1 to k aspects, a *first-order Linear Chain CRF* models the conditional distribution $p(y|b)$ as a globally normalized log-linear distribution (Lafferty et al., 2001):

$$p(y|b) \propto \prod_{i=1}^{m} \prod_{j=1}^{n_i} \psi_s(y_t, b_{ij}) \prod_{t=1}^{n_i-1} \psi_{s\curvearrowright}(y_{ij}, y_{ij+1}) \tag{6.2}$$

where $\psi_s(y_{ij}, b_{ij}) = exp(w \cdot f_s(y_{ij}, b_{ij}))$ models the aspect label of the j_{th} sentence of review i by means of a parameter vector w and a feature vector $f_s(y_{ij}, b_{ij})$, and $\psi_{s\curvearrowright}(y_{ij}, y_{ij+1})$ models the transition between the aspect variables at positions j and $j+1$ in the linear chain. Hence, this model accounts for sequential dependencies of aspects in the context of each review.

We consider three CRF models of increasing complexity: (i) an unstructured linear model (noted CRF) without sequential dependencies between the output variables; (ii) an undirected linear chain CRF (noted CRF-u) where the transitions are symmetric, using $(k^2+k)/2$ variables; and (iii) a directed linear chain CRF (noted CRF-d) where the transitions are asymmetric, using k^2 variables. For implementation we use the *Pystruct* library (Müller and Behnke, 2014).

6.3.2 Aspect Saliency and Sentiment Features

The typical bag-of-words features mentioned above lack information about high-level semantics such as the saliency and the sentiment of the sentences towards an aspect, which could be useful when deciding whether a sentence discusses one aspect or another. For example, when the salience of a given sentence with respect to an is high, it is likely that it refers to this particular aspect. Similarly, the polar sentiment towards an aspect of a given sentence is an indicator that it discusses the specific aspect.

Based on this observation, we propose a set of contextual feature templates listed in Table 6.2 intended to capture the aspect saliency and sentiment information, by using the weighted MIR model described above. Figure 6.2 displays the augmentation process of word or topic feature spaces with the proposed features. The saliency features are computed by the weight estimate $\hat{\psi}_a$ and the sentiment features are computed by $\hat{y}_a$ for each aspect a. The features in the first three rows (of each type) make use of the *local* context and are computed directly at the sentence-level, while the ones in the last row exploit the *global* context and are computed at the review-level: *cw* is the average sentence saliency and *cp* is the aspect rating prediction.

Sentences	**Saliency** ($\hat{\psi}$)	**Sentiment** ($\hat{y}$)	**Context**
current	*sw*	*sp*	*local*
previous	*sw_prev*	*sp_prev*	*local*
next	*sw_next*	*sp_next*	*local*
all	*cw*	*cp*	*global*

Table 6.2: Notations for the aspect saliency and sentiment feature templates based on MIR for a given review. The aspect saliency features are computed by the weight estimate $\hat{\psi}$ and the sentiment features are computed by the rating estimate $\hat{y}$ of the respective aspect models.

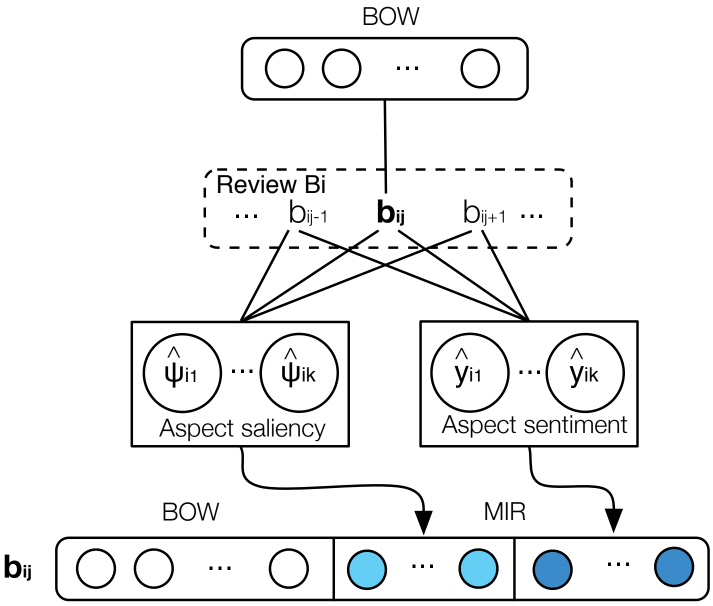

BOW
Review Bi
... b_{ij-1} $\mathbf{b_{ij}}$ b_{ij+1} ...
$\hat{\psi}_{i1}$ ... $\hat{\psi}_{ik}$
Aspect saliency
$\hat{y}_{i1}$ ... $\hat{y}_{ik}$
Aspect sentiment
BOW
MIR
$\mathbf{b_{ij}}$

Figure 6.2: Augmenting BOW with aspect saliency and sentiment features from MIR model.

Diversifying the segmented output

As observed by McAuley et al. (2012), sentences from different aspects tend to be assigned a single label when their aspects are very similar (e.g. *smell* and *taste* aspects on BeerAdvocate). These effects were diminished by enforcing diversity on the predicted output, i.e. choosing the most likely aspect assignments with the constraint that each aspect is assigned at least once. We adopt this method throughout our experiments.

For each review B_i a bipartite graph A is constructed which maps n_i sentences to their most likely aspect, where n_i is the length of the review. The compatibility between a sentence s of a review i and an aspect a is defined in this paper as the saliency estimate for the unsupervised case: $c_{sa} = \hat{\psi}_{is}^{(a)}$ or as the CRF probability estimate for the supervised case: $c_{sa} = p(a|s)$. The graph edges are defined as:

$$A_{s,l}^{(B_i)} = \begin{cases} c_{sl} & \text{if } 1 \leq l \leq k \\ \underset{a}{max}\, c_{sa} & \text{otherwise} \end{cases} \tag{6.3}$$

The first part of the function enforces each of the k aspects to have a matching sentence, while the second part allows other sentences to match any aspect (unconstrained). Given an assignment function f_a, the optimal cover a is given by Kuhn-Munkres algorithm as follows:

$$\hat{f}_a = \underset{f_a}{argmax} \sum_{s \in B_i} A_{s,f_a(s)}^{B_i} \tag{6.4}$$

The above objective is used for segmentation, while the same objective without the uncon-

Dataset	**Reviews** (n_r)	**Sentences** (n_s)	**Words** (n_w)	**Dim. (d)**	n_s/n_r	n_w/n_r
BeerAdvocate	1,586,259	16,883,058	109,733,059	19,418	10.6	69.1
Pubs	53,492	769,009	5,817,459	18,350	14.3	108.7
Toys & Games	373,974	2,105,647	16,136,343	31,984	5.6	43.1
Audiobooks	10,989	44,487	286,335	3,971	4.0	26.0
RateBeer (FR)	17,998	105,569	938,564	903	5.8	52.1
RateBeer (ES)	1,259	3,511	31,105	2,120	2.7	24.7

Table 6.3: Description and statistics of the six multi-aspect datasets.

strained nodes is used for the summarization task i.e. choosing k sentences which are the most compatible with the k aspects. In case of reviews with fewer than k sentences, the above constraints are discarded.

6.4 Data, Evaluation and Baselines

6.4.1 Description and Statistics of the Datasets

To evaluate our models we use six publicly available datasets of product reviews (McAuley et al., 2012) presented in Table 6.3: BeerAdvocate (1.6M reviews), Pubs (53K), Toys & Games (373K), Audible (10K), Ratebeer (FR, 18K), and RateBeer (ES, 1.2K). The aspect ratings were assigned by their creators from the respective websites, and the aspect labels were annotated by humans through crowdsourcing for a subset shown in Table 6.4. For instance, toys are rated in terms of educational value, durability, fun, and overall quality, while beers are rated according to how they feel, look, smell, or taste, plus an overall rating. The baseline features are binary vectors from a bag-of-words model (BOW) provided with the datasets. For segmentation and summarization, the sentence vectors were normalized with L2, so that the MIR features in Section 6.3.2 are in the same range as the BOW features.

6.4.2 Evaluation Protocol and Metrics

To compare between the three CRF models in a supervised setting, we will evaluate them in Section 6.5 over five random splits, 80% for training and 20% for testing. To compare our models with previous work (McAuley et al., 2012), we also replicate their experimental setup: a single 50/50 random split. To account for randomness effects in the evaluation, we report the average scores of each method over several runs. Specifically, we report scores of five random splits for segmentation and summarization, and two random splits for aspect rating prediction[1]. The methods are evaluated using the same accuracy and error metrics as those used by McAuley et al. (2012).

For segmentation, the goal is to predict the aspect labels of sentences, while for summarization,

[1] For segmentation and summarization we use more random splits than for aspect rating prediction to account for the fact that the corresponding evaluation sets are much smaller and have more variability.

Dataset	Reviews	Sentences	S/R	Aspects
BeerAdvocate	992 (.06%)	8,399 (.05%)	8.5	feel, look, overall, smell, taste
Pubs	100 (0.2%)	981 (0.1%)	9.8	vibe, selection, quality, service, food, price
Toys & Games	101 (.03%)	510 (.02%)	5.1	educational, durability, fun, overall
Audible	95 (0.9%)	439 (1.0%)	4.6	performance, story, overall
RateBeer (FR)	57 (0.3%)	279 (0.2%)	4.9	appearance, aroma, overall, palate, taste
RateBeer (ES)	115 (9.1%)	319 (9.1%)	2.8	appearance, aroma, overall, palate, taste

Table 6.4: Statistics of the segmentation labels for the six multi-aspect datasets.

the goal is to choose one sentence per aspect for each review and compare it to the ground-truth aspect labels. This particular definition of summarization requires one representative sentence for each aspect of an item, but it ignores the attitude of the user towards the given aspect of the given item. For instance, one positive and one negative sentence which discuss the same aspect are treated equally and are both correct according to the above definition (see also our related remarks in Section 6.7.2). For both tasks, we report the fraction of correct predictions as the accuracy. Summarization can also be cast as retrieving the most relevant sentences for each aspect; here, probabilities for each sentence are computed per aspect to produce a ranking, and then the Area Under Curve (AUC) is computed using the ground-truth aspect labels. For aspect rating prediction, we report the Mean Squared Error (MSE).

The model and feature selection are always performed by cross-validation on the training data using the same range of values for their regularization terms. All the CRF models are optimized over the same range of values for their regularization term, noted $a \cdot 10^b$ with $a \in \{1,\dots,9\}$ and $b \in \{-3,\dots,+3\}$. For the three regularization terms of the MIR model, we use the optimal values from the previous chapter, in Section 5.6. To select the best MIR features, we train MIR on each dataset (unsupervisedly with respect to segmentation labels) and we test all the unique combinations of features from Table 6.2 in addition to BOW features.

6.4.3 Baseline Systems for Review Segmentation and Summarization

For segmentation and summarization, the baselines from previous work include unsupervised, semi-supervised and fully-supervised methods, as listed below:

1. **LDA**: Latent Dirichlet Allocation (Blei et al., 2003) is a popular topic modeling method which is trained here with various numbers of topics: k (the number of aspects of each dataset), 10, or 50 and is optimized over the training set for an optimal alignment between topics and aspects. Therefore, this baseline is considered to be semi-supervised.

2. **PALE LAGER**: The probabilistic model proposed by McAuley et al. (2012) which jointly learns which words discuss a particular aspect, and which words are associated with a particular rating. This model achieves state-of-the-art performance on the tasks we consider and supports three types of learning, namely unsupervised, semi-supervised and fully-supervised.

	BeerAdvocate		**Pubs**		**Toys & Games**		**Audible**		**RateBeer (FR)**		**RateBeer (ES)**	
Model	BOW	+MIR	BOW	+MIR	BOW	+MIR	BOW	+MIR	BOW	+MIR	BOW	+MIR
CRF	81.30	**81.70**†	**64.77**	64.03	58.62	**60.69***	55.83	**58.07**†	87.57	**89.84***	81.02	**81.49**
CRF-u	87.55	**89.09***	67.35	**67.76**	59.94	**60.88**†	61.85	**65.08***	88.28	**89.28***	**83.07**	81.15
CRF-d	88.96	**89.03**	**69.73**	69.12	58.30	**60.60**	64.56	**64.68**	88.88	**90.90***	**83.42**†	82.73
CRF	88.07	**88.10**	62.54	**62.69**	35.74	**41.34***	56.20	**58.15***	74.86	**75.62**	53.33	**55.00**
CRF-u	87.89	**88.22***	61.18	**64.68***	40.23	**41.67**†	**56.39**	56.06	73.62	**74.29**†	55.33	**56.33**
CRF-d	87.53	**87.93**	61.14	**61.88**	38.41	**42.52***	54.91	**56.02**†	74.29	**74.29**	54.33	**56.00**
CRF	87.40	**88.06**	65.72	**67.79***	46.60	**48.96***	57.44	56.20	70.13	**70.64**†	47.08	**52.07**†
CRF-u	**87.40**	87.11	**67.21***	66.86	47.03	**48.05**	54.51	**56.44**†	70.22	**71.35**	49.85	**52.53** *
CRF-d	**86.67**	86.64	**67.24***	65.56	46.93	**48.47**†	55.17	**59.86**†	70.17	**72.21**†	51.18	**52.91**†

Table 6.5: Performance of CRF models, with and without MIR features, for: aspect segmentation (top three lines); summarization (middle three lines); and sentence ranking (bottom three lines). Accuracy is used as a metric for the first two tasks and AUC for the third one, all with 5-fold cross-validation. The best scores for each comparison between BOW and BOW+MIR are in bold, and the best scores for each dataset and task are underlined. The best scores marked with a '*' are significant at the $p < 0.1$ level, while those with a '†' at the $p < 0.16$ level.

6.4.4 Baseline Systems for Aspect Rating Prediction

For aspect rating prediction, the baselines from previous studies are structured and unstructured methods trained either over (i) segmented text, i.e. only the review sentences which belong to the target aspect, or over (ii) unsegmented text, i.e. all review sentences are used:

1. **SVM**: A well-known supervised learning model for classification or regression, Support Vector Machine (Cortes and Vapnik, 1995), with segmented or unsegmented text.
2. **Structured-SVM**: This method is generalization of SVM for training a classifier for general structured output labels (Tsochantaridis et al., 2005), and it learns the relationship between aspect ratings (rating model).
3. **PALE LAGER**: The probabilistic model proposed by (McAuley et al., 2012), combined with a structured SVM which learns the relationship between aspect ratings (rating model).

6.5 Results across CRF Models

The CRF models with MIR features for aspect saliency and sentiment outperform those not using these features, as shown in Table 6.5. Specifically, when looking at the 18 combinations of the three tasks and the six datasets, the CRF model with BOW+MIR features is the best performing one for 16 of them (89%). Furthermore, in 81% of the pairwise model comparisons (44 out of 54), the CRF model with BOW+MIR features improves over the CRF model using solely the BOW features.

6.5.1 Review Segmentation

The BOW+MIR features segment the reviews more accurately than only BOW features in 78% of the pairwise comparisons for this task, shown in the first three lines of Table 6.5. The structured models (CRF-u and CRF-d) outperform unstructured ones (CRF) over all datasets, highlighting the importance of modeling the relationships between aspects. The unstructured model (CRF) benefited the most from the MIR features, on average, followed by the structured models CRF-u and CRF-d. The MIR features were not informative on Pubs and Ratebeer (ES).

The CRF models with MIR features had the lowest segmentation scores on Pubs and Ratebeer (ES). One reason why MIR features were not informative is that feature selection was not efficient and caused models to overfit. Indeed, over the training sets, the CRF models with MIR features always outperformed the others, but this did not generalize to the testing sets.

6.5.2 Review Summarization

On the summarization task, the linear chain CRFs were outperformed by the unstructured CRF on two datasets (lines 4–6 of Table 6.5). When the linear chain CRFs are are given BOW+MIR features (lines 5–6 of Table 6.5 for BeerAdvocate, Pubs and RateBeer-FR), their performance increases and this leads to an improvement over the unstructured CRF (line 4 of Table 6.5 for the same datasets), highlighting once more their added value.

For sentence ranking (lines 7–9 of Table 6.5), the CRF models with MIR features always achieve the best scores, as in summarization above. Here, the linear chain CRFs perform better than for summarization (k most likely sentences for k aspects) but they still benefit from MIR features over all datasets apart from BeerAdvocate and Pubs, which are the datasets with the largest number of sentences per review (10.6 and 14.3, vs. less than 6 for the others). This indicates that a larger context allows the linear chain to learn more accurately the relationship between aspects for this task.

6.5.3 Feature Analysis

To determine the exact MIR features that are the most useful for each task, we analyze the feature importance over the training data, both for summarization and segmentation. We measure the importance of each type of features by the probability of an individual feature to appear in the top-5 models which showed an improvement over structured or unstructured CRF with BOW, separately per dataset. The results are represented as heatmap diagrams in Figure 6.3 for each dataset. When averaged over the six datasets, it appears that the saliency features have a higher importance compared to the sentiment ones for both tasks, namely 0.86 vs. 0.63 and 0.79 vs. 0.70.

For segmentation, the sentiment features score higher than saliency ones over Audible, Toys and BeerAdvocate. Over the other datasets, the saliency of the neighboring sentences (*sw_prev*,

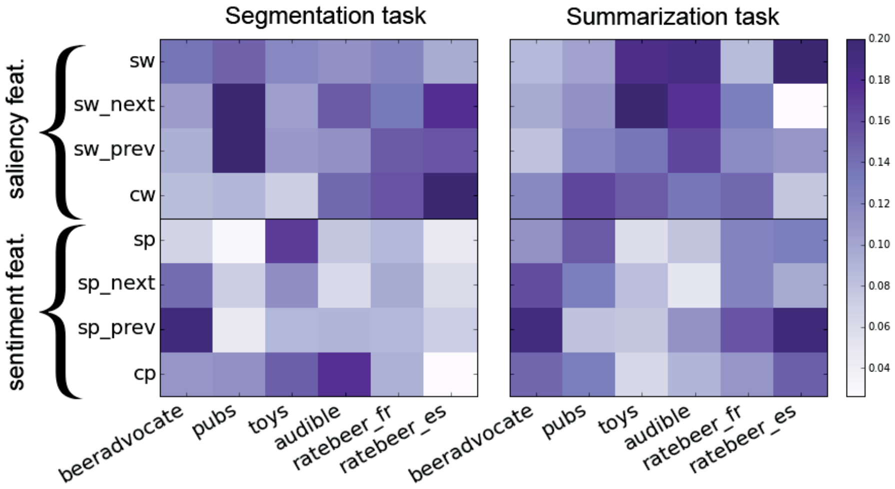


Figure 6.3: MIR feature importance for segmentation and summarization (10-fold c.-v. on the training data), as their probability to appear in the top-5 models better than CRF with BOW.

sw_next) and the global saliency (*cw*) are the most informative features. For summarization, the saliency of the current sentence (*sw*) is the most important feature for Toys, Audible and Ratebeer (ES), the global saliency (*cw*) for Ratebeer (FR) and Pubs, and the aspect sentiment of neighboring sentences (*sp_prev*, *sp_next*) for BeerAdvocate.

The proposed MIR features have discriminative potential for both tasks, although the optimal features are data dependent. The best performing features across all datasets and models are: for segmentation, the saliency of the next sentence (*sw_next* at 0.91) and the global sentiment (*cp* at 0.72); and for summarization, the saliency of the current sentence (*sw* at 0.86) and the sentiment of the previous sentence (*sw_prev* at 0.81).

6.6 Results of MIR-based Models Compared to SOA Models

We now compare the performance of state-of-the-art models presented by McAuley et al. (2012): (i) with the proposed models which use MIR features, namely one unstructured model, CRF, and the top-performing structured model, CRF-u, for segmentation and summarization, and (ii) with the weighted MIR model proposed in the previous chapter, for aspect rating prediction. To obtain comparable scores, we follow the evaluation protocol used by McAuley et al., i.e. uniformly splitting the data in training and testing subsets multiple times, and reporting the average score of all runs.

6.6.1 Review Segmentation

For segmentation, the proposed linear chain CRF with MIR features outperforms all other baselines over all datasets (see Fig. 6.4, upper part), with about 15% relative improvement

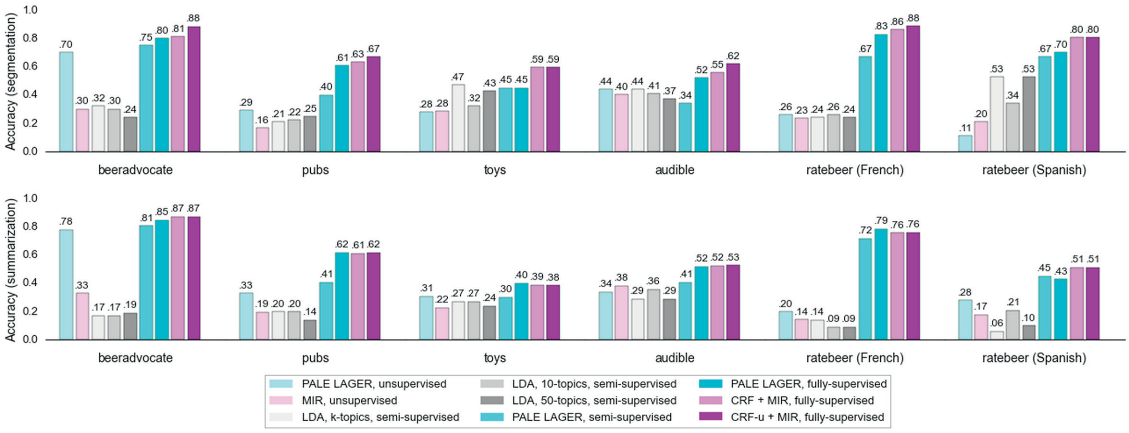


Figure 6.4: Accuracy on review segmentation (top) and summarization (bottom) of the CRF models with BOW+MIR features, compared to several unsupervised, semi-supervised and fully-supervised baselines. Note that the number of topics k in the "LDA k-topics" method is the number of aspects of the respective dataset. The scores of LDA and PALE LAGER are from McAuley et al. (2012).

on average over the best baseline (PALE LAGER, fully-supervised). The largest differences appear on the Toys and Audible datasets with about 31% and 14% improvement respectively. Compared to the semi-supervised models, LDA and PALE LAGER, the improvements are much higher, as expected. The CRF-u model performed better than the simple CRF, demonstrating that capturing the relationship between aspects is beneficial for these tasks, as already shown in Section 6.5.

The unsupervised PALE LAGER exhibits high scores especially on large datasets such as BeerAdvocate and Pubs, where it clearly outperforms the unsupervised saliency weights from MIR, while on the smaller datasets the performances of the two are similar. This can be attributed to the ability of PALE LAGER to decorrelate aspect words from aspect sentiment words, which is not the case for MIR weights. Nonetheless, the unsupervised MIR weights have reasonable performance on most datasets except from Pubs, presumably due to its high number of aspects.

6.6.2 Review Summarization

The performance on review summarization of the CRF with MIR features is slightly higher on average (+2.3%) than the fully-supervised PALE LAGER, with the best scores recorded on Ratebeer (ES) and BeerAdvocate (+18.6% and +2.3%) and the worst on Ratebeer (FR) and Toys (-3.8% and -5%), as shown in Fig. 6.4, lower part. Hence, the proposed model has better segmentation than PALE LAGER as well as good scores on summarization. One explanation for not always making an improvement on summarization (e.g. on Toys), with a better segmentation model than PALE LAGER, is that here only the reviews with at least *k* sentences are considered. It appears that such reviews are more difficult to segment than

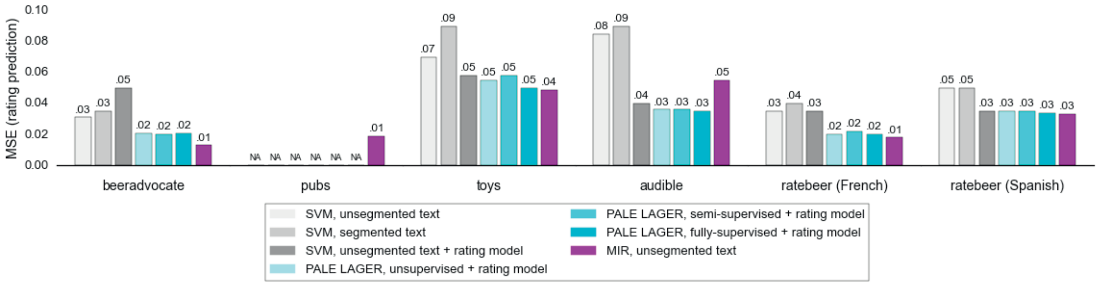


Figure 6.5: Mean squared error (MSE) on rating prediction for the MIR model with unsegmented text, compared to structured or unstructured supervised baselines, with unsegmented or segmented text. The scores of SVM and PALE LAGER are from McAuley et al. (2012).

reviews with fewer sentences.

As for the unsupervised setting, PALE LAGER outperforms the MIR and LDA models in most cases. The unsupervised use of saliency weights from MIR scored below PALE LAGER (except for Audible), but similarly to or above most LDA models over all datasets. Therefore, the best option for unsupervised learning is the PALE LAGER model for both segmentation and summarization. For sentence ranking, PALE LAGER was evaluated only on BeerAdvocate, with an average AUC of 0.87 (fully-supervised case), which is slightly below that of our best performing CRF, which achieved 0.88 AUC.

6.6.3 Aspect Rating Prediction Task

Figure 6.5 shows that the MIR model outperforms all other models over all datasets except Audible and Pubs (the latter was, however, not evaluated by McAuley et al. (2012)). This indicates that by uncovering the saliency of sentences for aspect ratings, even with unsegmented text only, the MIR model can go beyond increasingly complex baselines, including structured models (structured SVM), which make use of either unsegmented or segmented text (PALE LAGER). This means that, unlike PALE LAGER which requires a segmentation procedure to achieve the best scores, MIR does not require human intervention to perform as well or even better. Hence, MIR is able to learn structural information without explicitly modeling the relationship between aspect ratings. The structured models are more successful than MIR only on Audible, likely because this is the dataset with the fewest number of aspects.

Moreover, the MIR model is able to learn structural information without having to explicitly model the relationship between aspect ratings. Nonetheless, the structured learning approaches are more successful compared to MIR on Audible dataset. In fact, this dataset had the smallest number of aspects, which shows that the ability of MIR to capture structural information is lower than in reviews with greater number of aspects.

	BOW features		BOW + MIR features	
Aspects	**Sentence** s	$p(a\|s)$	**Sentence** s	$p(a\|s)$
Perf.	job interested keep changing listener narrator great	0.91	job interested keep changing listener narrator great	0.92
	didnt help narrator	0.88	parts female would read series suggest narrator continues	0.71
	decided think narrator voices characters english	0.68	didnt help narrator	0.70
	parts female would read series suggest narrator continues	0.64	performance dale comes reading mr given opinion	0.66
	audio job listen content laugh wonderful dont narrator	0.61	audio job listen content laugh wonderful don’t narrator	0.65
Story	story original compelling thought	0.82	story original compelling thought	0.85
	wow say	0.58	could short story told easily	0.66
	one	0.58	yet best changing constantly may style writing	0.50
	could short story told easily	0.58	story ability strength tell understand follow way easily	0.49
	intended perhaps value well attempt potential different	0.55	bringing interesting excellent story job life brown come	0.49
Overall	star gave 5	0.83	audio surprise totally took admit book	0.48
	audio surprise totally took admit book	0.73	audio recommended highly great book	0.42
	choice start first glad finish enjoyed really audible book	0.59	choice start first glad finish enjoyed really audible book	0.33
	world books still picture addition short local liberal future	0.45	much last books first 2 half horrible time book	0.26
	otherwise notes	0.45	world books still picture addition short local liberal future	0.25

Table 6.6: Top ranking sentences according to probability per aspect for an unstructured multi-class CRF with and without MIR features from the Audible test set.

6.7 Qualitative Analysis

We examine here the segmentation output of the best-performing CRF employing MIR features in terms of sentence ranking and summarization results on the Audible dataset.

6.7.1 Sentence Ranking Examples

Table 6.6 displays the five most probable sentences per aspect found by an unstructured multi-class CRF which employs MIR features compared to the BOW only baseline on examples from the Audible test set. Overall, it can be observed that the sentences obtained with MIR features tend to be longer and more related to the target aspect than those obtained using the BOW features alone. For example, with BOW, sentences 2 and 3 from the *story* aspect and sentences 1 and 5 from the *overall* aspect are very short and do not represent clearly the respective aspect. Another observation is that the MIR features tend to rank higher sentences which, apart from discussing the particular aspect, have higher sentiment – compared to those selected by using BOW features only, which tend to be more neutral. For example, this can be seen by comparing sentence 3 from BOW compared to sentence 4 from MIR on *performance*,

Review 1	**Selected sentence**	y_{ij}	$\hat{y}_{ij}$
Performance	(BOW) rushed story action felt fast paced (+MIR) nice listen real conversations series	8	9
Story	story good opinion	6	8
Overall	still enough give enjoyed rating	8	9
Review 2	**Selected sentence**	y	$\hat{y}$
Performance	(BOW) start honor comes stop series listening (+MIR) job good johnson narrating book	10	10
Story	political drama intense	10	10
Overall	(BOW) job good johnson narrating book (+MIR) exciting battles weber space famous	10	10
Review 3	**Selected sentence**	y	$\hat{y}$
Performance	(BOW) read hear really need (+MIR) story reader great superb	8	9
Story	like without story best sex defined type graphic	10	10
Overall	(BOW) havent eyre read forward something (+MIR) passion great book	10	10

Table 6.7: Summarization examples of three audiobooks using CRF with BOW+MIR features and BOW only. The last two columns display the actual and estimated rating by MIR for each aspect at the review-level. The different sentences produced by the two methods in the summaries are marked for correctness with green and red colors (correct and incorrect).

sentences 3 and 5 on *story* and sentences 4 and 5 on the *overall* aspect.

6.7.2 Summarization Examples

Table 6.7 lists three example summaries for both models along with the actual and estimated sentiment of each aspect at the review-level, which have decreasing agreement with the ground-truth, for CRF with BOW+MIR features. On the first example, it can be seen that BOW selected a sentence for the *performance* aspect which has in reality an unclear topic, as it might also refer to the *story* aspect, and does not reflect well the aspect rating since it has a negative sentiment. Similarly, on the second and third examples, BOW *performance* and *overall* sentences do not reflect well the aspect sentiment, and the *overall* sentences are more related to the *performance* aspect. It appears thus that MIR features tend to select sentences which are more representative of the aspect topic and sentiment, giving additional value on the produced summaries. This is a beneficial property which is not measured quantitatively by the current metrics used for the summarization task.

6.8 Conclusion

We have shown in this chapter that the MIR framework can greatly benefit three important tasks pertaining to review understanding. In contrast with previous methods, MIR formulates aspect rating prediction as a weakly supervised problem and allows models to have access to the structure of the input, thus making them more accurate and interpretable. MIR can augment word or topic feature spaces with high-level semantics regarding the aspect saliency and the sentiment of sentences within the context of each review, thus providing informative features for review segmentation and summarization, which in combination with CRF models demonstrate superior or similar performance to the state-of-the-art.

These results reinforce our findings regarding the effectiveness of MIR for aspect rating prediction presented in Chapter 5, and generalize them over six large publicly available datasets. Moreover, the results validate the ability of MIR to capture meaningful structural information of the input for review segmentation and summarization. Finally, our findings have a potential impact on several similar text regression tasks with weak labels, and make a step forward towards better understanding of user-generated ratings such as sentiment, emotion and preferences from their corresponding texts.

7 Conclusion

In this thesis, we proposed novel methods based on machine learning to model user sentiment and preferences for information filtering in social networks, which are effective, scalable and interpretable by humans. Mainly, we aimed to extract high-level semantic information from natural language, and to understand how this information is expressed in texts, in order to overcome the challenges of jointly modeling various types of user traces and explaining model estimations to users.

We first investigated semantic representations of items from their textual descriptions to improve generic and personalized recommendations. Moreover, we leveraged and adapted the sentiment of user comments to improve recommendation with unary feedback at collective or individual levels. Lastly, we proposed a novel way to cast text regression problems as weakly supervised ones and solved them with a weighted multiple-instance learning model which is effective and efficient. The learned instance saliency weights are easily interpretable and enable explainable model estimations; they are also useful indicators for segmenting and summarizing written texts.

This conclusion is organized as follows: we first review the achievements and specific contributions of the thesis, and then we discuss the limitations of the proposed methods and propose a series of promising future work directions.

7.1 Achievements

The achievements of this thesis cover two main directions: (1) the successful mining of semantic information from texts to improve the learning of unary preferences, and (2) the learning of explainable or easy-to-interpret models of sentiment from texts in online social networks. Both of them, as shown throughout this thesis, are beneficial to information filtering via generic or personalized recommendation (Chapters 3 and 4), aspect rating prediction (Chapters 5 and 6), and review summarization and segmentation (Chapter 6). Our contributions related to these achievements are summarized below.

Mining semantic information from text to improve the learning of unary preferences

In Chapter 3, we created a new metadata set extracted from the TED lectures to evaluate generic and personalized multimedia recommendation based on unary feedback. Feature selection experiments over the most active TED users showed that the most informative data fields for content-based (CB) methods are the description and the title of each lecture. Moreover, we showed that the semantic-based methods (ESA, RP and LSI) were more accurate than keyword-based ones (TF-IDF) in a cold-start setting, and that CB methods were outperformed by collaborative filtering (CF) in a non-cold-start one. The semantic-based methods which rely on external knowledge (ESA) were more informative for personalized recommendation, while the ones which rely on domain knowledge (LSI) were more informative for generic recommendation. To achieve performance closer to CF ones, we proposed a combined method using a neighborhood model, user/item popularity biases and content-based similarities (LSI, TF-IDF). The proposed method can be used when newly-added and older items are both present, as it does not rely entirely on collaborative rating similarities.

In Chapter 4, we proposed sentiment-aware models to improve one-class CF on three real-world multimedia datasets, namely TED talks, Vimeo videos and Flickr images, demonstrating significant improvements over models that do not use sentiment information. Moreover, we showed that the improvements of sentiment-aware models hold for all three negative class assumptions, meaning that the benefits gained from various strategies for balancing the negative class are likely to be preserved when these strategies are combined with our model. We also showed that the adaptive sentiment-aware models performed better than those with a fixed mapping. This is likely because the former are able to adapt the sentiment scores to the user preferences, and in particular to model cases in which the output scores of the sentiment classifier do not exactly match actual preferences. The proposed models are relevant to many real-world applications to communities where users interact both in terms of explicit feedback (favorites, likes) and in terms of textual feedback (comments, discussions). We experimented with three different types of content – lectures, general-purpose videos, and images – thus showing that our models are not constrained by a domain, and can adapt to domain data through learning.

Learning explainable models of fine-grained sentiment from text

In Chapter 5, we introduced a novel multiple-instance regression model (MIR) for text regression tasks which learns instance relevance weights together with target labels. Unlike previous work on relevance assignment, the proposed model can estimate efficiently the instance weights of unseen bags. Our proposal outperforms state-of-the-art MIR and non-MIR methods on aspect rating prediction, sentiment prediction and emotion prediction, over seven publicly available datasets. These results show that the model is applicable to a large variety of texts labeled with numerical target labels. Our model outperforms bag-of-words baselines (SVR, Lasso) without using additional knowledge or features, and the improvement persists even when the sophistication of the feature space increases, suggesting that our contribution is to a certain extent independent from feature engineering or learning. Our results validate the hypothesis that the labels in text regression tasks are only weakly connected to the various segments of a text, and that MIR is an appropriate framework for learning as it accounts for this uncertainty in the input. Apart from being more accurate, the weakly supervised models are more interpretable than fully supervised ones, which has great potential impact on several applications such as the explanation of recommendations, sentiment or emotion attributes, and the design of new text-based algorithms which can better justify their estimations.

In Chapter 6, we showed that the saliency weights from MIR, apart from being interpretable, are beneficial to three important tasks pertaining to review understanding, namely aspect rating prediction, review segmentation and summarization. MIR can augment word or topic feature spaces with high-level semantics regarding the aspect saliency and the sentiment of sentences within the context of each review, thus providing informative features for review segmentation and summarization, which in combination with CRF models demonstrate superior or similar performance to the state-of-the-art. These results generalize our findings from Chapter 5 over six large publicly available datasets and validate the ability of MIR to capture meaningful structural information of the input for review segmentation and summarization. Lastly, our results contribute to better understanding of user-generated ratings such as aspect, sentiment and emotion from their corresponding texts.

7.2 Discussion and Perspectives

The proposed approaches exhibit desirable properties in terms of effectiveness, efficiency and interpretability across several different information filtering tasks, but several of their limitations should be addressed in future work. We first provide a quick overview of limitations with their possible solutions, then develop each of them to some extent.

Brief Overview of Limitations and Possible Solutions

In Chapter 3, the semantic-based methods were most effective when ratings were scarce and, conversely, less effective when ratings were abundant. This property makes them mostly applicable to repositories in which new items are inserted frequently and therefore have few ratings. An effective application to abundant ratings could be achieved by a more principled integration with collaborative filtering methods, for instance through hybrid methods. The sentiment-aware models proposed in Chapter 4 are only applicable to users who have written at least a few comments, and are therefore vulnerable to the cold-start scenario with scarce comments. This limitation could be overcome by generalizing the model to other types of predictors, for example a nearest neighbor model with user-based similarities instead of item-based ones, which would allow transferring sentiment information from comments between similar users and thus addressing to an extent the comment sparsity issue.

The MIL-based methods proposed in Chapter 5 provided strong evidence that text regression tasks can effectively be cast as weakly supervised problems. In this new spectrum of weakly supervised methods, there are several possibilities for improvement, such as a joint learning objective for the proposed weighted MIR model along with alternative MIR objectives that provide desirable properties for learning. For instance, with multiple-instance logistic regression, we can transfer information from group labels to instance labels. In Chapter 6, the optimal combination of aspect saliency and sentiment features was obtained with exhaustive search using cross-validation which is a time-consuming procedure. One way to overcome this issue is by using a more robust and efficient feature selection method during the learning procedure or by learning the aspect saliency and sentiment features jointly for multiple tasks using multi-task learning. In the rest of this section, we highlight the most interesting research direction per chapter, and provide motivation for other future research directions.

Chapter 3: Integrate content-based and collaborative information

An interesting future work direction is to further explore algorithms which integrate both CB and CF, in particular hybrid algorithms, especially given that the TED dataset has rich content information to be exploited. More specifically, the hybrid method could exploit ensemble learning from two individual predictors, CB and CF. Another idea is to perform early fusion of content-based and collaborative information with other learning models, such as matrix factorization. Lastly, another interesting direction is to assess recommendation performance when automatically-assigned values are available for metadata fields, for instance through automatic speech recognition (for the transcript of a media file), speaker detection, or automatic summarization.

Chapter 4: Generalize sentiment-aware models to other types of predictors

The adaptation of sentiment-aware models to other predictors, such as low-rank matrix factorization, is another interesting future work direction. This could be done by parametrizing the prediction function with new variables that will influence it according to the output of a sentiment classifier (for binary feedback) or a regressor (for real-valued feedback), as shown in this thesis for the case of local predictors such as neighborhood models. Moreover, it would be interesting to investigate other loss functions than least squares for adapting the sentiment to unary feedback. For instance, a ranking-based loss function could help learn a better mapping for top-N recommendation than the one explored in Chapter 4. Another research direction is the inference of more granular preference information from text by performing multi-aspect sentiment analysis, for instance based on the methods proposed in Chapter 5 (see also proposed extension below) again for improving the one-class CF task.

Chapters 5 and 6: Investigate alternative MIL objectives

The multiple-instance learning framework for text regression tasks proposed in Chapter 5 paves the way for several promising research directions. For instance, one of them is to investigate alternative formulations and assumptions for multiple-instance learning such as multiple-instance logistic regression (MILR) (Settles et al., 2008). Such a model can be tested on sentiment classification at the sentence-level and at the document-level, based only on document-level supervision (Täckström and McDonald, 2011). Note that the proposed weighted MIR model is not able to directly transfer label information from bags to individual instances, rather it computes the contribution of each instance to the target label. MILR computes the conditional probabilities of each instance using logistic regression as:

$$o_{ij} = P(y_{ij} = 1|b_{ij}) = \frac{1}{1 + e^{-(w \cdot b_{ij} + \delta)}} \tag{7.1}$$

where y_{ij} is the class of the instance b_{ij} and w is the vector of the regression coefficients associated with the features of the instance. These estimates are combined into a probability estimate for a bag using the softmax function:

$$o_i = P(y_i = 1|B_i) = softmax_a(o_{i1}, \ldots, o_{in}) = \frac{\sum_{j=1}^{n_i} o_{ij} e^{a o_{ij}}}{\sum_{j=1}^{n_i} e^{a o_{ij}}} \tag{7.2}$$

where a controls to which extend softmax approximates a hard max function. At this stage, it would be possible to directly compute the probability of a bag belonging to a class if we knew the labels of the instances, however most often they are not available. One way to learn to predict instance labels jointly with bag labels using only bag supervision is by minimizing the following least squares objective: $\mathscr{L}(w) = \sum_{i=1}^{m}(y_i - o_i)^2 + \Omega(w)$, where Ω is a regularization term (e.g. $\ell 2$ norm) for the regression coefficients w. The above objective can be solved with stochastic gradient descent.

Chapters 5 and 6: Combine MIL objective with feature learning

Perhaps the most closely related work to this research direction is a recent study by Kotzias et al. (2015). The authors used MIL with pre-trained deep learning features to transfer information from group labels to individual labels. The results of this study showed that the approach outperforms baselines for sentiment classification at the sentence and document levels. However, it is not clear whether the improvement is mostly due to the deep features or to the MIL objective, because simpler features such as bags-of-words were not included in the experiments. At this stage, one interesting line of work is to avoid using predefined features, and directly learn deep features for instances using a MIL objective over a large sentiment corpus, e.g. from Maas et al. (2011), by modifying the MILR model presented above (Eq. 7.2).

Chapters 5 and 6: Improve weighted-MIL objective

As for the weighted MIR model, new methods to estimate instance weights at prediction time could be investigated, e.g. using a unified objective. Further experiments with other model settings would be beneficial, such as regularization norms other than ℓ_2 and feature spaces other than BOW or TF-IDF. Moreover, it would be interesting to evaluate intrinsically the learned weights on sentence ranking when sentiment is assumed to be known using human judges. Lastly, another promising direction is to train jointly multiple tasks, for example aspect rating prediction and review summarization or review segmentation, as in multi-task learning.

Chapters 5 and 6: Extend to multiple modalities

Apart from the text regression tasks on which we focused on this thesis, an important research direction is the investigation of multiple modalities for example predicting the sentiment or emotion of images or videos using audio, visual and text modalities. For images as in (Borth et al., 2013), the instances could be predefined patches or regions of interest to predict the sentiment label of image tweets, while for videos as in (Ellis et al., 2014) the instances could result from a predefined temporal segmentation to predict the emotion of videos. The features per modality can be a combination of high-level and low-level features commonly used in the literature on multimodal sentiment analysis, typically resulting from the following procedures: (i) visual analysis, such as smile and head pose estimates, face muscle movements features and basic emotion estimates, (ii) acoustic analysis such as prosody features, energy features, voice probabilities, spectral features and (iii) linguistic analysis such as n-gram bag-of-word features and topic modeling, respectively.

Broader perspectives

We now turn to three broader and longer-term research perspectives, which are timely and compelling at the same time, namely: sentiment-aware recommendation across languages, explaining recommendations or decisions to users in interactive recommender systems, and mining massive amounts of user-generated texts to detect opinion norms and trends.

1. **Cross-lingual recommendation.** Users in online communities are not usually constrained by a single language for communicating their opinions. Dealing with multiple languages in information filtering and access systems is a challenging problem which requires attention. For instance, how to recommend products or services when opinions are written in several different languages? Which language to use when recommending items to a particular user? More importantly, in the context of sentiment-aware recommendation, how do sentiments expressed by users vary across languages, and what expressions are used to convey the same sentiments in different languages? Answering these questions would enable the development of applications such as cross-lingual recommendation. Recently, Jou et al. (2015) investigated how visual sentiment and

emotion semantics are expressed in social multimedia across languages. To achieve this, they developed an ontology comprised of multilingual sentiment and emotion polarized visual concepts by adapting semantic structures called adjective-noun pairs. One interesting future research direction is to use this ontology for learning language-independent semantic representations of text, to be used for representing items and users for the purpose of cross-lingual recommendation.

2. **Generating explanations of recommendations.** The models of aspect, sentiment or emotion proposed in this thesis are able to identify the relevance of textual parts with respect to target label. At the same time, they are easily interpretable and beneficial for explaining predictions to the users. One interesting future research direction is to use them to generate explicit explanations within recommender or dialogue systems. This can be achieved by analyzing known reviews with MIL methods, estimating the most relevant or salient sentences per user, and then showing them as explanations for unseen items with similar estimated ratings. To avoid presenting to users sentences exactly as they appeared in texts from other users, the system can learn to generate them based on a given target rating using user-independent or user-specific language modeling. Therefore, the system will generate original sentences which are descriptive of and compatible with the preferences of the given user or across all users. The generation model can then be used to explain a recommendation based on its estimated rating in three different ways: (i) according to the user-specific model, (ii) according to the user-specific model of similar users, or (iii) according to a general, user-independent model. The main challenges here are the fluency of such a generation model and the quality of adaptation to the particular user for whom recommendations are made

3. **Detecting opinion norms and trends.** The ability to process user-generated texts in terms of sentiment and emotion towards products or aspects of products, apart from benefiting information filtering tasks, also provides an opportunity for processing massive datasets to discover opinion norms and trends towards particular topics such as products, events, brands or individuals. The opinion norms could be described as global opinion tendencies aggregated from a large number of users which are more or less invariant of the time component (e.g. the event of 'Christmas holidays'), while the opinion trends could be described as sudden or abrupt changes in the opinion norms (e.g. the breaking news about a 'company scandal'). Given that this aggregated information is a result of numerous individual opinions which are hard for a human to inspect one by one, it should be possible to automatically provide comprehensible summaries of groups of opinions as shown in this thesis for individual ones (Chapters 5 and 6). This could help measuring the impact of products, news or events on the general public or monitoring the public responses towards them over time, thus enabling applications to social relations, industry, journalism, politics and psychology.

Bibliography

Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005. ISSN 1041-4347.

Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 217–253. Springer, 2011.

Gediminas Adomavicius, Nikos Manouselis, and YoungOk Kwon. Multi-criteria recommender systems. *Recommender Systems Handbook*, pages 769–803, 2011. doi: 10.1007/978-0-387-85820-3_24.

Deepak Agarwal, Bee-Chung Chen, and Bo Pang. Personalized recommendation of user comments via factor models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 571–582, Edinburgh, United Kingdom, 2011.

Fabio Aiolli. Convex AUC optimization for top-N recommendation with implicit feedback. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 293–296, Foster City, CA, USA, 2014. doi: 10.1145/2645710.2645770.

Jaume Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence*, 201:81 – 105, 2013. ISSN 0004-3702. doi: 10.1016/j.artint.2013.06.003.

Chris Anderson. *The Long Tail: Why the Future of Business Is Selling Less of More.* Hyperion, 2006.

Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems*, pages 561–568, Vancouver, BC, Canada, 2003.

Nino Antulov-Fantulin, Matko Bošnjak, Martin Žnidaršič, Miha Grčar, Mikolaj Morzy, and Tomislav Šmuc. ECML/PKDD 2011 discovery challenge overview. In *Proceedings of the ECML/PKDD 2011 Discovery Challenge Workshop*, Athens, Greece, 2011.

Ioannis Arapakis, Yashar Moshfeghi, Hideo Joho, Reede Ren, David Hannah, and Joemon M. Jose. Integrating facial expressions into user profiling for the improvement of a multimodal recommender system. In *Proceedings of the 2009 IEEE International Conference on Multimedia and Expo*, ICME'09, pages 1440–1443, New York, NY, USA, 2009a.

Ioannis Arapakis, Yashar Moshfeghi, Hideo Joho, Reede Ren, David Hannah, and Joemon M. Jose. Enriching user profiling with affective features for the improvement of a multimodal recommender system. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, CIVR '09, pages 29:1–29:8, Santorini, Greece, 2009b.

B. Babenko. Multiple instance learning: Algorithms and applications. *University of California, Notes*, 2009.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Multi-facet rating of product reviews. In Mohand Boughanem, Catherine Berrut, Josiane Mothe, and Chantal Soule-Dupuy, editors, *Advances in Information Retrieval*, volume 5478 of *Lecture Notes in Computer Science*, pages 461–472. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-00958-7_41.

Paul W. Ballantine, Yongjia Lin, and Ekant Veer. The influence of user comments on perceptions of facebook relationship status updates. *Computers in Human Behavior*, 49:50 – 55, 2015. ISSN 0747-5632. doi: 10.1016/j.chb.2015.02.055.

Robert M. Bell, Yehuda Koren, and Chris Volinsky. All together now: A perspective on the Netflix prize. *CHANCE*, 23:24–24, 2010. ISSN 0933-2480. doi: 10.1007/s00144-010-0005-2.

Chidansh Bhatt, Nikolaos Pappas, Maryam Habibi, and Andrei Popescu-Belis. Multimodal reranking of content-based recommendations for hyperlinking video snippets. In *Proceedings of International Conference on Multimedia Retrieval*, ICMR '14, pages 225:225–225:232, Glasgow, United Kingdom, 2014. doi: 10.1145/2578726.2578752.

Chidansh A. Bhatt, Nikolaos Pappas, Maryam Habibi, and Andrei Popescu-Belis. Idiap at MediaEval 2013: Search and hyperlinking task. In *MediaEval 2013 Workshop*, CEUR Workshop Proceedings, Barcelona, Spain, 2013a.

Chidansh Amitkumar Bhatt, Andrei Popescu-Belis, Maryam Habibi, Sandy Ingram, Stefano Masneri, Fergus McInnes, Nikolaos Pappas, and Oliver Schreer. Multi-factor segmentation for topic visualization and recommendation: the MUST-VIS system. In *Proceedings of the 21st ACM international conference on Multimedia*, MM '13, pages 365–368, Barcelona, Spain, 2013b. doi: 10.1145/2502081.2508120.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(30):993–1022, 2003. ISSN 1532-4435.

Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1 – 8, 2011. ISSN 1877-7503. doi: 10.1016/j.jocs.2010.12.007.

Damian Borth, Rongrong Ji, Tao Chen, Thomas Breuel, and Shih-Fu Chang. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, pages 223–232, Barcelona, Spain, 2013. ISBN 978-1-4503-2404-5. doi: 10.1145/2502081.2502282.

Samuel Brody and Noemie Elhadad. An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 804–812, Los Angeles, CA, USA, 2010.

Razvan C. Bunescu and Raymond J. Mooney. Multiple instance learning for sparse positive bags. In *Proceedings of the 24th Annual International Conference on Machine Learning*, ICML '07, Corvallis, OR, USA, 2007.

Robin Burke. Hybrid web recommender systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The adaptive web*, pages 377–408. Springer-Verlag, Berlin, Heidelberg, 2007.

Mauro Cettolo, Christian Girardi, and Marcello Federico. WIT^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation*, EAMT '12, pages 261–268, Trento, Italy, 2012.

Jean-Cédric Chappelier. Topic-based generative models for text information access. In *Textual Information Access*, pages 129–177. John Wiley and Sons, Inc, 2013. ISBN 9781118562796. doi: 10.1002/9781118562796.ch5.

Jean-Cédric Chappelier and Emmanuel Eckard. PLSI: The true Fisher kernel and beyond. In Wray Buntine, Marko Grobelnik, Dunja Mladenić, and John Shawe-Taylor, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 5781 of *Lecture Notes in Computer Science*, pages 195–210. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-04179-2. doi: 10.1007/978-3-642-04180-8_30.

Yan-Ying Chen, Tao Chen, Winston H. Hsu, Hong-Yuan Mark Liao, and Shih-Fu Chang. Predicting viewer affective comments based on image content in social media. In *Proceedings of International Conference on Multimedia Retrieval*, ICMR '14, pages 233:233–233:240, Glasgow, United Kingdom, 2014. ISBN 978-1-4503-2782-4. doi: 10.1145/2578726.2578756.

Yan-Ying Chen, Tao Chen, Taikun Liu, H.-Y.M. Liao, and Shih-Fu Chang. Assistive image comment robot - a novel mid-level concept-based representation. *IEEE Transactions on Affective Computing*, 6(3):298–311, 2015. ISSN 1949-3045. doi: 10.1109/TAFFC.2014.2388370.

Veronika Cheplygina, David M.J. Tax, and Marco Loog. Multiple instance learning with bag dissimilarities. *Pattern Recognition*, 48(1):264 – 275, 2015. ISSN 0031-3203. doi: 10.1016/j.patcog.2014.07.022.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273–297, 1995. ISSN 0885-6125. doi: 10.1023/A:1022627411411.

Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-N recommendation tasks. In *Proceedings of the 4th ACM Conference on Recommender Systems*, RecSys '10, Barcelona, Spain, 2010. doi: 10.1145/1864708.1864721.

Rafael M. D'Addio and Marcelo G. Manzato. A sentiment-based item description approach for knn collaborative filtering. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, SAC '15, pages 1060–1065, Salamanca, Spain, 2015. doi: 10.1145/2695664.2695747.

Stamatia Dasiopoulou, Vassilis Tzouvaras, Ioannis Kompatsiaris, and MichaelG. Strintzis. Enquiring MPEG-7 based multimedia ontologies. *Multimedia Tools and Applications*, 46 (2-3):331–370, 2010. ISSN 1380-7501. doi: 10.1007/s11042-009-0387-4.

Jesse Davis et al. Tightly integrating relational learning and multiple-instance regression for real-valued drug activity prediction. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 425–432, Corvallis, OR, USA, 2007.

Marco Degemmis, Pasquale Lops, and Giovanni Semeraro. A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction*, 17(3):217–255, 2007. ISSN 0924-1868. doi: 10.1007/s11257-006-9023-4.

Mukund Deshpande and George Karypis. Item-based top-N recommendation algorithms. *ACM Transactions on Information Systems*, 22(1):143–177, 2004. ISSN 1046-8188. doi: 10.1145/963770.963776.

Riccardo Di Massa, Maurizio Montagnuolo, and Alberto Messina. Implicit news recommendation based on user interest models and multimodal content analysis. In *Proceedings of the 3rd International Workshop on Automated Information Extraction in Media Production*, AIEMPro '10, pages 33–38, Firenze, Italy, 2010. doi: 10.1145/1877850.1877861.

Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 193–202, New York, NY, USA, 2014. doi: 10.1145/2623330.2623758.

Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1–2):31 – 71, 1997. ISSN 0004-3702. doi: 10.1016/S0004-3702(96)00034-3.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. Hate speech detection with comment embeddings. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 29–30, Florence, Italy, 2015. doi: 10.1145/2740908.2742760.

Gary Doran and Soumya Ray. A theoretical and empirical analysis of support vector machine methods for multiple-instance classification. *Machine Learning*, 97(1-2):79–102, 2014. ISSN 0885-6125. doi: 10.1007/s10994-013-5429-5.

Harris Drucker, Chris J.C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. In *Advances in Neural Information Processing systems*, pages 155–161, Denver, CO, USA, 1996.

Asmaa Elbadrawy and George Karypis. User-specific feature-based similarity models for top-N recommendation of new items. *ACM Transactions on Intelligent System and Technology*, 6 (3):33:1–33:20, 2015. ISSN 2157-6904. doi: 10.1145/2700495.

Joseph G. Ellis, Sabrina W. Lin, Ching-Yung Lin, and Shih-Fu Chang. Predicting evoked emotions in video. In *Proceedings of the IEEE International Symposium on Multimedia*, ISM '14, pages 287–294, 2014. doi: 10.1109/ISM.2014.69.

Siamak Faridani. Using canonical correlation analysis for generalized sentiment analysis, product recommendation and search. In *Proceedings of the 5th ACM Conference on Recommender Systems*, RecSys '11, pages 355–358, Chicago, Illinois, USA, 2011. doi: 10.1145/2043932.2044005.

Marcello Federico, Mauro Cettolo, Luisa Bentivogli, Michael Paul, and Sebastian Stüker. Overview of the IWSLT 2012 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation*, IWSLT '12, Hong Kong, China, 2012.

James Foulds and Eibe Frank. A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25:1:1–25, 2010.

Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI '07, pages 1606–1611, Hyderabad, India, 2007.

Gayatree Ganu, Noemie Elhadad, and Amélie Marian. Beyond the stars: Improving rating predictions using review text content. In *Proceedings of the 12th International Workshop on the Web and Databases*, WebDB '09, Providence, Rhode Island, 2009.

Miguel Á. García-Cumbreras, Arturo Montejo-Ráez, and Manuel C. Díaz-Galiano. Pessimists and optimists: Improving collaborative filtering through sentiment analysis. *Expert Systems with Applications*, 40(17):6758 – 6765, 2013. ISSN 0957-4174. doi: 10.1016/j.eswa.2013.06.049.

Narendra Gupta, Giuseppe Di Fabbrizio, and Patrick Haffner. Capturing the stars: Predicting ratings for service and product reviews. In *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*, SS '10, pages 36–43, Los Angeles, CA, USA, 2010.

Maryam Habibi and Andrei Popescu-Belis. Diverse keyword extraction from conversations. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics, Short Papers*, ACL '13, Sofia, Bulgaria, 2013.

Maryam Habibi and Andrei Popescu-Belis. Keyword extraction and clustering for document recommendation in conversations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(4):746–759, 2015. doi: 10.1109/TASLP.2015.2405482.

Hussam Hamdan, Patrice Bellot, and Frederic Bechet. Lsislif: CRF and logistic regression for opinion target extraction and sentiment polarity analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 753–758, Denver, Colorado, 2015.

Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, and Mauro Cettolo. Pronoun-focused mt and cross-lingual pronoun prediction: Findings of the 2015 discomt shared task on pronoun translation. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 1–16, Lisbon, Portugal, 2015.

Vasileios Hatzivassiloglou and Janyce M. Wiebe. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*, COLING '00, pages 299–305, Saarbrücken, Germany, 2000. doi: 10.3115/990820.990864.

Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. TriRank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1661–1670, Melbourne, VIC, Australia, 2015. doi: 10.1145/2806416.2806504.

Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22 (1):5–53, 2004. ISSN 1046-8188. doi: 10.1145/963770.963772.

Jonathan Lee Herlocker. *Understanding and Improving Automated Collaborative Filtering Systems*. PhD thesis, University of Minnesota, 2000. AAI9983577.

Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, Berkeley, CA, USA, 1999. doi: 10.1145/312624.312649.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge discovery and data mining*, KDD '04, pages 168–177, Seattle, WA, 2004. doi: 10.1145/1014052.1014073.

Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining*, ICDM '08, pages 263–272, Washington, DC, USA, 2008. doi: 10.1109/ICDM.2008.22.

Vidit Jain and Esther Galbrun. Topical organization of user comments and application to content recommendation. In *Proceedings of the 22nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 61–62, Rio de Janeiro, Brazil, 2013.

Niklas Jakob, Stefan Hagen Weber, Mark Christoph Müller, and Iryna Gurevych. Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations. In *Proceedings of the 1st International CIKM Workshop on Topic-sentiment Analysis for Mass Opinion*, TSA '09, pages 57–64, 2009. doi: 10.1145/1651461.1651473.

Thorsten Joachims. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 217–226, 2006. doi: 10.1145/1150402.1150429.

Pontus Johansson. Madfilm - a multimodal approach to handle search and organization in a movie recommendation system. In *Proceedings of the 1st Nordic Symposium on Multimodal Communication*, pages 53–65, Copenhagen, Denmark, 2003.

Brendan Jou, Tao Chen, Nikolaos Pappas, Miriam Redi, Mercan Topkara, and Shih-Fu Chang. Visual affect around the world: A large-scale multilingual visual sentiment ontology. In *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*, MM '15, pages 159–168, Brisbane, Australia, 2015. ISBN 978-1-4503-3459-4. doi: 10.1145/2733373.2806246.

Santosh Kabbur, Xia Ning, and George Karypis. FISM: factored item similarity models for top-N recommender systems. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 659–667, Chicago, Illinois, USA, 2013. doi: 10.1145/2487575.2487589.

Pythagoras Karampiperis, Antonis Koukourikos, and Giannis Stoitsis. Collaborative filtering recommendation of educational content in social environments utilizing sentiment analysis techniques. In Nikos Manouselis, Hendrik Drachsler, Katrien Verbert, and Olga C. Santos, editors, *Recommender Systems for Technology Enhanced Learning*, pages 3–23. Springer, Berlin, 2014. doi: 10.1007/978-1-4939-0530-0_1.

Noriaki Kawamae. Predicting future reviews: Sentiment analysis models for collaborative filtering. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 605–614, Hong Kong, China, 2011. doi: 10.1145/1935826.1935911.

Hyung W. Kim, Keejun Han, Yi Y. Mun, Joonmyun Cho, and Jinwoo Hong. MovieMine: personalized movie content search by utilizing user comments. *IEEE Transactions on Consumer Electronics*, 58(4):1416–1424, 2012. ISSN 0098-3063. doi: 10.1109/TCE.2012.6415015.

Hyunsoo Kim and Haesun Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23 (12):1495–1502, 2007. doi: 10.1093/bioinformatics/btm134.

Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 426–434, Las Vegas, NV, USA, 2008. doi: 10.1145/1401890.1401944.

Yehuda Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 447–456, Paris, France, 2009. doi: 10.1145/1557019.1557072.

Yehuda Koren and Robert Bell. Advances in collaborative filtering. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 145–186. Springer US, 2011. doi: 10.1007/978-0-387-85820-3_5.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. ISSN 0018-9162. doi: 10.1109/MC.2009.263.

Dimitrios Kotzias, Misha Denil, Nando de Freitas, and Padhraic Smyth. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 597–606, Sydney, NSW, Australia, 2015. doi: 10.1145/2783258.2783380.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *18th International Conference on Machine Learning*, pages 282–289, San Francisco, California, 2001.

John Lees-Miller, Fraser Anderson, Bret Hoehn, and Russell Greiner. Does Wikipedia information help Netflix predictions? In *Proceedings of the 7th International Conference on Machine Learning and Applications*, ICMLA '08, pages 337–343, San Diego, California, 2008.

Cane Leung, Stephen Chan, and Fu-Lai Chung. Integrating collaborative filtering and sentiment analysis: A rating inference approach. In *Proceedings of The ECAI 2006 Workshop on Recommender Systems*, pages 62–66, Riva del Garda, Italy, 2006.

Cane Leung, Stephen Chan, Fu-lai Chung, and Grace Ngai. A probabilistic rating inference framework for mining user preferences from reviews. *World Wide Web*, 14:187–215, 2011. ISSN 1386-145X.

Asher Levi, Osnat Mokryn, Christophe Diot, and Nina Taft. Finding a needle in a haystack of reviews: Cold start context-based hotel recommender system. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 115–122, Dublin, Ireland, 2012. doi: 10.1145/2365952.2365977.

Beibei Li, Shuting Xu, and Jun Zhang. Enhancing clustering blog documents by utilizing author/reader comments. In *Proceedings of the 45th Annual Southeast Regional Conference*, ACM-SE 45, pages 94–99, Winston-Salem, North Carolina, 2007. doi: 10.1145/1233341.1233359.

Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. Structure-aware review mining and summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 653–661, Beijing, China, 2010a.

Fangtao Li, Nathan Liu, Hongwei Jin, Kai Zhao, Qiang Yang, and Xiaoyan Zhu. Incorporating reviewer and product information for review rating prediction. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence - Volume 3*, IJCAI '11, pages 1820–1825, Barcelona, Spain, 2011. doi: 10.5591/978-1-57735-516-8/IJCAI11-305.

Qing Li, Jia Wang, Yuanzhu Peter Chen, and Zhangxi Lin. User comments for news recommendation in forum-based social media. *Journal of Information Science*, 180(24):4929–4939, 2010b. ISSN 0020-0255. doi: 10.1016/j.ins.2010.08.044.

Yanen Li, Jia Hu, Cheng Xiang Zhai, and Ye Chen. Improving one-class collaborative filtering by incorporating rich user information. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 959–968, Toronto, ON, Canada, 2010c. doi: 10.1145/1871437.1871559.

Yanen Li, ChengXiang Zhai, and Ye Chen. Exploiting rich user information for one-class collaborative filtering. *Knowledge and Information Systems*, 38(2):277–301, 2014. ISSN 0219-1377. doi: 10.1007/s10115-012-0583-9.

Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 375–384, Hong Kong, China, 2009. doi: 10.1145/1645953.1646003.

Guang Ling, Michael R. Lyu, and Irwin King. Ratings meet reviews, a combined approach to recommend. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 105–112, Foster City, CA, USA, 2014. doi: 10.1145/2645710.2645728.

Edward Loper and Steven Bird. NLTK: the natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, 2002. doi: 10.3115/1118108.1118117.

Pasquale Lops, Marco Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 73–105. Springer US, 2011. doi: 10.1007/978-0-387-85820-3_3.

Bin Lu, Myle Ott, Claire Cardie, and Benjamin K. Tsou. Multi-aspect sentiment analysis with topic models. In *Proceedings of the 11th IEEE International Conference on Data Mining Workshops*, ICDMW '11, pages 81–88, Washington, DC, USA, 2011. doi: 10.1109/ICDMW.2011.125.

Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the 4th ACM International Conference on Web search and Data Mining*, WSDM '11, pages 287–296, Hong Kong, China, 2011. doi: 10.1145/1935826.1935877.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 142–150, Portland, OR, USA, 2011.

Bernardo Magnini and Carlo Strapparava. Improving user modelling with content-based techniques. In Mathias Bauer, PiotrJ. Gmytrasiewicz, and Julita Vassileva, editors, *User Modeling 2001*, volume 2109 of *Lecture Notes in Computer Science*, pages 74–83. Springer, 2001.

Tariq Mahmood and Francesco Ricci. Improving recommender systems with adaptive conversational strategies. In *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*, HT '09, pages 73–82, Torino, Italy, 2009. doi: 10.1145/1557914.1557930.

José M. Martinez. Standards - MPEG-7 overview of MPEG-7 description tools, part 2. *Multi-Media, IEEE*, 9(3):83–93, 2002. ISSN 1070-986X. doi: 10.1109/MMUL.2002.1022862.

Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 165–172, Hong Kong, China, 2013. doi: 10.1145/2507157.2507163.

Julian McAuley, Jure Leskovec, and Dan Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In *Proceedings of the 12th IEEE International Conference on Data Mining*, ICDM '12, pages 1020–1025, Brussels, Belgium, 2012. doi: 10.1109/ICDM.2012.110.

Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th International Conference on the World Wide Web*, WWW '07, pages 171–180, Banff, AB, Canada, 2007. doi: 10.1145/1242572.1242596.

Tao Mei, Bo Yang, Xian-Sheng Hua, and Shipeng Li. Contextual video recommendation by multimodal relevance and user feedback. *ACM Transactions on Information Systems*, 29(2): 10:1–10:24, 2011. ISSN 1046-8188. doi: 10.1145/1961209.1961213.

Andrew Messenger and John Whittle. Recommendations based on user-generated comments in social media. In *Proceedings of the 3rd International Conference on Privacy, Security, Risk and Trust*, PASSAT '11, pages 505–508, Boston, USA, 2011. doi: 10.1109/PASSAT/SocialCom.2011.146.

Stuart E. Middleton, Nigel R. Shadbolt, and David C. De Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems*, 22(1):54–88, 2004. ISSN 1046-8188. doi: 10.1145/963770.963773.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In Christopher Burges, Léon Bottou, Max Welling, Zoubin Ghahramani, and Kilian Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119, 2013.

Yashar Moshfeghi, Benjamin Piwowarski, and Joemon M. Jose. Handling data sparsity in collaborative filtering using emotion and semantic based features. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 625–634, Beijing, China, 2011. doi: 10.1145/2009916.2010001.

Andreas C. Müller and Sven Behnke. pystruct - learning structured prediction in python. *Journal of Machine Learning Research*, 15:2055–2060, 2014.

Claudiu-Cristian Musat, Yizhong Liang, and Boi Faltings. Recommendation using textual opinions. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 2684–2690. AAAI Press, 2013.

Nathan Srebro Nati and Tommi Jaakkola. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning*, ICML '03, pages 720–727, Washington, DC, USA, 2003.

YanPing Nie, Yang Liu, and Xiaohui Yu. Weighted aspect-based collaborative filtering. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 1071–1074, Gold Coast, QLD, Australia, 2014. doi: 10.1145/2600428.2609512.

Xia Ning and George Karypis. SLIM: Sparse linear methods for top-N recommender systems. In *Proceedings of the 11th International Conference on Data Mining*, ICDM '11, pages 497–506, Vancouver, BC, Canada, 2011.

Rong Pan and Martin Scholz. Mind the gaps: Weighting the unknown in large-scale one-class collaborative filtering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 667–676, Paris, France, 2009. doi: 10.1145/1557019.1557094.

Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *8th IEEE International Conference on Data Mining*, ICDM '08, pages 502–511, Pisa, Italy, 2008. doi: 10.1109/ICDM.2008.16.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 115–124, Ann Arbor, Michigan, 2005. doi: 10.3115/1219840.1219855.

Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008. ISSN 1554-0669. doi: 10.1561/1500000011.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '02, pages 79–86, Philadelphia, PA, USA, 2002. doi: 10.3115/1118693.1118704.

Manos Papagelis and Dimitris Plexousakis. Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents. *Engineering Applications of Artificial Intelligence*, 18(7):781–789, 2005. ISSN 0952-1976. doi: 10.1016/j.engappai.2005.06.010.

Nikolaos Pappas and Andrei Popescu-Belis. Combining content with user preferences for ted lecture recommendation. In *11th International Workshop on Content-Based Multimedia Indexing*, CBMI '13, pages 47–52, 2013a. doi: 10.1109/CBMI.2013.6576551.

Nikolaos Pappas and Andrei Popescu-Belis. Sentiment analysis of user comments for one-class collaborative filtering over TED talks. In *Proceedings of the 36th international ACM SIGIR Conference on Research and development in information retrieval*, SIGIR '13, pages 773–776, Dublin, Ireland, 2013b. doi: 10.1145/2484028.2484116.

Nikolaos Pappas and Andrei Popescu-Belis. Explaining the stars: Weighted multiple-instance learning for aspect-based sentiment analysis. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP '14, pages 455–466, Doha, Qatar, 2014.

Nikolaos Pappas and Andrei Popescu-Belis. Combining content with user preferences for non-fiction multimedia recommendation: a study on ted lectures. *Multimedia Tools and Applications*, pages 1–23, 2015. ISSN 1380-7501. doi: 10.1007/s11042-013-1840-y.

Nikolaos Pappas and Andrei Popescu-Belis. Adaptive sentiment-aware one-class collaborative filtering. *Expert Systems with Applications*, 43:23 – 41, 2016a. ISSN 0957-4174. doi: 10.1016/j.eswa.2015.08.035.

Nikolaos Pappas and Andrei Popescu-Belis. Learning aspect saliency of sentences for review understanding. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics (Submitted)*, San Diego, CA, USA, 2016b.

Nikolaos Pappas, Georgios Katsimpras, and Efstathios Stamatatos. Distinguishing the popularity between topics: A system for up-to-date opinion retrieval and mining in the web. In *Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics*, CICLING '13, pages 197–209, 2013. doi: 10.1145/2362456.2362462.

Ulrich Paquet and Noam Koenigstein. One-class collaborative filtering with random graphs. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pages 999–1008, Rio de Janeiro, Brazil, 2013.

Souneil Park, Minsam Ko, Jungwoo Kim, Ying Liu, and Junehwa Song. The politics of comments: Predicting political orientation of news stories with commenters' sentiment patterns. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, CSCW '11, pages 113–122, Hangzhou, China, 2011. doi: 10.1145/1958824.1958842.

Braja Gopal Patra, Soumik Mandal, Dipankar Das, and Sivaji Bandyopadhyay. JU_CSE: A conditional random field (CRF) based approach to aspect based sentiment analysis. In

Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval '14, pages 370–374, Dublin, Ireland, 2014.

Paul A. Pavlou and Angelika Dimoka. The nature and role of feedback text comments in online marketplaces: Implications for trust building, price premiums, and seller differentiation. *Information Systems Research*, 17(4):392–414, 2006. ISSN 1526-5536. doi: 10.1287/isre.1060.0106.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in python. *CoRR*, abs/1201.0490, 2012.

Stefan Pero and Tomáš Horváth. Opinion-driven matrix factorization for rating prediction. In Sandra Carberry, Stephan Weibelzahl, Alessandro Micarelli, and Giovanni Semeraro, editors, *User Modeling, Adaptation, and Personalization*, volume 7899 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 2013. doi: 10.1007/978-3-642-38844-6_1.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation*, SemEval '14, pages 27–35, Dublin, Ireland, 2014.

Lizhen Qu, Georgiana Ifrim, and Gerhard Weikum. The bag-of-opinions method for review rating prediction from sparse text patterns. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 913–921, Beijing, China, 2010.

Soumya Ray and David Page. Multiple instance regression. In *Proceedings of the 18th International Conference on Machine Learning*, ICML '01, pages 425–432, 2001.

Radim Řehůřek and Petr Sojka. Software framework for topic modeling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, 2010.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Montreal, Quebec, Canada, 2009.

Paul Resnick and Hal R. Varian. Recommender systems. *Communications of the ACM*, 40(3): 56–58, 1997. ISSN 0001-0782. doi: 10.1145/245108.245121.

Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. *Recommender Systems Handbook*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

Anthony Rousseau, Paul Deléglise, and Yannick Estève. TED-LIUM: an automatic speech recognition dedicated corpus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, LREC '12, Istanbul, Turkey, 2012.

Magnus Sahlgren. An introduction to random indexing. In *Proceedings of the 7th International Conference on Terminology and Knowledge Engineering, Methods and Applications of Semantic Indexing Workshop*, volume 5, Copenhagen, Denmark, 2005.

Magnus Sahlgren. *The word-space model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces.* PhD thesis, Stockholm University, Stockholm, Sweden, 2006.

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988. ISSN 0306-4573. doi: 10.1016/0306-4573(88)90021-0.

Jose San Pedro, Tom Yeh, and Nuria Oliver. Leveraging user comments for aesthetic aware image search reranking. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 439–448, Lyon, France, 2012. doi: 10.1145/2187836.2187896.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, Hong Kong, China, 2001. doi: 10.1145/371920.372071.

Christina Sauper and Regina Barzilay. Automatic aggregation by joint modeling of aspects and values. *Journal of Artificial Intelligence Research*, 46(1):89–127, 2013. ISSN 1076-9757.

Christina Sauper, Aria Haghighi, and Regina Barzilay. Incorporating content structure into text analysis applications. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 377–387, Cambridge, MA, 2010.

J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 291–324. Springer Berlin Heidelberg, 2007. doi: 10.1007/978-3-540-72079-9_9.

Ingo Schwab, Wolfgang Pohl, and Ivan Koychev. Learning to recommend from positive evidence. In *Proceedings of the 5th International Conference on Intelligent User Interfaces*, IUI '00, pages 241–247, New York, NY, USA, 2000. ACM. doi: 10.1145/325737.325858.

Giovanni Semeraro, Marco Degemmis, Pasquale Lops, and Pierpaolo Basile. Combining learning and word sense disambiguation for intelligent user profiling. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2856–2861, Hyderabad, India, 2007.

Giovanni Semeraro, Pierpaolo Basile, Marco De Gemmis, and Pasquale Lops. User profiles for personalizing digital libraries. In Y. Theng, S. Foo, D. Goh, and Na J., editors, *Handbook of research on digital libraries design development and impact*, pages 149–158. Information Science Reference, 2009a.

Giovanni Semeraro, Pasquale Lops, Pierpaolo Basile, and Marco de Gemmis. Knowledge infusion into content-based recommender systems. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 301–304, New York, NY, USA, 2009b. doi: 10.1145/1639714.1639773.

Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *Advances in Neural Information Processing Systems*, NIPS '08, pages 1289–1296, Vancouver, BC, 2008.

Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 257–297. Springer US, 2011. doi: 10.1007/978-0-387-85820-3_8.

Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Nuria Oliver, and Alan Hanjalic. CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering. In *Proceedings of the 6th ACM Conference on Recommender Systems*, RecSys '12, pages 139–146, Dublin, Ireland, 2012. doi: 10.1145/2365952.2365981.

Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, and Alan Hanjalic. xCLiMF: optimizing expected reciprocal rank for data with multiple levels of relevance. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 431–434, Hong Kong, China, 2013. doi: 10.1145/2507157.2507227.

Hyoseop Shin, Minsoo Lee, and Eun Kim. Personalized digital TV content recommendation with integration of user behavior profiling and multimodal content rating. *IEEE Transactions on Consumer Electronics*, 55(3):1417–1423, 2009. ISSN 0098-3063. doi: 10.1109/TCE.2009.5278008.

Erez Shmueli, Amit Kagian, Yehuda Koren, and Ronny Lempel. Care to comment?: Recommendations for commenting on news stories. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 429–438, 2012. doi: 10.1145/2187836.2187895.

Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl, and Jose San Pedro. How useful are your comments?: Analyzing and predicting youtube comments and comment ratings. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 891–900, New York, NY, USA, 2010. ACM. doi: 10.1145/1772690.1772781.

V. Sindhwani, S. S. Bucak, J. Hu, and A. Mojsilovic. A family of non-negative matrix factorizations for one-class collaborative filtering problems. In *Proceedings of the 3rd ACM Conference on Recommender Systems, Recommender Based Industrial Applications Workshop*, RecSys '09, New York, NY, USA, 2009.

Vivek Kumar Singh, Mousumi Mukherjee, and Ghanshyam Kumar Mehta. Combining collaborative filtering and sentiment classification for improved movie recommendations. In Chattrakul Sombattheera, Arun Agarwal, Siba K. Udgata, and Kittichai Lavangnananda, editors, *Multi-disciplinary Trends in Artificial Intelligence*, volume 7080 of *Lecture Notes in Computer Science*, pages 38–50. Springer, Berlin/Heidelberg, 2011. doi: 10.1007/978-3-642-25725-4_4.

Alexander V. Smirnov and Andrew Krizhanovsky. Information filtering based on Wiki index database. *CoRR*, abs/0804.2354, 2008.

Benjamin Snyder and Regina Barzilay. Multiple aspect ranking using the good grief algorithm. In *In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '07, pages 300–307, Rochester, NY, USA, 2007.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 151–161, Edinburgh, United Kingdom, 2011.

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '13, pages 1631–1642, Portland, OR, USA, 2013.

Jianshan Sun, Gang Wang, Xusen Cheng, and Yelin Fu. Mining affective text to improve social media item recommendation. *Information Processing and Management*, 2014. ISSN 0306-4573. doi: 10.1016/j.ipm.2014.09.002.

Oscar Täckström and Ryan McDonald. Discovering fine-grained sentiment with latent variable structured prediction models. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, ECIR'11, pages 368–374, Berlin, Heidelberg, 2011. Springer-Verlag.

Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Investigation of various matrix factorization methods for large recommender systems. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, NETFLIX '08, pages 6:1–6:8, Las Vegas, NV, USA, 2008. doi: 10.1145/1722149.1722155.

Duyu Tang. Sentiment-specific representation learning for document-level sentiment analysis. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 447–452, Shanghai, China, 2015. doi: 10.1145/2684822.2697035.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, ACL '14, pages 1555–1565, Baltimore, MD, USA, 2014.

Alexandru Tatar, Jérémie Leguay, Panayotis Antoniadis, Arnaud Limbourg, Marcelo Dias de Amorim, and Serge Fdida. Predicting the popularity of online articles based on user comments. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, WIMS '11, pages 67:1–67:8, New York, NY, USA, 2011. ACM. doi: 10.1145/1988688.1988766.

Matt Thomas, Bo Pang, and Lillian Lee. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 327–335, Sydney, NSW, Australia, 2006.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, ICDEW '07, pages 801–810, Washington, DC, USA, 2007. IEEE Computer Society. doi: 10.1109/ICDEW.2007.4401070.

Ivan Titov and Ryan Mcdonald. A Joint Model of Text and Aspect Ratings for Sentiment Summarization. In *Proceedings of ACL-08: HLT*, pages 308–316, Columbus, OH, USA, 2008.

Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 111–120, Beijing, China, 2008. doi: 10.1145/1367497.1367513.

Manos Tsagkias, Wouter Weerkamp, and Maarten de Rijke. Predicting the volume of comments on online news stories. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 1765–1768, New York, NY, USA, 2009. ACM. doi: 10.1145/1645953.1646225.

Chrisa Tsinaraki and Stavros Christodoulakis. A multimedia user preference model that supports semantics and its application to MPEG 7/21. In *Proceedings of the 12th International Conference on Multi-Media Modelling*, page 8, Beijing, China, 2006. doi: 10.1109/MMMC.2006.1651299.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005. ISSN 1532-4435.

Peter D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Philadelphia, PA, 2002. doi: 10.3115/1073083.1073153.

Kiri L. Wagstaff and Terran Lane. Salience assignment for multiple-instance regression. In *ICML 2007 Workshop on Constrained Optimization and Structured Output Spaces*, Corvallis, OR, USA, 2007.

Kiri L. Wagstaff, Terran Lane, and Alex Roper. Multiple-instance regression with structured data. In *Proceedings of the IEEE International Conference on Data Mining Workshops*, ICDMW '08, pages 291–300, 2008. doi: 10.1109/ICDMW.2008.31.

Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 783–792, Washington, DC, USA, 2010a. doi: 10.1145/1835804.1835903.

Hua Wang, Feiping Nie, and Heng Huang. Learning instance specific distance for multi-instance classification. In *AAAI Conference on Artificial Intelligence*, 2011.

Jia Wang, Qing Li, and Yuanzhu Peter Chen. User comments for news recommendation in social media. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and development in information retrieval*, SIGIR '10, pages 881–882, 2010b. doi: 10.1145/1835449.1835663.

Jia Wang, Qing Li, Yuanzhu Peter Chen, and Zhangxi Lin. Recommendation in Internet forums and blogs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 257–265, Uppsala, Sweden, 2010c.

Zhuang Wang, Vladan Radosavljevic, Bo Han, Zoran Obradovic, and Slobodan Vucetic. Aerosol optical depth prediction from satellite observations by multiple instance regression. In *Proceedings of the SIAM International Conference on Data Mining*, SDM '08, pages 165–176, Atlanta, GA, USA, 2008. doi: 10.1137/1.9781611972788.15.

Zhuang Wang, Liang Lan, and S. Vucetic. Mixture model for multiple instance regression and applications in remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 50(6): 2226–2237, 2012. ISSN 0196-2892. doi: 10.1109/TGRS.2011.2171691.

Janyce Wiebe, Theresa Wilson, Rebecca Bruce, Matthew Bell, and Melanie Martin. Learning subjective language. *Computational Linguistics*, 30(3):277–308, 2004. ISSN 0891-2017. doi: 10.1162/0891201041850885.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Vancouver, BC, Canada, 2005. doi: 10.3115/1220575.1220619.

Yao Wu and Martin Ester. FLAME: A probabilistic model combining aspect based opinion mining and collaborative filtering. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 199–208, Shanghai, China, 2015. doi: 10.1145/2684822.2685291.

Bo Yang, Tao Mei, Xian-Sheng Hua, Linjun Yang, Shi-Qiang Yang, and Mingjing Li. Online video recommendation based on multimodal fusion and relevance feedback. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, CIVR '07, pages 73–80, Amsterdam, The Netherlands, 2007. doi: 10.1145/1282280.1282290.

Dawei Yin, Shengbo Guo, Boris Chidlovskii, Brian D. Davison, Cédric Archambeau, and Guillaume Bouchard. Connecting comments and tags: Improved modeling of social tagging systems. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 547–556, Rome, Italy, 2013. doi: 10.1145/2433396.2433466.

Ting Yuan, Jian Cheng, Xi Zhang, Qinshan Liu, and Hanqing Lu. A weighted one class collaborative filtering with content topic features. In Shipeng Li, Abdulmotaleb Saddik, Meng Wang, Tao Mei, Nicu Sebe, Shuicheng Yan, Richang Hong, and Cathal Gurrin, editors, *Advances in Multimedia Modeling*, volume 7733 of *Lecture Notes in Computer Science*, pages 417–427. Springer Berlin Heidelberg, 2013. doi: 10.1007/978-3-642-35728-2_40.

Min-Ling Zhang and Zhi-Hua Zhou. M3MIML: A maximum margin method for multi-instance multi-label learning. In *Proceedings of the 8th IEEE International Conference on Data Mining*, ICDM '08, pages 688–697, 2008. doi: 10.1109/ICDM.2008.27.

Min-Ling Zhang and Zhi-Hua Zhou. Multi-instance clustering with applications to multi-instance prediction. *Applied Intelligence*, 31(1):47–68, 2009. ISSN 0924-669X. doi: 10.1007/s10489-007-0111-x.

Rong Zhang, Yifan Gao, Wenzhe Yu, Pingfu Chao, Xiaoyan Yang, Ming Gao, and Aoying Zhou. Review comment analysis for predicting ratings. In Jian Li and Yizhou Sun, editors, *Web-Age Information Management*, volume 9098 of *Lecture Notes in Computer Science*, pages 247–259. Springer, Berlin, 2015. doi: 10.1007/978-3-319-21042-1_20.

Weishi Zhang, Guiguang Ding, Li Chen, and Chunping Li. Augmenting Chinese online video recommendations by using virtual ratings predicted by review sentiment classification. In *Proceedings of the 2010 IEEE International Conference on Data Mining Workshops*, ICDMW '10, pages 1143–1150, Sydney, NSW, Australia, 2010. doi: 10.1109/ICDMW.2010.27.

Weishi Zhang, Guiguang Ding, Li Chen, Chunping Li, and Chengbo Zhang. Generating virtual ratings from Chinese reviews to augment online recommendations. *ACM Transactions on Intelligent Systems and Technology*, 4(1):9:1–9:17, 2013. ISSN 2157-6904. doi: 10.1145/2414425.2414434.

Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 83–92, Gold Coast, QLD, Australia, 2014. doi: 10.1145/2600428.2609579.

Zhi-Hua Zhou, Kai Jiang, and Ming Li. Multi-instance learning based web mining. *Applied Intelligence*, 22(2):135–147, 2005. ISSN 0924-669X. doi: 10.1007/s10489-005-5602-z.

Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. Multi-instance learning by treating instances as non-i.i.d. samples. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1249–1256, Montreal, QC, Canada, 2009. doi: 10.1145/1553374.1553534.

Jingbo Zhu, Huizhen Wang, Benjamin K. Tsou, and Muhua Zhu. Multi-aspect opinion polling from textual reviews. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 1799–1802, Hong Kong, China, 2009. doi: 10.1145/1645953.1646233.

Jingbo Zhu, Chunliang Zhang, and Matthew Y. Ma. Multi-aspect rating inference with aspect-based segmentation. *IEEE Trans. on Affective Computing*, 3(4):469–481, 2012. ISSN 1949-3045. doi: 10.1109/T-AFFC.2012.18.

Jun-Yan Zhu, Jiajun Wu, Yan Xu, E. Chang, and Zhuowen Tu. Unsupervised object class discovery via saliency-guided multiple class learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(4):862–875, 2015. ISSN 0162-8828. doi: 10.1109/TPAMI.2014.2353617.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie review mining and summarization. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, pages 43–50, Arlington, VA, USA, 2006. doi: 10.1145/1183614.1183625.

Marinka Zitnik and Blaž Zupan. NIMFA: A Python library for nonnegative matrix factorization. *Journal of Machine Learning Research*, 13:849–853, 2012. ISSN 1532-4435.

Nikolaos Pappas

Ph.D. Student

Rue Marconi 19
Martigny 1920, Switzerland
📱 *+41 774968189*
☎ *+41 277205814*
✉ *nikolaos.pappas@idiap.ch*
🖰 *http://people.idiap.ch/npappas*

Research Interests

Natural Language Processing, Machine Learning, Recommender Systems

Education

2012–2016 **Ph.D. Electrical Engineering Doctoral program (EDEE)**.
Ecole Polytechnique Fédérale de Lausanne, Switzerland

Thesis Learning explainable user sentiment and preferences for information filtering

2009–2011 **M.Sc. Information Management**, *GPA: 9.53/10.*
University of the Aegean, Samos, Greece

Thesis Sentiment analysis of user-generated content from topic-specific Web sources

2003–2009 **Dipl. Eng. Information and Communication Systems**, *GPA: 8.09/10.*
University of the Aegean, Samos, Greece

Thesis A semantic wiki based ontology argumentation system

Professional Experience

Jun–Dec 2015 **Visiting Scholar**, *DVMM laboratory, Columbia University*, New York, NY, USA.

2012–2016 **Research Assistant**, *NLP group, Idiap Research Institute*, Martigny, Switzerland.

2009–2011 **Informatics Engineer**, *I-Sieve*, Athens, Greece.

2007–2011 **Software Engineer**, *Endysis Group*, Athens, Greece.

Mar–Jul 2007 **Software Developer**, *Optimor Labs*, Oxford, England.

Publications

1. **Nikolaos Pappas**, Andrei Popescu-Belis, "Adaptive Sentiment-Aware One-Class Collaborative Filtering: Application to Multimedia Datasets", *Expert Systems with Applications*, 43(1): 23-41, 2015.
2. **Nikolaos Pappas**, Andrei Popescu-Belis, "Combining Content with User Preferences for Non-Fiction Multimedia Recommendation: A Study on TED Lectures", *Multimedia Tools and Applications*, Special Issue on Content-Based Multimedia Indexing, 74(4): 1175-1197, 2014.
3. **Nikolaos Pappas**, Andrei Popescu-Belis, "Learning Aspect Saliency of Sentences for Review Understanding", Submitted to the *15th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2016.
4. **Nikolaos Pappas**, Andrei Popescu-Belis, "Explaining the Stars: Weighted Multiple-Instance Learning for Aspect-Based Sentiment Analysis", In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 455–466, Doha, Qatar, 2014.

5. **Nikolaos Pappas**, Andrei Popescu-Belis, “Sentiment Analysis of User Comments for One-Class Collaborative Filtering over TED Talks”, In *Proceedings of 36th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 773–776, Dublin, Ireland, 2013.
6. **Nikolaos Pappas**, Andrei Popescu-Belis, “Combining Content with User Preferences for TED Lecture Recommendation”, In *Proceedings of 11th International Workshop on Content Based Multimedia Indexing*, pages 47–52, Veszprém, Hungary, 2013.
7. Brendan Jou, Tao Chen,**Nikolaos Pappas**, Miriam Redi, Mercan Topkara and Shih-Fu Chang, “Visual Affect Around the World: A Large-scale Multilingual Visual Sentiment Ontology”, In *Proceedings of ACM International Conference on Multimedia*, Brisbane, Australia, 2015.
8. Chidansh A. Bhatt, **Nikolaos Pappas**, Maryam Habibi and Andrei Popescu-Belis, “Multimodal Reranking of Content-based Recommendations for Hyperlinking Video Snippets”, In *Proceedings of ACM International Conference on Multimedia Retrieval*, pages 225–232, Glasgow, UK, 2014.
9. Chidansh A. Bhatt, Andrei Popescu-Belis, Maryam Habibi, Sandy Ingram, Stefano Masneri, Fergus McInnes, **Nikolaos Pappas**, Oliver Schreer, “Multi-factor Segmentation for Topic Visualization and Recommendation: the MUST-VIS System”, In *Proceedings of 21st ACM International Conference on Multimedia*, pages 365–368, Barcelona, Spain, 2013
10. Chidansh A. Bhatt, **Nikolaos Pappas**, Maryam Habibi and Andrei Popescu-Belis, “Idiap at MediaEval 2013: Search and Hyperlinking Task”, In *Proceedings of MediaEval 2013 Workshop*, Barcelona, Spain, 2013.
11. Hervé Bourlard, Marc Ferras, **Nikolaos Pappas**, Andrei Popescu-Belis, Steve Renals, Fergus McInnes, Peter Bell, Sandy Ingram and Maël Guillemot, “Processing and Linking Audio Events in Large Multimedia Archives: The EU inEvent Project”, In *Proceedings of 1st Workshop on Speech, Language and Audio in Multimedia, InterSpeech satellite event*, Marseille, France, 2013.
12. **Nikolaos Pappas**, Georgios Katsimpras, Efstathios Stamatatos, “Distinguishing the Popularity Between Topics: A System for Up-to-date Opinion Retrieval and Mining in the Web", In *Proceedings of 14th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 197–209, Samos, Greece, 2013.
13. **Nikolaos Pappas**, Georgios Katsimpras, Efstathios Stamatatos, “An Agent-Based Focused Crawling Framework for Topic- and Genre-Related Web Document Discovery", In *Proceedings of 24th IEEE International Conference on Tools with Artificial Intelligence*, pages 508–515, Athens, Greece, 2012.
14. **Nikolaos Pappas**, Georgios Katsimpras, Efstathios Stamatatos, “Extracting Informative Textual Parts from Web Pages Containing User-Generated Content", In *Proceedings of 12th International Conference on Knowledge Management and Knowledge Technologies*, pages 4:1–4:8, Graz, Austria, 2012.
15. Konstantinos Kotis, Andreas Papasalouros, George Vouros, **Nikolaos Pappas** Konstantinos Zoumpatianos, “Enhancing the Collective Knowledge for the Engineering of Ontologies in Open and Socially Constructed Learning Spaces”, *Journal of Universal Computer Science*, pages 1710–1742, 2011.

Awards and Honors

2015 **Best paper award "Adaptive Sentiment-aware One-Class Collaborative Filtering"**.
Idiap Research Institute

2013 **Ranked first on video hyperlinking task (with Idiap NLP group)**.
MediaEval 2013 Workshop

2013 **Winner on video lecture annotation and search challenge (with Idiap NLP group)**.
ACM MM Conference 2013

2010 **Award for top graduates of Greek Polytechnic Schools (2009 – 2010)**.
TEE (Technical Chamber of Greece)

2006 **Scholarship for the third academic year (top 1%)**.
IKY (Greek State Scholarships Foundation)

2004 **Scholarship for the first academic year (top 1.5%)**.
IKY (Greek State Scholarships Foundation)

Projects

2015 **HYBRID - Hybrid Recommender System for a University Social Network**.

Hybrid is a project funded by TheArc which aims at transferring scientific know-how in the domain of search and recommendation from Idiap to Unono to improve existing or create new recommendation functions of news and jobs in their university social network created.

2014 **AROLES - Automatic Recommendation of Lectures and Snipets**.

AROLES is a project funded by SNSF which aims at transferring scientific know-how in the domains of audio-visual processing and multimedia retrieval from the IM2 NCCR to the Klewel SME dedicated to lecture capture and web-based broadcasting.

2012–2014 **inEvent - Accessing Dynamic Networked Multimedia Events**.

inEvent FP7-ICT n. 287872 is a project funded by the European Union. The main goal of inEvent is to develop new means to structure, retrieve, and share large archives of networked, and dynamically changing, multimedia recordings, mainly consisting of meetings, video-conferences, and lectures.

Languages (proficiency)

Greek	**Native**	
English	**Professional working**	*First certificate in English (University of Cambridge)*
French	**Limited working**	*Certificat de Langue Française (Institut Français de Thessalonique)*

Computer skills

Programming	Python, Ruby, Java, C, C++, C Shell, Perl, Prolog, PHP, Javascript, JSP, Octave, Matlab.
Visualizing	Matplotlib, Gnuplot, Seaborn, Vincent, D3.js, neo4j, Gephi, OmniGraffle
OS	Linux, Mac OS X, Windows
Misc	Subversion, Mercurial, Git, Vi, Emacs, Apache Web Server, Nginx, Mongrel, Django, ROR, MySQL, MongoDB

Software and Resources

Code	`http://github.com/nik0spapp/`
Datasets	`http://www.idiap.ch/dataset/ted/`
	`http://mvso.cs.columbia.edu/`

January 14, 2016