AIX-MARSEILLE UNIVERSITÉ
ÉCOLE DOCTORALE 184

LIS UMR 7020 - R2I
Cléo UMS CNRS 3287 - OpenEdition

Thèse présentée pour obtenir le grade universitaire de docteur

Discipline: Informatique

Amal HTAIT

Thesis: Sentiment Analysis at the Service of Book Search.

Titre de la thèse : Analyse de Sentiment au Service de la Recherche de Livres.

Soutenue le 05/07/2019 devant le jury composé de:

Antoine DOUCET	Pr., Université de La Rochelle	Rapporteur
Karen PINEL-SAUVAGNAT	MCF HDR, Université Paul Sabatier	Rapporteure
Gabriella PASI	Pr., Université de Milano-Bicocca, Italie	Examinatrice
Lorraine GOEURIOT	MCF, Université Grenoble Alpes	Examinatrice
Patrice BELLOT	Pr., Université Aix-Marseille	Directeur de thèse
Sébastien FOURNIER	MCF, Université Aix-Marseille	Co-directeur de thèse

v0.1 2019-07-14
This work has been supported by the French State, managed by the National Research Agency under the "Investissements d'avenir" program under the EquipEx DILOH project (ANR-11-EQPX-0013).
Comments, corrections, and other remarks are welcome to:
amal.htait@univ-amu.fr
Laboratoire d'Informatique et Systèmes, LIS UMR 7020 CNRS,
Université Aix-Marseille,
Batiment Polytech,
Avenue Escadrille Normandie-Niemen,
13397 MARSEILLE CEDEX 20

Acknowledgments

I would like to thank the members of the jury: Pr. Antoine Doucet and Dr. Karen Pinel-Sauvagnat, for accepting to review the manuscript and for their precious remarks, as well as Pr. Gabriella Pasi and Dr. Lorraine Goeuriot, who participated in this thesis defense as examiners. A special additional thank you to Pr. Pasi for welcoming me last summer in a research visit at IR Lab, Milan, and for supporting me during the last year of the thesis.

I would like to express my sincere appreciation to my supervisor Pr. Patrice Bellot and my co-supervisor Dr. Sébastien Fournier for their guidance, encouragement, and mostly for entrusting me with the responsibility of taking the right initiatives throughout the thesis.

I thank my lab colleagues of doctoral students, researchers, teachers, engineers and staff, that I worked with during my thesis in LIS lab and OpenEdition Lab, it was a great pleasure.

Last but not least, I would like to thank my loving family: my parents, my brothers and especially my sister Rabab and her family, for their moral support and encouragement.

Abstract

The web technology is in an ongoing growth, and a huge volume of data is generated in the social web, where users would exchange a variety of information. In addition to the fact that social web text may be rich of information, the writers are often guided by provoked sentiments reflected in their writings. Based on that concept, locating sentiment in a text can play an important role for information extraction.

The purpose of this thesis is to improve the book search and recommendation quality of the OpenEdition's multilingual Books platform. The Books platform also offers additional information through users generated information (e.g. book reviews) connected to the books and rich in emotions expressed in the users' writings. Therefore, the previous analysis, concerning locating sentiment in a text for information extraction, plays an important role in this thesis, and can serve the purpose of quality improvement concerning book search, using the shared users generated information. Accordingly, we choose to follow a main path in this thesis to combine Sentiment Analysis (SA) and Information Retrieval (IR) fields, for the purpose of improving the quality of book search. Even thought the successful contribution of SA in different fields, but it has a limited contribution in the IR and search domain, for that reason, we propose new uses of SA in IR and in the book retrieval field. Two objectives are summarised in the following, which serve the main purpose of the thesis in the IR quality improvement using SA:

- An approach for SA prediction, easily applicable on different languages, low cost in time and annotated data.
- New approaches for book search quality improvement, based on SA employment in information filtering, retrieving and classifying.

To reach these objectives, we propose a semi-supervised method for sentiment intensity prediction, on words level, based on adapted to domain seed-words lexicons and word embeddings models. Within the proposed SA method, we suggest two methods for the seed-words' extraction. The proposed SA method serves next as the axis of two book search quality improvement propositions:

- A pseudo relevance feedback's method, where SA assists in the information extraction from social web resources of retrieved books.

- A classification of sentences in very long Book Search queries, where we analyse the SA role in this classification.

Furthermore, to improve the book recommendation quality of OpenEdition's multilingual Books platform, we propose a method to extract documents' bibliographical zone, as a pre-step for a book recommendation method based on an inter documents citation, tested in several languages.

In addition, we expand this thesis horizon by an additional proposition that serves indirectly one of the main objective of the thesis, the book search quality improvement. We present a method for an automatic creation of normalisation thesaurus used to decrease the difficulties caused by the social web's informal language and to improve the sentiment prediction.

Keywords: Sentiment Analysis, Sentiment Intensity, Information Retrieval, Book Search, Word Embedding, Seed-words, Pseudo relevance feedback, Language Model.

Résumé

Le Web est en croissance continue, et une quantité énorme de données est générée par les réseaux sociaux, permettant aux utilisateurs d'échanger une grande diversité d'informations. En outre, les textes au sein des réseaux sociaux sont souvent subjectifs. L'exploitation de cette subjectivité présente au sein des textes peut être un facteur important lors d'une recherche d'information.

Plus particulièrement, cette thèse est réalisée pour répondre aux besoins de la plate-forme Books de OpenEdition[1] en matière d'amélioration de la recherche et la recommandation de livres, en plusieurs langues. La plateforme offre des informations générées par des utilisateurs (par exemple les comptes rendus des livres), riches en sentiments. Par conséquent, l'analyse précédente, concernant l'exploitation de sentiment en recherche d'information, joue un rôle important dans cette thèse et peut servir l'objectif d'une amélioration de qualité de la recherche de livres en utilisant les informations générées par les utilisateurs. Par conséquent, nous avons choisi de suivre une voie principale dans cette thèse consistant à combiner les domaines Analyse de Sentiment (AS) et Recherche d'Information (RI), dans le but d'améliorer les suggestions de la recherche de livres. Malgré sa contribution fructueuse dans différents domaines, AS a une contribution limitée dans le domaine des RI et de la recherche de livres. Pour cette raison, nous proposons de nouvelles utilisations de AS en RI et dans le domaine de la recherche de livres.

Nos objectifs peuvent être résumés en plusieurs points:

- Une approche d'analyse de sentiment, facilement applicable sur différentes langues, peu coûteuse en temps et en données annotées.
- De nouvelles approches pour l'amélioration de la qualité lors de la recherche de livres, basées sur l'utilisation de l'analyse de sentiment dans le filtrage, l'extraction et la classification des informations.

Pour atteindre ces objectifs, nous proposons une méthode semi-supervisée de prédiction de l'intensité des sentiments, au niveau des mots, basée sur des lexiques de mots-germes et des modèles de plongement de mots, adaptés au domaine. Et dans le cadre de la méthode d'AS suggérée, nous proposons deux méthodes d'extraction des mots-germes adaptées à differents domaines. La méthode d'AS

[1] https://books.openedition.org/

proposée a ensuite servi à deux propositions d'amélioration de la qualité de la recherche de livres:

- Une méthode de reformulation des requêtes par réinjection de pertinence, dans laquelle l'analyse de sentiment aide à extraire l'information à partir de ressources Web sociales de livres.
- Une classification des phrases appartenant à des requêtes de recherche de livres, où nous analysons le rôle de l'analyse de sentiment dans cette classification.

En outre, afin d'améliorer la qualité des recommandations de livres de la plateforme Books de OpenEdition, nous proposons également une méthode permettant d'extraire la zone bibliographique dans les documents, comme pré-étape pour une méthode de recommandation de livres basée sur la citation inter documents.

De plus, nous élargissons l'horizon de la thèse avec une proposition supplémentaire qui sert indirectement l'un des objectifs principaux de la thèse dans l'amélioration de la qualité de la recherche et la recommandation de livres. Nous proposons une méthode de normalisation de texte pour réduire les difficultés causées par le langage informel du Web, dans le domaine de l'analyse des sentiments, et dans le but d'améliorer la prédiction des sentiments.

Mots clés: Analyse de sentiment, Intensité de sentiment, Recherche de livres, Recherche d'informations, Plongement de mots, Mots-grains, Réinjection de pertinence, Modèle de langue.

Publications

International conference articles

- **Amal Htait**, Sébastien Fournier, and Patrice Bellot. Sentiment Analysis and Sentence Classification in Long Book-Search Queries. CICLing. 2019.
- **Amal Htait**, Sébastien Fournier, and Patrice Bellot. Unsupervised Creation of Normalisation Dictionaries for Micro-Blogs in Arabic, French and English. CICLing. 2018. **-> Best Poster Award**
- **Amal Htait**, Sébastien Fournier, and Patrice Bellot. LSIS at SemEval-2017 Task 4: Using Adapted Sentiment Similarity Seed Words For English and Arabic Tweet Polarity Classification. SemEval. 2017.
- **Amal Htait**, Sébastien Fournier, and Patrice Bellot. SBS 2016: Combining Query Expansion Result and Books Information Score for Book Recommendation. CLEF (Working Notes). 2016.
- **Amal Htait**, Sébastien Fournier, and Patrice Bellot. LSIS at SemEval-2016 Task 7: Using Web Search Engines for English and Arabic Unsupervised Sentiment Intensity Prediction. SemEval@ NAACL-HLT. 2016.
- **Amal Htait**, Sébastien Fournier, and Patrice Bellot. Bilbo-Val: Automatic Identification of Bibliographical Zone in Papers. LREC. 2016.

International Journals

- **Amal Htait**, Sébastien Fournier, and Patrice Bellot. Unsupervised Creation of Normalization Dictionaries for Micro-Blogs in Arabic, French and English. Computación y Sistemas 22.3 (2018).

National conference articles

- **Amal Htait**. Adapted Sentiment Similarity Seed Words For French Tweets' Polarity Classification. DEFT@CORIA/TALN. 2018.
- **Amal Htait**, Sébastien Fournier, and Patrice Bellot. Identification Semi-Automatique de Mots-Germes pour l'Analyse de Sentiments et son Intensité. CORIA (RJCRI). 2017.

Softwares

- **ASID** (Adapted Sentiment Intensity Detection): The Software predicts the sentiment intensity of words, in Tweets of Arabic, French and English languages, and in Book Reviews of English language. It is based on an semi-supervised method using adapted to domain lexicon seed-words and distributed representations of words, known also as "word embedding".
  Shared as Open Source on Github: https://github.com/amalhtait/ASID
- **NormAFE**: The Software creates dictionaries for micro-blogs normalisation, in a form of pairs of misspelled word with its standard-form word, in the languages: Arabic, French and English. It is based on an unsupervised method for text normalisation using word embeddings models.
  Shared as Open Source on Github: https://github.com/amalhtait/NormAFE

Contents

List of Figures

List of Tables

List of abbreviations

- **ASID** Adapted Sentiment Intensity Detector
- **API** Application Programming Interface
- **CLEF** Conference and Labs of the Evalution Forum
- **Cléo** Centre pour l'édition électronique ouverte
- **CRF** Conditional Random Fields
- **DeFT** Défi Fouille de Textes
- **DH** Digital Humanities
- **IF** Information Filtering
- **IR** Information Retrieval
- **LT** LibraryThing
- **NDCG** Normalized Discounted Cumulative Gain
- **PMI** Pointwise mutual information
- **PRF** Pseudo Relevance Feedback
- **RF** Relevance Feedback
- **RSV** Retrieval Status Value
- **SA** Sentiment Analysis
- **SBS** Social Book Search
- **SHS** Sciences Humaines et Sociales
- **SI** Sentiment Intensity
- **SO** Sentiment orientation

General Introduction

The information resources on the web are of great variety, and are in continuous growth. As a result, the information seeker is confronted with a progressive level of difficulty to manage this mass of information, and to find relevant information to his/her expectations and needs. The answer would be with information retrieval and recommendation systems that can assure the access and the retrieval of information, by first determining the exact needs of the user's information, expressed as a query or based on his/her profile, then by retrieving a ranked list of documents ordered on the basis of their relevance to his/her needs. Classic information retrieval and recommendation approaches typically exploit the content of documents disregarding the social information of the documents or their linkage to other documents. In addition, most information retrieval systems expect a search query of *keywords*, therefore, these systems fail to assure a good quality retrieval for long natural language search queries.

The social web applications, in particular, offer a specific type of information, like opinions about items, arguments about beliefs or teachings, and even personal information, problems or events, which can form a rich source of social information. Such information, once connected to documents, can play an important role in these documents' retrieval. In addition, the writers in social web applications are often guided by sentiments expressed in their writings (e.g. opinions about an item in a review or a comment). Therefore, locating emotional sentences in such text can help in locating information within the text. For example, in the following part of a book review: [... ***The illustrations are beautiful and well thought-out and the book is educational and empowering***. *I bought this for my boyfriend's two-year-old niece ...*], the "emotional" sentence, in bold, contains information about the book as, for example, it has beautiful illustrations.

The general purpose in this thesis is to improve the book search and recommendation quality of OpenEdition's Books platform. Therefore, we choose to follow a main path in this thesis that can be resumed in the improvement of the information retrieval quality with the assistance of sentiment analysis. We divide the manuscript into two parts: the first part covers our contribution in the sentiment analysis field, mainly, and it includes our propositions for new approaches in sentiment analysis classification, in addition to a proposed method for normalisation thesaurus creation, used for decreasing the difficulties caused by the social web's informal language, and therefore improving the sentiment

prediction. The second part covers the information retrieval and filtering fields, as in book search and recommendation, and it includes our suggestions for new employments of sentiment analysis in information retrieval quality improvement, in addition to a presented approach to extract documents' bibliographical zone, as a pre-step for a book recommendation method based on an inter documents citation.

EquipEx DILOH project

This thesis is part of the EquipEx DILOH project[2] (Digital Library for Open Humanities), managed by the National Research Agency under the "Investissements d'avenir" program. The project is supported by Aix-Marseille University (AMU) alongside the CNRS, the École des Hautes Études en Sciences Sociales (EHESS) and the University of Avignon, and it is run by the Centre pour l'édition électronique ouverte (Cléo), in partnership with the Centre pour la Communication Scientifique Directe[3] (CCSD), the Open Access Publishing in European Networks[4] (OAPEN) Foundation and the Laboratoire d'Informatique et des Systèmes[5] (LIS).

The project is invested in Openedition's portal[6] of four platforms dedicated to electronic resources in the humanities and social sciences, developed and managed by Cléo. The portal includes four multilingual publishing and information platforms: OpenEdition Journals Catalog[7], OpenEdition Books[8], Blog Catalog[9] (research blogs) and Calenda[10] (announcements of international academic events). The portal is thus a space dedicated to the promotion of research, publishing tens of thousands of scientific documents that promote open access, while respecting the economic equilibrium of publications. OpenEdition platform involves a collection of documents in a form of a digital library catalog, extended by user-generated information of readers and professional editors. And it offers to its users a range of innovative tools exploiting that collection, like new methods of knowledge dissemination, advanced bibliographic features and advanced document search and recommendation system.

[2] http://www.agence-nationale-recherche.fr/ProjetIA-11-EQPX-0013
[3] https://www.ccsd.cnrs.fr/en/
[4] http://www.oapen.org/home
[5] https://www.lis-lab.fr/
[6] http://www.openedition.org
[7] https://www.openedition.org/catalogue-journals
[8] https://books.openedition.org/
[9] https://www.openedition.org/catalogue-notebooks
[10] https://calenda.org/

Approaches and contributions

Online digital catalogs, as Openedition's collection of documents, are often enriched with social information of user-generated content like comments and reviews. Such information could play an important factor in the search and recommendation procedures, but unfortunately it is not sufficiently exploited. In order to exploit these publicly available information, and since they usually have a sentimental trait, sentiment analysis is employed in this thesis as an information detector and extractor for new methods of documents search and recommendation.

In addition, this thesis is accomplished to meet OpenEdition's platforms needs for a multiple language search and recommendation improvement. Therefore, it consist on developing new approaches while covering the multilingual characteristic of the platforms. Accordingly, an approach for sentiment analysis prediction is suggested, easily applicable on different languages, thus, inexpensive in time and annotated data.

Based on the previously mentioned objective, we present our contributions:

- A semi-supervised method for sentiment intensity and polarity prediction [Htait, Sébastien Fournier, and Bellot 2016b][Htait, Sébastien Fournier, and Bellot 2017] [Htait 2018].
- Two methods, based on sentiment analysis, to improve book search by query expansion and sub-query classification [Htait, Sébastien Fournier, and Bellot 2019].

Furthermore, two additional contributions are presented in this manuscript:

- For the purpose of improving the sentiment analysis prediction by normalizing the informal language in social web applications, an unsupervised method is presented to create text normalisation thesaurus [Htait, Sébastien Fournier, and Bellot 2018].
- For the intent of books search and recommendation quality improvement, based on books inter-citation graph, a method for bibliographical zone detection is presented [Htait, Sébastien Fournier, and Bellot 2016a].

Overview of the thesis

This manuscript is divided into two main parts, preceded by this current general introduction. The first part of this manuscript covers our contribution in the sentiment analysis field, it presents several proposed and evaluated methods related to sentiment analysis field.The first part includes three chapters:

- The Chapter *Background to Sentiment Analysis*, where we present a briefing about the notions used throughout the first part of this manuscript, concerning Sentiment Analysis, including the main concepts and approaches used for sentiment prediction in text.

- The Chapter *Proposed Methods for Sentiment Analysis Prediction*, where we present our proposed methods for sentiment intensity and sentiment polarity prediction, tested in social web applications (Tweets and Reviews) and in several languages.

- The Chapter *Automatic Creation of Thesaurus for Text Normalisation*, where we seek a text normalisation by proposing a method for normalisation thesaurus' creation, to be used in the purpose of improving sentiment analysis prediction in social web applications of informal language.

The second part of this manuscript covers the information retrieval and filtering fields, it examines the proposed methods for information retrieval improvement, in addition to a bibliographical zone detection method serving for a future purpose of information filtering improvement. The second part of this manuscript includes three chapters:

- The Chapter *Background to Information Retrieval & Information Filtering*, where we present the basic knowledge concerning two interfering domains: Information Retrieval and Information Filtering, with their concepts and their main approaches by focusing on the ones used in the following chapters.

- The Chapter *Automatic Detection of Bibliographical Zone for Inter Citation Linkage*, where we suggest a method of bibliographical zone detection in articles and books (e.g. scientific books), for the purpose of providing a linking source between the books, to be then used in a future work related to the information filtering field, as in book recommendation or in retrieved books' re-ranking.

- The Chapter *Sentiment Analysis for Book Retrieval*, where we present our proposed methods to improve the information retrieval quality (book search) with the intervention of sentiment analysis: by introducing sentiment analysis in pseudo relevance feedback, then by a study of the correlation between sentiment analysis and topics in long book search queries, for the purpose of a future sub-queries classification.

The closing section of the thesis includes the conclusion and the perspectives. It recaps our main contributions in the thesis, in addition to a presentation of several future directions of our work.

Part I.

Sentiment Analysis in Social Web Applications

Chapter 1:
Background to Sentiment Analysis

Summary

In this chapter we briefly present the notions used throughout the first part of this manuscript, concerning Sentiment Analysis (SA). We start with the concepts of SA, its definition, its aspects and its relation with opinion extraction. Then, we present the main approaches used for sentiment prediction in text. And finally, we briefly present the evaluation measures employed in this part of the manuscript.

1.1. Concepts in the Sentiment Analysis field

Definition: Sentiment analysis (SA) is the automatic identification and extraction of subjectivity in text, and sentiment orientation (SO) is the measure of that subjectivity. In general, a subjective sentence expresses personal sentiment, feeling, emotion or attitude, and which can come in many forms, like opinions, allegations, desires, beliefs, suspicions, and speculations [Liu 2015].

Several researchers considered opinion mining as a second naming to SA [Pang and L. Lee 2008], but that is not accurate since an opinion is a sentiment oriented towards an entity or an aspect of the entity by an opinion holder, where an entity is a product, service, person, event, organization, or topic [Liu and L. Zhang 2012]. For example, in the sentence "*I loved this book*", the person expresses his feelings toward a book, which is considered an opinion about that *book*, but in the sentence "*I am happy*", we have a sentiment but it is not directed toward any entity, therefore it is not an opinion. In our thesis, we discuss SA in general and not opinion mining.

SO can be defined at different text granularities: from sentiment in words, phrases, sentences, micro-blogs messages; to sentiment in reviews and whole documents. And we call "sentiment lexicons" the association of word-SO (or phrase-SO), usually created by a manual annotation or through an automatic means (e.g. rainbow : positive).

1.2. Sentiment Analysis aspects

Generally, the sentiment analysis prediction consists of two aspects: Sentiment Polarity and Sentiment Intensity.

The **Sentiment polarity** includes two types of classification: binary sentiment classification, with a labeling as one of two predefined categories (*positive and negative*), and multi-class sentiment classification, with a longer set of predefined categories (e.g. *strong positive, positive, neutral, negative, strong negative*). Many researches seek binary sentiment classification of text, but the multi-class approach, with the *neutral* label, proved its effectiveness with a better distinction

between positive and negative texts [Koppel and Schler 2005].

The **Sentiment Intensity** consists of a scoring within the positive or negative range of values. According to [Russell 1980], the sentiment in text can be measured by two independent scales: the first measure is valence, or sentiment from negative to positive, and the other measure is arousal, or intensity from low to high. Therefore, as shown in Figure 1.1, the words on the left of vertical axis (arousal) has negative polarity, and words on the right has positive polarity. But also, the intensity of sentiment get stronger whenever we go higher with the vertical axis. For example the word *Delighted* is positively stronger than *Glad*, and the word *Frustrated* is negatively stronger than *Bored*. Eventually, each word is given a sentiment score, representing its sentiment intensity, combining its degree of positivity, negativity and objectivity.

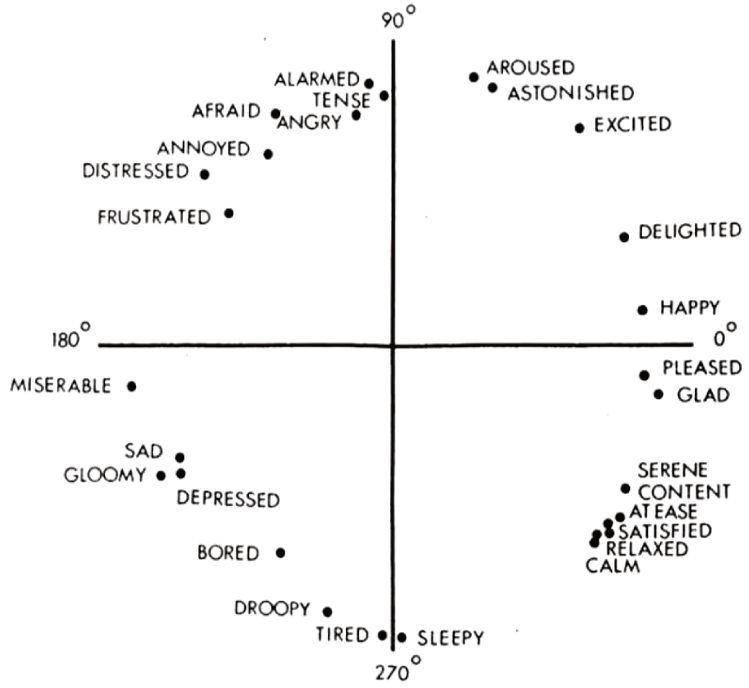


Figure 1.1.: Russell's circumplex model of affect [Russell 1980]

1.3. Sentiment Analysis prediction methods

SA belongs to the text mining field, and it serves of statistics, natural language processing (NLP), and/or machine learning for the process of subjectivity identification and extraction. The automatic extraction of sentiment in text is employed, in general, through two main approaches: the lexicon-based approach, by adopting sentiments lexicons in the process of sentiment prediction, and

the corpus-based approach, by building models or classifiers from a corpora of texts or sentences, used afterwards for sentiment prediction [Pang, L. Lee, and Vaithyanathan 2002].

1.3.1. Lexicon-based

The lexicon-based approach depends mainly on sentiment lexicons and dictionaries of labeled phrases or sentences, each assigned with a sentiment polarity or with a score reflecting its sentiment intensity. In addition the lexicon-based approach relies, in general, on the assumption that the sentiment orientation of a text is the combination of the sentiment orientation of its words or phrases. Established on the concept of lexicon-based approach, [Turney 2002] employed the Pointwise Mutual Information (PMI) to estimate the sentiment orientation of each phrase, using lexicons, to predict the sentiment orientation of whole reviews. Note that PMI is one of the standard measures of association in the collocation extraction, and it was brought into lexicography by [Church and Hanks 1990]. The PMI is used to measure the similarity between a phrase and a polarity class (positive or negative). The sentiment orientation of a given phrase is determined by calculating the difference between its similarity (PMI) to the positive polarity class and its similarity to the negative polarity class. The sentiment orientation (SO-PMI) of a phrase (several words or one word) is calculated as follows [Turney and Littman 2003]:

$$SO - PMI(w) = PMI(w, pos) - PMI(w, neg) \tag{1.1}$$

where PMI is given by the equation:

$$PMI(w, pos) = \log_2 \frac{freq(w, pos) \cdot N}{freq(w) \cdot freq(pos)} \tag{1.2}$$

where $freq(w, pos)$ is the number of times a phrase w occurs in a positive sentence or it is annotated as positive in the sentiment lexicons, $freq(w)$ is the frequency of the phrase w in the entire sentiment lexicons, $freq(pos)$ is the total number of positive phrases, and N is the total number of phrases. And $PMI(w, neg)$ is calculated similarly, as:

$$PMI(w, neg) = \log_2 \frac{freq(w, neg) \cdot N}{freq(w) \cdot freq(neg)} \tag{1.3}$$

As a result, a phrase would have a positive sentiment orientation when it has more associations with the positive polarity class (e.g., "lovely evening"), and a negative sentiment orientation when it has more associations with the negative

the polarity class (e.g., "nightmare event").

1.3.2. Corpus-based

The main factor in a corpus-based sentiments analysis system is the establishment of a corpus. The corpus is created by collecting textual data from a source (e.g. Twitter), then employing these data into the training of a machine learning model. We distinguish between supervised and unsupervised learning.

1.3.2.1. Supervised Learning

The corpus-based supervised learning approach is based on machine learning classifiers, such as Support Vector Machine (SVM) [Cortes and V. Vapnik 1995] [Valdimir and N. Vapnik 1995], Naïve Bayes (NB) [McCallum, Nigam, et al. 1998] and Deep Convolutional Neural Networks (CNN) [Simard, Steinkraus, Platt, et al. 2003], employed to an annotated dataset. The dataset is split into a training set and a testing set, where the classifier learns from the training set and builds a model used for the testing set classification.

For many years, SVM has been one of the most popular machine learning classifiers, it is a discriminative classifier defined by a separating hyperplane. Otherwise speaking, given an annotated training data, the algorithm outputs an optimal hyperplane which categorizes new examples of test data.

But since almost a decade, deep learning and neural classifiers has arisen as a powerful machine learning technique. And applying deep learning to SA has become very popular as well. Deep Learning is about learning multiple levels of representation and abstraction that help to make sense of data. Inspired by the structure of the biological brain, deep learning neural networks consist of a large number of information processing units arranged in layers, for the purpose of learning in multiple levels. Unfortunately, training a deep neural network is complicated and computationally very expensive [Lei, Shuai, and Bing 2018].

1.3.2.2. Unsupervised Learning

The corpus-based unsupervised learning approach include unlabeled dataset, where text is not labeled with the appropriate sentiments, such as Latent Dirichlet Allocation (LDA)[Blei, Ng, and Jordan 2003] and K-Nearest Neighbors (KNN) [Altman 1992]. Also word embeddings is an unsupervised learning approach that we are mainly using in this thesis.

Word embeddings is a word representations embedded on semantic vector spaces, in other words, word embeddings provides vector representations of

words, where every word gets assigned a unique N-dimensional vector, and similar words end up having values closer to each other. For example, the vectors of the words *Monday* and *Tuesday* have a much higher similarity than the vectors of *Monday* and *work*. Note that several approaches of sentiment analysis included a word embeddings employment for the purpose of combining unsupervised and supervised techniques [Maas, Daly, Pham, et al. 2011][Giatsoglou, Vozalis, Diamantaras, et al. 2017].

Word2Vec [Mikolov, K. Chen, Corrado, et al. 2013] is one of the most popular techniques to learn word embeddings using shallow neural network. Note that shallow neural networks is a technique of machine learning that usually has only one hidden layer. We can distinguish two models, implemented in Word2Vec, with their architecture shown in Figure 1.2: the *continuous Bag-of-Words* (CBOW), in which the model attempts to predict a center word given a sequence of its surrounding words. And the *skip-gram*, in which the model seeks to maximize the likelihood of the prediction of contextual words based on a given centered word.

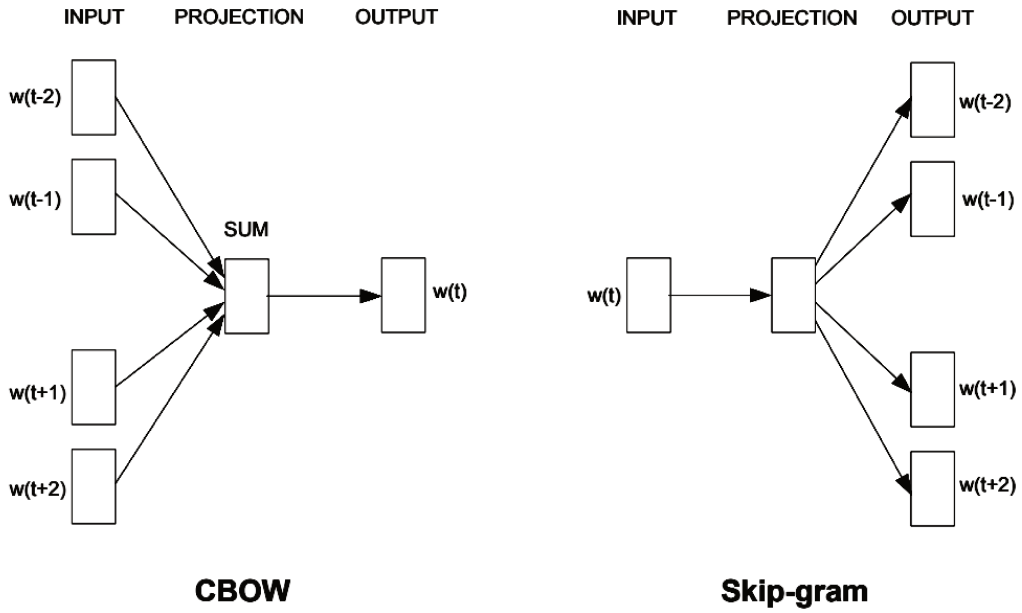


Figure 1.2.: Illustration of Word2vec models: Skip-gram and Continuous Bag-of-Word (CBOW) [Mikolov, K. Chen, Corrado, et al. 2013].

1.4. Evaluation measures in the Sentiment Analysis field

In order to evaluate the effectiveness of a sentiment analysis system, various performance measures have been proposed. In this part of the manuscript, we are presenting two sets of performance measures: the classification measures, as Precision, Recall, F-measure and Accuracy, presented briefly in Section 1.4.1, used for sentiment polarity prediction. The rank correlation coefficient measure, as Kendall and Spearman, presented in Section 1.4.2, used as a sentiment intensity performance measure by comparing the predicted intensity ranking of a list of phrases with its correct ranking.

1.4.1. Precision, Recall, F-measure and Accuracy

We consider, as an example, a sentiment analysis system, that can classify a set of words as *Positive* or *Negative*. As shown in Figure 1.3, in the group of words predicted as *Positive*, we can have *True Positive* (in green) as for the correct prediction of the positive class, and *False Positive* (in blue) as for the incorrect prediction of the positive class. Likewise, in the group of words predicted as *Negative*, we can have *True Negative* (in red) as for the correct prediction of the negative class, and *False Negative* (in orange) as for the incorrect prediction of the negative class.

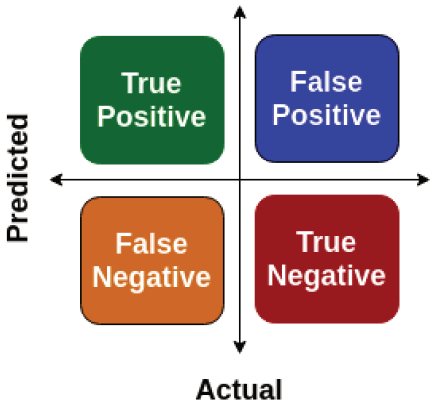


Figure 1.3.: The distribution of predicted and actual classification between positive and negative classes.

Based on the segmentation of prediction results, presented in Figure 1.3, the performance measures are calculated as:

- The *Precision* (or Confidence) is the proportion of predicted positive cases that are correctly real positives (True Positive), and it is calculated as fol-

lows:

$$Precision = \frac{True\ Positive}{True\ Positive\ + False\ Positive} \tag{1.4}$$

- The *Recall* (or Sensitivity) is the proportion of real positive cases that are correctly predicted positive, and it is calculated as follows:

$$Recall = \frac{True\ Positive}{True\ Positive\ + False\ Negative} \tag{1.5}$$

- The *F -β Measure* is a weighted harmonic mean of *Precision* and *Recall* and is generally used with $\beta = 1$. Therefore, it is calculated as follows:

$$F - \beta Measure = \frac{(1 + \beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \tag{1.6}$$

- The *Accuracy* is the ratio of correct predictions in all classes out of all the tested collection, and it is calculated as follows:

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \tag{1.7}$$

1.4.2. Kendall and Spearman

Kendall's tau [Kendall 1938] (or Kendall) and Spearman's rho [Spearman 1904] (or Spearman) are two commonly used criterion for detecting associations between two variables. In this thesis, they are used to evaluate statistical associations between two lists based on the ranks of the lists' elements. The first list is the ordered predicted intensity of a group of phrases, and the second list is the ordered list with correct sentiment intensity of the same group of phrases. As a brief and general comparison between Kendall and Spearman correlation measures: Kendall's values are usually smaller than Spearman's correlation values, Kendall's measure is less sensitive to error, and its p values are more accurate with smaller size of samples.

The Kendall and Spearman correlation coefficients can take values from -1 to +1, and the equations are presented as below:

- Kendall tau:

$$Kendall\ \tau\ = \frac{C - D}{C + D} \tag{1.8}$$

  Where C is the number of concordant pairs, and D is the number of discordant pairs.

- Spearman rho:

$$Spearman\,\rho = 1 - \frac{6 \cdot \sum_i d_i^2}{n(n^2 - 1)} \tag{1.9}$$

  Where n is the number of elements in the two variables and d_i is the difference in the ranks of the element at position i of each variable.

1.5. Conclusion

In this chapter we have presented the notions used throughout the first part of this manuscript, including the concepts of sentiment analysis. We continued by explaining the approaches used for sentiment extraction from text. Finally, we presented the employed evaluation measures.

Next, we present our contributions within the first part of this manuscript, with the first chapter of proposed methods for sentiment analysis: sentiment intensity and polarity predictions, in social web applications (Tweets and Reviews). Then, in the second chapter, we propose an unsupervised method to create normalisation thesaurus, for the purpose of improving sentiment analysis prediction in social web applications of informal language.

Chapter 2: Proposed Methods for Sentiment Analysis Prediction

Summary

2.1. Introduction

This chapter covers the proposed methods of Sentiment Analysis (SA) prediction, in microblogs and book reviews text. We focus, in our work, on predicting the sentiment intensity for its capacity of capturing more accurately the sentiment in text, as we are able to compare sentences having the same polarity orientation. For example, the sentence "*It's an excellent job*" is positively stronger than "*It's a good job*", even thought both are positive. Multiple supervised methods were exploited successfully in sentiment analysis prediction, but such methods need large-scale of annotated training data, rarely available, which limits strongly their adaptability to new domains and languages. Therefore, in this chapter, we suggest semi-supervised methods, requiring small annotated corpora. In addition, the experiments of the methods covered multiple languages : English, French and Arabic, to show their easy adaptability to different languages.

To be able to experiment and compare our proposed methods, we use the datasets provided by Semantic Evaluation (SemEval) workshops[11], and by DEFT workshop[12], concerning sentiment analysis in tweets. Twitter is considered a rich source of opinions, and several data mining tasks are mainly based on tweets. Since each tweet mostly reflects one sentiment orientation [Pak and Paroubek 2010][Nabil, Aly, and Atiya 2015], Twitter occupies a large section in SA researches. Furthermore, Twitter is supported by many languages and tweets of 34 languages are available [13], which gives the chance to experiment with several languages.

The chapter contains two sections, where we present the suggested methods for sentiment prediction in text, presented as below:

- Combining Web search engines and sentiment lexicons methods for a semi-supervised sentiment intensity prediction, employed on tweets.
- Using adapted seed-words lists and word embeddings models for a sentiment intensity and polarity prediction, applied on tweets and book reviews.

2.2. Related work

The lexicon-based approach is one of the well known approaches for sentiment prediction, and it has been the ground of many sentiment analysis systems [Liu 2012].

[11] https://aclanthology.info/venues/semeval
[12] https://deft.limsi.fr/2018/
[13] https://dev.twitter.com/web/overview/languages

The lexicon-based approach can be accomplished by making use of semantic relations between words in a lexical resource (e.g. WordNet [14]). [Kamps, Marx, Mokken, et al. 2004] built a network of adjectives by considering that the sentiment polarity of an adjective can be determined based on its shortest path to positive and negative adjectives ($good$ and bad) relying on the specified synonym relation in WordNet. And SentiWordNet lexicon was constructed by [Esuli and Sebastiani 2006] based on a relational network of word senses based on the specified word definitions in WordNet.

[Turney and Littman 2003] and [Turney 2002] made use of associative relations in a corpus, and their work form the base of our proposed method in Section 2.3, for its simplicity and efficiency. Their main purpose was to conclude sentiment orientation using PMI to measure the similarity between a phrase and a polarity class (positive or negative). That method was successfully applied by other researchers, like [Kiritchenko, Zhu, and Mohammad 2014] for creating their large scale of tweets sentiment lexicons. In addition, [Turney and Littman 2003] pursued the co-occurrence of a term with the words ($excellent$ and $poor$) by the returned number of hits of a search engine, a method applied also in many other researches [Prabowo and Thelwall 2009][Demartini and Siersdorfer 2010]. For our proposition in Section 2.3, we combine the two mentioned methods for the purpose of terms' sentiment intensity prediction.

Seed-words, on the other hand, were the base of many sentiment analysis experiments, mostly based on semi-supervised learning to reduce the need of large annotated training corpora. For example, [Ju, Shoushan Li, Su, et al. 2012] worked on a sentiment classification method that aims to train a classifier with a small number of labeled data (called seed-data). [Turney 2002; Turney and Littman 2003] also worked on a semi-supervised method where they used the statistical measures, such as PMI, to calculate the similarities between words and a list of 14 seed words, manually extracted form restaurants reviews. Also, [Maas, Daly, Pham, et al. 2011] used a similar concept as [Turney 2002], but by using a larger list of seed-words and with cosine similarity measure in word embeddings, for the purpose of combining unsupervised and supervised techniques. These mentioned work does not take into consideration the differences in expressions meanings between domains. For example the word $cool$ is an adjective that refers to a moderately low temperature and has no strong sentiment orientation, but it is often used in microblogs as an expression of admiration or approval. Then, $cool$ is considered a positive seed-word in microblogs. Therefore in Section 2.4, we focus on a proposed method that uses adapted positive and negative seed-words and word embeddings models, a method that can be applied to different domains and different languages.

[14] https://wordnet.princeton.edu/

In the following sections, we propose two methods for sentiment analysis and we test them in different languages, including the Arabic language since it faces more processing challenges than the English language. The words in Arabic language can have several meanings depending on their position within a sentence, on their type (Verb, Noun, etc), and on the position of their Arabic vowel marks. For example the word "ساق" can have the meaning of "drive" as a verb, and "leg", "barman" or "trunk" as a noun. But even though, we can still find some interesting experiments in lexical-based sentiment analysis, in Arabic language: [El-Beltagy and Ali 2013] built a sentiment lexicon based on a manually constructed seed list of sentiment lexicons of 380 words. Using this list, with assigned sentiment intensity score for each value, they were able to calculate the sentiment orientation for a set of tweets in Arabic language (Egyptian dialect). In another work, [Eskander and Rambow 2015] presented a large list of sentiment lexicons called SLSA (Sentiment Lexicon for Standard Arabic) where each value is associated with a sentiment intensity score. The scores were assigned due to a link created between the English annotation of each Arabic entry to a synset from SentiWordNet [Cambria, Speer, Havasi, et al. 2010].

2.3. Sentiment Intensity prediction combining sentiment lexicons and Web search engines

Our proposal, for the sentiment intensity prediction, is a lexicon-based method that uses the pointwise mutual information (PMI) to calculate the words association with a polarity class (positive or negative). Such method depends on the presence of all the words, which we question their sentiment intensity, in the sentiment lexicon corpora. Unfortunately, this is not always the case, especially in microblogs and social information, with slang words and abbreviations. Therefore, we suggest the use of web search engines to calculate the statistical dependency between words and a polarity class, since the web is a rich environment of informal language.

2.3.1. General overview of the proposed method: Combining lexicons and search engines

The proposed method is a semi-supervised lexicon-based method, followed by the use of a web search engine to maximize the chances of finding all informal expressions that a classic sentiment lexicons would not include. Algorithm 1 presents the general followed path of the proposed method, in sentiment intensity prediction combining lexicons and search engines. For the first step in this

method, if the phrase w is part of the sentiment lexicons, we calculate the sentiment score of that phrase using the PMI measure, as shown in the Equation 1.1 used by [Turney and Littman 2003].

The dictionaries of sentiment lexicons and labeled tweets can not include all the possible test words and phrases, for example: the hashtag phrases (e.g. #live_love_laugh), the phrases with no space between the words (e.g. goodvibes), and in Arabic language the English words written in Arabic characters (e.g. cute written as كيوت). To solve that issue, we suggest a second step applied for the phrases that we could not find in the sentiment lexicons, and that step is based on web search engines to calculate the sentiment intensity. According to [Turney 2002], a sentiment orientation of a word can be calculated based on the frequency of its co-occurrence with a positive reference word "$excellent$" and its co-occurrence with a negative reference word "$poor$" within the web pages content. The words "$excellent$" and "$poor$" are considered reference words, and also called seed-words. For that purpose we propose the Equation 2.10 to calculate the sentiment orientation (SO) of a phrase p, once it is not found in sentiment lexicons, as shown in the Algorithm 1 with the condition for that phrase to be found by the search engine:

$$SO(p) = \log_2 \frac{hits(p\ NEAR\ "excellent") \cdot hits("poor")}{hits(p\ NEAR\ "poor") \cdot hits("excellent")} \tag{2.10}$$

Where $hits(x)$ is the number of pages returned from a search engine for a query x. For example, $hits('poor')$ represents the number of pages returned for the query 'poor'. When the query is presented by the phrase p and the word $excellent$ (or $poor$ as a seed-word) connected by the operator $NEAR$, it represents the co-occurrences of p and the seed-word in the same page on a specified range of words (the range of 10 words is usually chosen [Turney 2002]). The extracted score of SO, for each phrase, is considered the sentiment intensity score for that phrase.

For the Arabic language, we suggest the same previously explained method and equations. As for the reference words (or seed-words), we first tested using the translation of "$poor$" and "$excellent$" to the Arabic language (فقير and ممتاز), but the bad results conduct us to the conclusion that these words are not as commonly used in Arabic language tweets as in English language tweets. Thus, a new seed-words list is suggested, with strong sentiment orientation, manually created for this task:

1. Arabic Positive seed-words:
   رائع، جميل، أحسن، أفضل، فرح، جيد، ذكي
   Translated to English language as:

wonderful, beautiful, better, preferable, good, joy, clever

2. Arabic Negative seed-words:
   مخيف، قبيح، أسوأ، غلط، حزين، سيئ، غبي
   Translated to English language as:
   scary, ugly, worst, mistake, sad, bad, stupid

Eventually, as presented in the last step of the Algorithm 1, the phrase is assigned the value 0.5 as its sentiment intensity score, when it is not part of the lexicons, neither found by the search engine.

Algorithm 1: Sentiment Intensity prediction combining lexicons and search engines

Result: Sentiment Intensity of the phrase p, as $Senti$
Prerequisites: Sentiment lexicons L, Search engine E, Phrase p
$Senti(p) \leftarrow 0.0$;
if $p \subset L$ **then**
  $Senti(p) = PMI(p, pos)$ - $PMI(p, neg)$, using Equation 1.1 ;
else
  nb_hits_pos = number of returned pages with the co-occurrences of p and the positive seed-word;
  nb_hits_neg = number of returned pages with the co-occurrences of p and the negative seed-word;
  if $nb\_hits\_pos > 0$ *and* $nb\_hits\_neg > 0$ **then**
    $Senti(p) = SO(p)$ based on nb_hits_pos and nb_hits_neg, using Equation 2.10 ;
  else
    $Senti(p) \leftarrow 0.5$;

2.3.2. Experiments of Sentiment Intensity prediction on Tweets

For the experiments, two datasets are used: the trial data provided by SemEval-2015 [Rosenthal, Nakov, Kiritchenko, et al. 2015] Task-10 Subtask-E[15] (400 phrases of General English language phrases), and the trial data provided by SemEval-2016 [Nakov, Ritter, Rosenthal, et al. 2016] Task 7[16] (200 phrases of Arabic language phrases). These datasets are provided as phrases with an approximate value of sentiment intensity for each phrase. In Table 2.1, we present a sample of the data where "General English" represents the phrases in English language of homogeneous polarity, and "Arabic" represents the phrases in Arabic

[15] http://alt.qcri.org/semeval2015/task10/
[16] http://alt.qcri.org/semeval2016/task7/index.php?id=data-and-tools

language with the translation to English for each. For evaluation purposes, we use the rank correlation coefficient of Kendall and Spearman to measure the degree of similarity between the ranks given by SemEval's Organisers and our ranks. And to increase the chances of finding the phrases in the sentiment lexicons or by search engines, a pre-processing is applied on the phrases, by removing hashtags and replacing the underscores by spaces.

Table 2.1.: A sample of the dataset provided in the sentiment intensity prediction task of SemEval-2016 Task-7 [Nakov, Ritter, Rosenthal, et al. 2016], where each term (or phrase) is assigned a sentiment intensity score between 0 and 1.

General English	Score	Arabic	Translation	Score
romantic	0.972	حب	love	0.981
superlative	0.778	تلاقي	encounter	0.694
pain	0.028	طب	medicine	0.694
criminal	0.021	قاتل	killer	0.025
violent	0.021	تدمير	destruction	0.019
menace	0.014	ارهابي	terrorist	0.000

The first part of the method is based on the number of times a phrase occurs as positive or negative in the sentiment lexicons. Therefore, and to avoid an "over fitting" to one collection of sentiment lexicons, we use several collections where we add up the number of occurrences of a phrase in all of them. In addition, using several sentiment lexicons collections helps predicting more accurately the sentiment intensity score. For example, a phrase occurring as positive in all collections would have a higher score (more positive) than a phrase occurring only in some collections.

For the English language, we use the below manually constructed sentiment lexicons:

- Bing Liu Opinion Lexicon, which is a list of English positive and negative sentiment words (almost 6800 words) compiled over many years [Hu and Liu 2004].
- MPQA (Multi-Perspective Question Answering) Subjectivity Lexicon, a Multi-Perspective Question Answering Subjectivity Lexicon [Wilson, Wiebe, and Hoffmann 2005].

And the below automatically constructed sentiment lexicons:

- Sentiment140 corpus, of 1.6M tweets with positive and negative emoticons [Go, Bhayani, and L. Huang 2009]. It was automatically annotated, by assuming that any tweet with a positive emoticon, like :), is positive, and any tweet with negative emoticon, like :(, is negative. Note that even thought such concept tend to be correct, but it does not take into consideration $Sarcasm$, where a tweet can be negative with a positive emoticon (e.g. Please shoot me :)).

- NRC Hashtag Sentiment Lexicon, of 62k terms extracted from tweets with sentiment-word hashtags such as $\#amazing$ and $\#terrible$. [Mohammad and Turney 2013]. Note that such automatic annotation method risk a margin of error where a tweet can be neutral with a positive or a negative hashtag (e.g. A turtle coming out of brumation #amazing).

- SentiWordNet 3.0 [17] [Baccianella, Esuli, and Sebastiani 2010], a set of 8k terms which are the result of automatic annotation of all WordNet synsets according to their degrees of positivity, negativity, and neutrality. Note that this method is limited to the available vocabulary in SentiWordNet.

- Sentiment words from the MPQA list of 3k positive words and 5k negative words, automatically annotated using the word senses for subjectivity [Wilson, Wiebe, and Hoffmann 2005].

- And we also use the test file's data of Semeval-2013 Task-2 (subtask-A) [18] with 10k positive and negative annotated tweets.

For the Arabic language, we use the below sentiment lexicons:

- Arabic Sentiment Tweets Dataset[19] [Nabil, Aly, and Atiya 2015], a set of Arabic tweets containing over 10k entries, manually annotated.

- Twitter data-set for Arabic Sentiment Analysis[20], 1k positive tweets and 1k negative ones on various topics such as: politics and arts, annotated manually.

- LABR Lexicons, a large scale of Arabic sentiment analysis benchmark, containing over 63k book reviews each with a rating of 1 to 5 stars, where we selected the ones rated 5 as positive, and the ones rated 1 as negative [21].

- NRC Hashtag Sentiment Lexicon is provided with many languages, including Arabic language [Mohammad and Turney 2013].

[17] File:subjclueslen1-HLTEMNLP05.tff (http://www.cs.pitt.edu/mpqa/)
[18] https://www.cs.york.ac.uk/semeval-2013/task2/index.html
[19] http://www.mohamedaly.info/datasets/astd
[20] https://archive.ics.uci.edu
[21] http://www.mohamedaly.info/datasets/labr

The second step of our method requires a searching engine, therefore, we start with two search engines' comparison for the purpose of choosing the one achieving best results: Bing Search Engine API and Google Search Engine API. Bing Search Engine[22] gives till 5k Transactions by month, set at 50 results per query. The limited number of search returns can decrease the efficiency of Bing Search Engine in our proposed method. To test our theory, we take the following example by searching for the word "awesome" near the word "poor", at "near:5", we get 21360 results. And the same search is applied at "near:10", we get 21332 results, although, logically we should get a larger number since we are searching in a wider range (10 instead of 5). We assume that this issue is caused by the limitation in the number of search returned by Bing Search API, which as consequences can give bad results for the sentiment intensity prediction, since the method is based on the number of search returned. And as expected, by applying the Equation 2.10 on the test data provided by SemEval-2015 Task-10 Subtask-E (General English phrases), using Bing Search Engine API, we get the below very low scores, reflecting the weak level of correlation:

- Kendall rank correlation coefficient = 0.029
- Spearman rank correlation coefficient = 0.039

Google Search API[23] is then used in the experiments, with queries consisted of a phrase (which the sentiment orientation is in question) and the reference word "*excellent*" (or "*poor*"), within the interval of ten words (in either order).

Google Search API allows a limited number of queries by user, by day, therefore, we tested its efficiency, in English language, with a sample of first 40 phrases from the test data file of SemEval-2015 Task-10. Table 2.2 shows a comparison between the results of Bing Search API, Google Search API and SO-PMI (lexicon-based from Equation 1.1), with the prediction of sentiment intensity for the sample of 40 phrases. Google Search API achieved the best results in this sample comparison.

Table 2.2.: Sentiment Intensity prediction for 40 phrases using Bing Search API, Google Search API and SO-PMI.

Method	Kendall	Spearman
Bing Search API	0.015	0.023
Google Search API	**0.287**	**0.412**
SO-PMI	0.207	0.305

Combining the use of SO-PMI and Google Search API method, where the search engine is used when SO-PMI fails to predict the sentiment intensity, al-

[22] http://www.bing.com/toolbox/bingsearchapi
[23] https://developers.google.com/web-search/docs/

lows to benefit from Google Search API's good results by surpassing it limitation in search queries number. Note that SO-PMI fails to predict the sentiment intensity when the phrase is not included into the sentiment lexicons. Table 2.3 shows that combining these two methods improved the results of sentiment intensity prediction of the 400 phrases in SemEval-2015's General English language phrases. To note that Google Search API is applied on 5% only of the dataset's phrases, in this methods combination, since those 5% were the only phrases not found in the dictionaries of sentiment lexicons. And in case of no results returned from Google Search API, the phrase is classified as Neutral and the value 0.5 is given as its sentiment intensity.

Table 2.3.: Sentiment Intensity prediction for SemEval-2015 phrases, in English language, using the methods: SO-PMI and "SO-PMI + Google Search API".

Method	Kendall	Spearman
SO-PMI	0.443	0.620
SO-PMI + Google Search API	**0.452**	**0.631**

Table 2.4 presents the results of the following methods: SO-PMI and "SO-PMI + Google Search API", using the development dataset of SemEval-2016 task-7 (200 of Arabic phrases) for sentiment intensity prediction in short phrases extracted from tweets of Arabic language, where the Google Search API is used with 20% of the phrases (since those 20% were not found in our sentiment lexicons). The results shows that the use of Google Search API did not increase the values, and that could be due to our choice in Arabic positive and negative seed-words included in the search query.

Table 2.4.: Sentiment Intensity prediction for SemEval-2015 phrases, in Arabic language, using the methods SO-PMI and "SO-PMI + Google Search API".

Method	Kendall	Spearman
SO-PMI	**0.417**	**0.584**
SO-PMI + Google Search API	0.402	0.561

For our participation [Htait, Sébastien Fournier, and Bellot 2016b] at SemEval-2016 task-7[24], we applied the method with the higher scores in Tables 2.3 and 2.4, on the challenge's datasets including: a list of 2799 phrases in English language of homogeneous polarity (described as General), a list of 1069 phrases in English language with mixed polarity (e.g. happy accident), and a list of

[24]http://alt.qcri.org/semeval2016/task7/

1078 phrases in Arabic language. The official results of our participation are presented in Tables 2.5, 2.6 and 2.7. By comparing the results of our system with the best results [Feixiang Wang, Z. Zhang, and Lan 2016][Refaee and Rieser 2016], which are achieved using supervised methods with extremely large training corpora (e.g. 1.6M phrases), we can say that our system attained competitive results in the sub-tasks of: mixed polarity English phrases and Arabic phrases, despite its relatively *low-cost* as a semi-supervised method, requiring a limited size of sentiment lexicons.

Table 2.5.: SemEval-2016 Task-7 results for General phrases in English language.

	Rank	**Kendall**	**Spearman**	**Supervision**
Team ECNU	1	0.704	0.863	Yes
Team UWB	2	0.659	0.854	Yes
Our Participation	3	0.345	0.508	No + Lexicons

Table 2.6.: SemEval-2016 Task-7 results for Mixed Polarity phrases in English language.

	Rank	**Kendall**	**Spearman**	**Supervision**
Team ECNU	1	0.523	0.674	Yes
Our Participation	2	0.422	0.590	No + Lexicons
Team UWB	3	0.414	0.578	Yes

Table 2.7.: SemEval-2016 Task-7 results for phrases in Arabic language

	Rank	**Kendall**	**Spearman**	**Supervision**
Team iLab-Edinb.	1	0.536	0.680	Yes
Team NileTMRG	2	0.475	0.658	Yes
Our Participation	3	0.424	0.583	No + Lexicons

2.3.3. Discussion

In this section, we presented our proposed method, and experiments, of sentiment intensity prediction by combining a lexicon-based method with a "search engine" method, in English and Arabic languages. In addition, we presented our participation's results in SemEval workshop of three sub-tasks. For the General English sub-task, Table 2.5, our system has modest but interesting results. For the Mixed Polarity English sub-task, Table 2.6, our system achieved the second place. And for the Arabic phrases sub-task, Table 2.7, our system has very interesting and promising results.

The results, in general, are encouraging but we can indicate the suggested method's weak points that we work on solving in the following section:

- The dependency on large sentiment lexicons makes it difficult to be applied on other languages.
- The reference words of strong sentiment orientation, or seed-words, employed in the experiments are not very common in the microblogs' vocabulary.
- Technical difficulties and limitations faces the use of the web search engines (e.g. Google Search API allows a limited number of queries by user, by day).

2.4. Sentiment Intensity and Polarity prediction using adapted seed-words and word embeddings models

A new method is proposed in this section, which carries several suggested improvements to the previous method: first, and to solve the issue related to uncommon seed-words, we propose the creation of new seed-words lists, more relevant to the domain of text (tweets or book reviews). Second, we suggest replacing the dictionaries of sentiment lexicons and the search engines by a word embeddings' model, where the similarity between words is calculated with the cosine similarity measure. This change can eliminate the dependency on large annotated dictionaries and on search engines of limited access.

In the following sections, we present the suggested method with new seed-words lists extraction, the creation of word embeddings models, and the effectiveness evaluation of the new proposed method.

2.4.1. General overview of the proposed method: Employing adapted seed-words and word embeddings

The new suggested method requires new seed-words lists for the sentiment intensity prediction, more specifically, adapted to domain seed-words. Such seed-words capture the specific local meanings within the domain, for example, the word $touching$ can refer to a positive opinion about a movie or a book, but that's not the case in other contexts, where it could be a felony in jurisdiction domain context. To fulfill that purpose, we propose following procedures that can be easily applied on different languages. Therefore, two methods are suggested for

adapted seed-words extraction, a semi-automatic and an automatic method, explained and applied in Sections 2.4.2 and 2.4.3.

In addition, the new suggested method, involves the employement of word embeddings models. Word embeddings, or distributed representations of words in a vector space, are currently considered to be among a small number of successful applications of unsupervised learning. Also, they are capable of capturing lexical, semantic, syntactic, and contextual similarity between words. We suggest the creation of word embeddings models trained by text corpora of same domain as the seed-words, since embeddings from generic and general domain corpora fail to capture specific local meanings within the domain.

Consequently, the created seed-words lists and word embeddings models are used to predict sentiment intensity in words as the following example: To predict the sentiment intensity of the word W:

1. First, the cosine similarities between the vector representing W in the word embeddings model and all the extracted positive seed-words' vectors are calculated. The average of these scores would be W_p, representing the similarity between W and all positive seed-words.

2. Then, the cosine similarities between the vector representing W and all extracted negative seed-words's vectors are calculated. The average of these scores W_n would represent the similarity between W and all negative seed-words.

3. Finally, the difference between W_p and W_n represents the sentiment intensity of the word W.

And an example of sentiment intensity prediction, of the word "$exceptional$" in the book reviews domain is presented in Figure 2.4.

Note that in the work of [Turney and Littman 2003], fourteen seed-words were selected for their lack of sensitivity to context, presented in Table 2.8, as they preserve their sentiment polarity independently from the context (e.g. $Excellent$ is a positive seed-word). We can notice that some of these words are far from being common words, for example the words $superior$, $fortunate$ and $inferior$ are rarely found in social media texts. In our experiments, we compare the results of Turney's seed-words with our new adapted to domain seed-words of our proposed method.

2.4.2. Semi-automatic extraction of adapted seed-words

A semi-automatic method is suggested for extracting adapted to domain seed-words, and it can be resumed in the following steps:

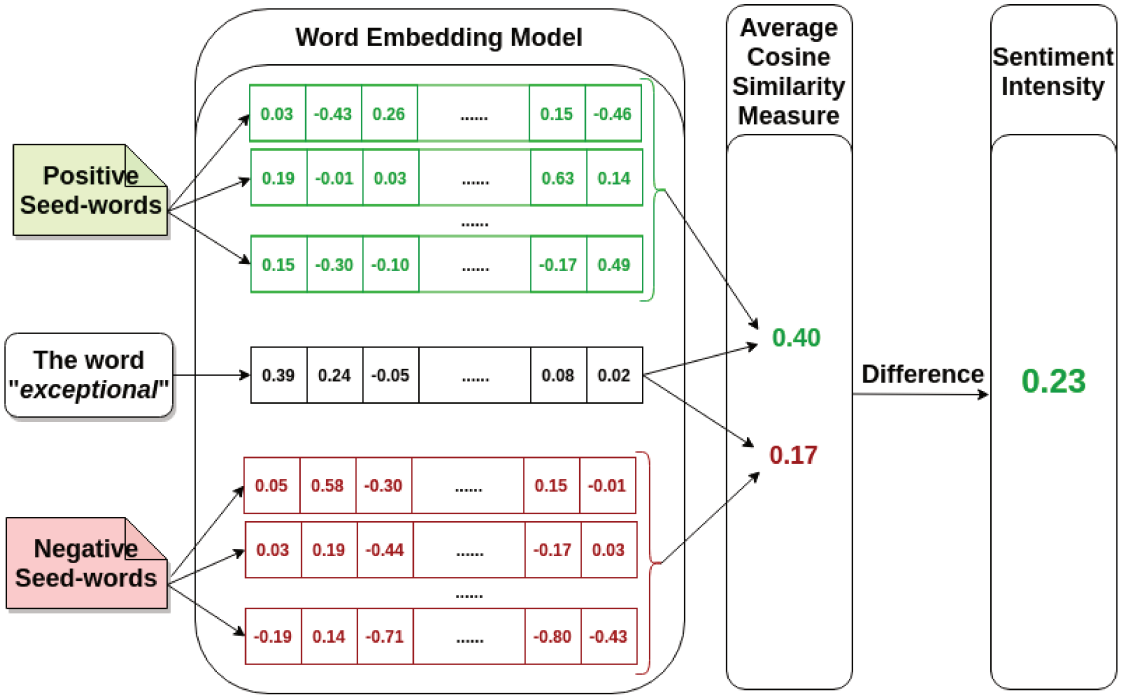


Figure 2.4.: An example of sentiment intensity prediction for the word *exceptional* in the book reviews domain, where the score is greater than zero, therefore positive.

Table 2.8.: The classic seed words suggested by [Turney and Littman 2003].

Positive	good, nice, excellent, positive, fortunate, correct, and superior.
Negative	bad, nasty, poor, negative, unfortunate, wrong, and inferior.

- First, collecting the most frequent words (or tokens) from an annotated dataset of positive and negative text (e.g. a set of annotated tweets) to create a list of most common words in positive text and a list of most common words in negative text.
- Then, after removing stop-words from the previously created lists, two sets of words are selected manually based on their relevance to the text domain, and on their strong sentiment orientation.

In the following sections, we present the semi-automatic extraction of seed-words applied on tweets, in English, Arabic and French languages, and on book reviews in English language.

2.4.2.1. English Tweets' seed-words

For the extraction of tweets' seed-words in English language, from the annotated tweets of Sentiment140 [Go, Bhayani, and L. Huang 2009], we extract automat-

ically the most frequent words in positive tweets and in negative tweets after removing the stop words. Then, 40 most relevant to tweets words, between the first 100 words, has been selected manually. The English positive and negative extracted seed-words, related to tweets domain, are as shown in Table 2.9.

Table 2.9.: The lists of Positive and Negative seed-words extracted from tweets, in English language.

Positive	**Negative**
love, like, good, win ,	ill, fucking, shit,
lol, hope, best, thanks,	fuck, hate, bad, break,
funny, haha, god, amazing,	sucks, cry, damn, sad,
fun,beautiful, nice, cute,	stupid, dead, pain, sick,
cool, perfect, awesome,	wtf, lost, worst, fail,
okay, special, hopefully,	bored, scared, hurts,
glad, congrats, excellent,	afraid, upset, broken,
dreams, sunshine, hehe,	died, stuck, boring,
positive,fantastic, dance,	horrible, negative,
correct, fabulous, superior,	unfortunate, inferior,
fortunate, relaxing,	unfortunately,poor,
happy,great, kind, laugh,	need, suck, wrong,
haven, wonderful, yay,	evil, missed, sore, alone,
enjoying, sweet,	crap, hell, tired, nasty.

2.4.2.2. Arabic Tweets' seed-words

For the extraction of tweets' seed-words in Arabic language, we automatically select the most frequent words in positive tweets and in negative tweets from two annotated corporas of Arabic tweets (Arabic Sentiment Tweets Dataset[25] and Twitter data-set for Arabic Sentiment Analysis[26]). Then, due to the Arabic language complexity, as some extracted terms are not comprehensible due to differences in dialects, we suggest the use of a list with 240 positive and negative standard Arabic language seed-words [Mohammad and Turney 2013] to filter our list, instead of the manual extraction previously used with the English language. Thereby, the extracted seed-words lists are formed from the intersection between the most frequent words in the annotated tweets and the predefined list of [Mohammad and Turney 2013]. Therefore, the seed-words lists belong to modern standard Arabic lexicon (as the words in the predefined list), that are mostly common in social media texts, without any dialect specification (e.g. Lebanese, Egyptian, Algerian). Table 2.10 presents the final extracted seed-word list.

[25]http://www.mohamedaly.info/datasets/astd
[26]https://archive.ics.uci.edu

Table 2.10.: The lists of Positive and Negative seed-words extracted from tweets, in Arabic language.

Positive words	Translation	Negative words	Translation
خير	benevolent	بال	worn
الجمال	fairness	بشع	ugly
كبير	grand	وسخ	filthy
أعلى	superior	جائر	unjust
حسن	well	عيب	flaw
عظيم	great	خطير	dangerous
رائع	wonderful	حقير	despicable
نادر	exceptive	بايخ	vapid
جمال	beauty	حزين	sad
كريم	generous	قذر	dirty
أعظم	greatest	هائل	massive
نبيل	noble	مقرف	nasty
جميل	beautiful	باطل	invalid
صالح	valid	تافه	trifle
دقيق	accurate	ملعون	damned
مشرق	bright	مرفوض	unacceptable
طيب	delicious	مسكين	poor
حلو	sweet	فاسد	corrupt
جيد	good	مؤسف	regrettable
عبقري	genius	فظيع	horrible

Also, and for comparison purposes, we translate the English seed-words, from Table 2.9, to Arabic language, as shown in Table 2.11. The purpose of this translated list is to verify if a simple translation of adapted seeds can be more or less successful in sentiment intensity prediction, than the creation of new lists. Note that the translation is applied using "Google Translate" [27], since it is able to translate slang words. In case of several meanings of a single word, we apply a manual selection to extract the word with the most accurate meaning, and also to select the word with stronger sentiment orientation.

[27] https://translate.google.com/

Table 2.11.: The Translated Arabic seed-words from the Tweets' adapted English seed-words list of Table 2.9.

Positive Translated Seed-words
هههه , جيد , اعجاب , حب , متفوق , صحيح , محظوظ , إيجابي , لطيف , جيد , ممتاز , جذاب , جميلة , مدهش , ياي , أفضل , رائع , مرح , أمل , سعيد , هاها , عظيم , شكر , إشراق , أحلام , رقص , مثالي , حنون , ملاذ , تتمتع , مبروك , انتصر , حسنا , الله , حلو , مريح , ضحك , مميز
Negative Translated Seed-words
كراهية , حاجة , حزين , افتقد , وضيع , خاطئ , مؤسف , سلبي , شرير , سيئ , فقير , توفي , اللعنة , الم , بكاء , وحده , مريض , شر , عالق , قرف , غبي , يؤلم , ضجر , متعب , سخيف , كسر , مكسور , أسوأ , للأسف , ميت , فشل , قرحة , حماقة , جحيم , ممل , ضائع , كريه , خائف , خائفة , منزعج , فظيع ,

2.4.2.3. French Tweets' seed-words

For the French language, we worked on tweets about French public transport provided by DEFT-2018 workshop[28] [Claveau, Minard, Cellier, et al. 2018] task2 of global tweet polarity prediction as positive, negative, neutral or mixed polarity. Therefore, the extracted seed-words would be adapted to tweets of public transport domain. For example the word *retard* (as *late* in French) is considered a negative seed-word since a late train or bus usually provoke negative feelings.

The procedure of extracting seed-words is done by creating two sets of datasets as positive and negative tweets based on the DEFT-2018 training corpora of public transport tweets in French language. Then, the most frequent words in the two lists of tweets are automatically extracted, after eliminating stop-words. As a result, two lists (positive and negative) of most frequent words are created. And finally, a manual filter is applied on these two lists to eliminate the irrelevant words. The list of French seed-words adapted to public transport tweets is as shown in Table 2.12, with 63 positive seed-words and 63 negative seed-words.

Also, for comparison purposes, we translate the English seed-words, from Table 2.9, to French language as shown in Table 2.13. "Google Translate" [29] is used for the translation, and in case of several meanings of a single word, we apply a manual selection.

[28] https://deft.limsi.fr/2018/
[29] https://translate.google.fr/

Table 2.12.: The lists of Positive and Negative seed-words extracted from public transport tweets, in French language.

Positive	**Negative**
mdr, bien, merci, bon, mdrr, ptdr, juste, aime, beau, cool, heureusement, rire, adore, bravo, belle, blague, ok, gagner, ptdrrr, ptdrr, génial, gnial, ouais,, content, courage, vive, offre, joie, haha, ptdrrrr, tranquille, gentil, parfait, bonheur, magnifique, jéme, jme, mignon, gratuit, amour, bons, ahah, ouf, direct, trql, heureux, remercie, jolie, respect, bisous, mdrrr, coucou, mdrrrr, ptdrrrrr, sourire, mddr, mdrrrrr, rigole, bonne, super, meilleur, prfre, chou.	retard, louper, grve, grave, ratp, trafic, problme, pute, puent, coup, panne, mal, flemme, fdp, gueule, bordel, accident, rater, couilles, retards, perdu, pue, rat, pu, grves, graves, bah, taper, con, loup, franais, travaux, galre, chiant, gnant, incident, galérer, chelou, perdre, foutre, morte, tard, mauvais, loin, manque, connard, problmes, tape, odeur, ptn, putain, problèmes, marre, chier, fou, horrible, merde, casse, honte, bizarre, galère, peine, problème.

Table 2.13.: The Translated French seed-words from the Tweets' adapted English seed-words list of Table 2.9

Positive Translated Seed-words
amour, plaire, bien, gagner, lol, espoir, meilleur, merci, drôle, haha, dieu, surprenant, amusement, beau, mignon, cool, parfait, magnifique, ok, spécial, optimisme, content, félicitations, excellent, rêves, bonheur, hehe, positif, fantastique, danse, correct, fabuleux, supérieur, chanceux, doux, relaxant, heureux, génial, gentil, rire, refuge, merveilleux, yay, profitant.
Negative Translated Seed-words
souffrant, foutu, merde, bordel, haine, mauvais, casser, nul, pleurer, putain, triste, stupide, mort, douleur, malade, wtf, perdu, pire, échec, ennuyé, effrayé, blesse, peur, bouleversé, brisé, décédé, coincé, ennuyeux, horrible, négatif, malheureux, inférieur, malheureusement, pauvre, besoin, chiant, mal, méchant, manqué, douloureux, seul, connerie, enfer, fatigué, vilain.

2.4.2.4. English Book Reviews' seed-words

For the seed-words lists creation of book reviews domain, in English language, two lists of most common positive and negative words are automatically collected from the annotated book reviews as by Blitzer et al.[30] [Blitzer, Dredze,

[30]Book reviews from Multi-Domain Sentiment Dataset by http://www.cs.jhu.edu/ mdredze/-datasets/sentiment/index2.html

and Pereira 2007]. Then, after removing the stop-words, 40 words, which are the more relevant to the book reviews context, with strong sentiment orientation, are manually selected from each previously described list, as positive and negative seed-words adapted to the book domain. Table 2.14 shows the extracted seed-words.

Table 2.14.: The lists of Positive and Negative seed-words extracted from book reviews, in English language.

Positive	**Negative**
amazing, awesome, beauty, charm, enjoy entertain, excellent, extraordinary fabulous, fantastic, fascinate, favorite fun, good, great, happy, hilarious humor, incredible, informative, insightful inspirational, interesting, intriguing, joy like, love, magical,masterpiece, nice perfect, perfection, perfectly, positive recommend, recommendation, superb touching, wonderful, worth	angry, annoy, bad, bore, bother depress, disappoint, disturb, dull dumb, endless, fail, frustration garbage, hate, heavy, irritate mislead, mistake, negative, pain pathetic, pointless, poor, poorly sad, shame, sick, silly, struggle stuck, stupid, superficial, tedious terrible, unfortunate, unnecessary useless, waste, worse

2.4.3. Automatic extraction of adapted seed-words

For the purpose of a full automated creation of seed-words, we handle the distribution and interconnection of words in a word embeddings model in an attempt to locate the seed-words within. The following sections cover an explanation of the proposed method of an automatic extraction of seed-words.

2.4.3.1. Terms clustering within a word embeddings model

The first step to automatically create an adapted to domain sentiment seed-words list is the creation of a word embeddings model based on text extracted from a certain domain (e.g. microblogs). Then, a clustering is suggested to be employed on the created word embeddings model. Clustering (or Cluster analysis) is the task of segmenting a set of objects into partitions where elements in the same group (called cluster) are more similar to each other than those in other clusters [Allahyari, Pouriyeh, Assefi, et al. 2017]. Text clustering can be applied on many levels of granularity: documents, paragraphs, sentences and words. In our work, clustering is applied on words level, with the purpose of a simple reduction of massive word embeddings to centroids of words, since our goal is to locate a group of words, within the embeddings, with the characteristics of seed-words. For that purpose, K-means clustering [Hartigan and M. A. Wong 1979]

is suggested, as it is a simple unsupervised machine learning algorithm, and it demonstrated its effectiveness to clustering quality [Shipeng Li, Zeng, Ke, et al. 2012].

2.4.3.2. Clusters' content filtering

The dense center of each cluster is our extraction target, since it represents the part with the most similar words in the cluster. Therefore, we suggest a first method presented in Algorithm 2, and we call it method A, to extract the clusters' centers by first, calculating the similarity between all cluster's words using cosine similarity measure. Then, by getting the median of these similarity values in each cluster, and selecting the words having a higher similarity score than the median of that cluster.

A second method is proposed and presented in Algorithm 3, we call it method B, based on the method A but with additional steps (marked in red). In method B, we take into consideration the number of high similarity connections these words have with other words in the cluster, and not just the fact of a high similarity existence between them. Therefore we calculate the number of "high similarity connections", that these words have with other words in their cluster, and then we extract the ones having the number of connections higher than the average number of connections in their cluster.

For distinguishing the words carrying a sentiment in the extracted clusters' dense center, by methods A and B, SentiWordNet [Baccianella, Esuli, and Sebastiani 2010] is used to detect each word's objectivity, positivity and negativity score. From the previously selected words (the dense center of the clusters), we select positive and negative words, therefore, we extract the sentimental centered part of each cluster. As a result, the method A would give a positive seed-words' list Pos_A, and a negative seed-words' list Neg_A, and the same for method B with Pos_B and Neg_B.

A third method is also suggested and presented in Algorithm 4, we call it method C, also based on method A but with an additional step. We calculate the entropy of the average objectivity, positivity and negativity in SentiWordNet, with the Equation 2.11 associated to each word in the lists extracted in method A. Then, we extract the words with entropy lower than 0.8, since those words tend to hold a less ambiguous sentiment than others. As a result, two lists are created Pos_C and Neg_C.

$$H(p) = -\sum_{j} p_j \log_2 p_j \tag{2.11}$$

where p is the probability of the word to be in a class (positive, negative or neutral) and j is the number of words.

Algorithm 2: The method A of cluster dense center extraction.

Result: Two list of positive and negative seed-words from dense centers of all clusters, as Pos_A and Neg_A

Prerequisites: the clusters of a word embeddings model as $Clusters$, SentiWordNet.

for *each* Cl *in* $Clusters$ **do**
- $ListSim \leftarrow$ List of pairs of all Cl words with their similarities Sim calculated using cosine measure;
- $medianSim \leftarrow$ Median of Sim in $ListSim$;
- **for** *each* Sim *in* $ListSim$ **do**
  - **if** $Sim > medianSim$ **then**
    - $w \leftarrow$ the pair of words connected by that Sim;
- $Words \leftarrow Words + w$;

$Pos\_A \leftarrow$ filtering positive words from $Words$ by SentiWordNet;

$Neg\_A \leftarrow$ filtering negative words from $Words$ by SentiWordNet;

Algorithm 3: The method B of cluster dense center extraction.

Result: Two list of positive and negative seed-words from dense centers of all clusters, as Pos_B and Neg_B

Prerequisites: the clusters of a word embeddings model as $Clusters$, SentiWordNet.

for *each* Cl *in* $Clusters$ **do**
- $ListSim \leftarrow$ List of pairs of all Cl words with their similarities Sim calculated using cosine measure;
- $medianSim \leftarrow$ Median of Sim in $ListSim$;
- $ListCloseSim \leftarrow$ List of pairs of all Cl words with $Sim > medianSim$;
- $AvgCloseSim \leftarrow$ Average number of close connections;
- **for** *each* $CloseSim$ *in* $ListCloseSim$ **do**
  - **if** $CloseSim > AvgCloseSim$ **then**
    - $w \leftarrow$ the pair of words connected by that $CloseSim$;
- $Words \leftarrow Words + w$;

$Pos\_B \leftarrow$ filtering positive words from $Words$ by SentiWordNet;

$Neg\_B \leftarrow$ filtering negative words from $Words$ by SentiWordNet;

2.4.3.3. Experiments of automatic seed-words extraction

The previously explained method is tested through an example of microblogs' domain in English language. For the word embeddings model creation, the corpora of Sentiment140 [Go, Bhayani, and L. Huang 2009], of annotated tweets, is used with Gensim[31] framework for Python. According to [Mikolov, K. Chen, Corrado,

[31] https://radimrehurek.com/gensim/index.html

Algorithm 4: The method C of cluster dense center extraction.

Result: Two list of positive and negative seed-words, as Pos_C and Neg_C

Prerequisites: The seed-words extracted by method A, as Pos_A and Neg_A.

$Pos\_C \leftarrow$ empty;
$Neg\_C \leftarrow$ empty;
for *each* Pos_word *in* Pos_A **do**
    $entr \leftarrow$ the entropy of Pos_word;
    if $entr < 0.8$ **then**
        $Pos\_C \leftarrow Pos\_C + Pos\_word$;
for *each* Neg_word *in* Neg_A **do**
    $entr \leftarrow$ the entropy of Neg_word;
    if $entr < 0.8$ **then**
        $Neg\_C \leftarrow Neg\_C + Neg\_word$;

et al. 2013], Skip-Gram is more efficient in presenting infrequent words than CBOW in word embeddings, therefore, the Skip-Gram architecture is generally more used in sentiment analysis researches [Godin, Vandersmissen, De Neve, et al. 2015][Ay Karakuş, Talo, Hallaç, et al. 2018].Thus, Skip-Gram architecture of word2vec [Mikolov, K. Chen, Corrado, et al. 2013] is selected, and as for the parameters, the models are trained with: word representations of dimensionality 400, a context window of one and negative sampling for five iterations (k = 5).

Then the seed-words lists are created following the procedures previously explained, and as a result, two lists are extracted by the method A: Pos_A of 1972 words and Neg_A of 1959 words, and two lists by the method B: Pos_B of 961 words and Neg_B of 1093 words. And finally, two lists are created by the method C: Pos_C of 15 positive words and Neg_C of 25 negative words, shown in Table 2.15.

Table 2.15.: The seed-words automatically extracted from English language tweets employing the method C.

Positive seed-words	morality, elegance, admirable, jest, majestic, jesting, legendary, respected, engaging, props, bliss, excellent, deserts, greatest, jesting
Negative seed-words	malice, thorny, naproxen, dishonest, demonic, mislead, distressing, depraved, mischief, schlep, sheltered, cad, frigid, brokenhearted, paranormal, grotty, solace, mediocre, snotty, filthy, horrid, messy, gloomy, nasty, miserable.

2.4.4. Word embeddings models for Sentiment Intensity prediction

The suggested method of sentiment intensity prediction requires word embeddings models in addition to the lists of seed-words, therefore, we created several models of different domains and languages for testing purposes.

The word embeddings' training datasets of tweets is extracted from the archived Twitter streams[32], which is a collection of JSON[33] format data from the general Twitter stream, available for the purposes of research, history, testing and memory. This collection contains tweets in many languages, it allowed the extraction of tweets in the three languages: Arabic , French and English. The extracted files of archived Twitter streams were chosen randomly, dated between the years 2012 and 2017. A pre-processing is applied on the three corpora, to improve their usefulness:

1. The tweets' corpora is tokenized.
2. The user names and hyperlinks are replaced by $uuser$ and $http$.
3. The emoticons and emojis are replaced by *positive_emoji*, *negative_emoji* or *neutral_emoji* according to their polarity, based on a manually created list.
4. Some characters and punctuations were removed.
5. And also, the duplicated tweets were eliminated.

As a result one billion tweets in English language were extracted, in addition to 238 million tweets in Arabic language and 48 million tweets in French language.

As for the creation of the word embeddings model for the book-reviews domain, the training datasets is based on more than 22 million Amazon's book reviews [R. He and McAuley 2016], created after applying a similar pre-processing to the corpora as for the tweets.

For the purpose of learning word embeddings from the previously prepared corpora (which is raw text), Word2Vec [Mikolov, K. Chen, Corrado, et al. 2013] is selected. Always relying on [Mikolov, K. Chen, Corrado, et al. 2013]'s notion, considering that Skip-Gram is more efficient in presenting infrequent words than CBOW in word embeddings, therefore it is more efficient using it in sentiment analysis researches, the same method and parameters in Section 2.4.3.3, with Skip-Gram architecture, are applied to create the word embeddings models.

[32] https://archive.org/details/twitterstream
[33] JavaScript Object Notation is an open-standard file format

By applying the previously mentioned strategy to the datasets, three tweets word embeddings models are created with a vocabulary size of 9 million words for the Arabic model, 5 million words for the English model and 683 thousand words for the French model. And a book-reviews' word embeddings model is created with a vocabulary size of more than 2.5 million words.

2.4.5. Sentiment prediction experiments and results

In Section 2.4.5.1, we present the experiments and results of sentiment intensity prediction, obtained by the different lists of adapted to microblogs' seed-words, in English and Arabic languages. Another level of experiments is presented in Section 2.4.5.2, where we test the ability of adapted seed-words to predict a sentence polarity prediction as positive, negative or neutral. These experiments are applied on tweets in English, Arabic and French languages, in addition to the experiments on book reviews in English language, in Section 2.4.5.3. The experiments will be then followed by a discussion section, to interpret the methods results.

2.4.5.1. Sentiment Intensity prediction in Tweets

For the experiments in the domain of tweets, the used datasets are the ones provided by SemEval-2016. For the tweets of General and Mixed polarity in English language, Tables 2.16 and 2.17 show a comparison between the results of the proposed method with the different created seed-words lists (the semi-automatically created list and the automatically created lists), in addition to a comparison with Turney's seed-words [Turney 2002] list of Table 2.8, and the method applied in Section 2.3 (combining lexicons and search engines methods). The best results in Table 2.16 are achieved by the use of the automatically extracted seed-words of method A, followed by the semi-automatically extracted seed-words method. But the best results in Table 2.17 are achieved by the use of the semi-automatically extracted seed-words, and the automatically extracted seed-words of method A was not able to surpass it, which can be cause by the nature of mixed polarities in the phrases what makes the sentiment prediction more challenging for automatically extracted seed-words.

For the tweets in Arabic language, Table 2.18 shows a comparison between the results of the sentiment intensity prediction with the two created seed-words lists (the semi-automatically created list and the translated from the English language list), in addition to a comparison with the method applied in Section 2.3. The best results in Table 2.18 are achieved with the seed-words translated from English of Table 2.9.

Table 2.16.: Sentiment Intensity results with General English tweets using different methods, and seed-words lists.

Method	Kendall	Spearman
Lexicons + Search Engine	0.345	0.508
Word2vec + Turney's seed-words	0.312	0.455
Word2vec + Semi-Auto New seed-words	**0.442**	**0.615**
Word2vec + Auto New seed-words A	**0.465**	**0.656**
Word2vec + Auto New seed-words B	0.416	0.595
Word2vec + Auto New seed-words C	0.414	0.589

Table 2.17.: Sentiment Intensity results with Mixed English tweets using different methods, and seed-words lists.

Method	Kendall	Spearman
Lexicons + Search Engine	0.422	0.590
Word2vec + Turney's seed-words	0.300	0.442
Word2vec + Semi-Auto New seed-words	**0.432**	**0.598**
Word2vec + Auto New seed-words A	0.292	0.431
Word2vec + Auto New seed-words B	0.227	0.340
Word2vec + Auto New seed-words C	0.234	0.345

Table 2.18.: Sentiment Intensity results with Arabic tweets using different Methods, and different seed-words Lists.

Method	Kendall	Spearman
Lexicons + Search Engine	0.424	0.583
Word2vec + Semi-Auto New seed-words	0.405	0.562
Word2vec + Translated From English seed-words	**0.464**	**0.633**

2.4.5.2. Sentiment Polarity prediction in Tweets

The method proposed to predict the sentiment polarity of an entire tweet is based on calculating the whole tweets' sentiment intensity, and the tweet's sentiment intensity score is equal to the average of its words sentiment intensity, excluding stop words. Based on that, tweets with scores greater than 0.01 are considered positive, and the ones with scores lower than -0.01 are considered negative and the tweets of scores between -0.01 and 0.01 are neutral.

It should be noted that for this procedure, and before being segmented into

tokens, each tweet is cleaned by removing links, user names, numeric tokens and characters except the emoticons (e.g *:)*). In addition, all emoticons and emojis are replaced by *positive_emoji*, *negative_emoji* or *neutral_emoji* according to their polarity, predefined in a manually created list.

For the experiments of tweets sentiment polarity prediction in English and Arabic languages, the dataset is provided by SemEval-2017 task-4[34][Rosenthal, Farra, and Nakov 2017], with 12284 tweets in English language, and 671 tweets in Arabic language. Table 2.19 shows the results of sentiment polarity prediction experiments, in English language, using different seed-words lists: Turney's, the semi-automatically extracted list, and the three automatically extracted lists. The best results are achieved using the semi-automatically created list.

Table 2.19.: Sentiment Polarity results with English Language tweets using different seed-words Lists.

Seed-words List	**f1_measure**	**Recall**	**Accuracy**
Turney's	0.416	0.413	0.361
Semi-Auto	**0.579**	**0.574**	**0.466**
Auto List A	0.499	0.505	0.449
Auto List B	0.439	0.450	0.390
Auto List C	0.521	0.515	0.429

Table 2.20 shows the results of sentiment polarity prediction experiments, in Arabic language, using two seed-words lists: the semi-automatically created list and the translated from semi-automatically created English seed-words list. The best results are achieved using the translated list.

Table 2.20.: Sentiment Polarity results with Arabic Language tweets using different seed-words Lists.

Seed-words List	**f1_measure**	**Recall**	**Accuracy**
Semi-Auto	0.391	0.383	0.346
Translated from English	**0.487**	**0.472**	**0.419**

The results of our participation [Htait, Sébastien Fournier, and Bellot 2017] at SemEval-2017 Task4, using the previously explained method with semi-automatically created seed-words lists, for English and Arabic languages, are presented in Table 2.21 and 2.22. For the English language, the word embeddings model by [Godin, Vandersmissen, De Neve, et al. 2015] was used, since the workshop took place before the creation of our word embeddings models. Godin's model is a word2vec model trained on 400 millions tweets in English language and it

[34]http://alt.qcri.org/semeval2017/task4/

has word representations of dimensionality 400. For the Arabic language, the created word embeddings model was based on only 42 millions tweets in Arabic language from archived Twitter streams [35], applying the same methods and parameters as in Section 2.4.4.

Table 2.21.: Our participation's results at semEval2017 Task 4 subtask A - for English Language.

English	**Team**	**f1_measure**	**Recall**	**Accuracy**
	Best Results	0.685	0.681	0.658
	Our participation	0.561	0.571	0.521

Table 2.22.: Our participation's results at semEval2017 Task 4 subtask A - for Arabic Language.

Arabic	**Team**	**f1_measure**	**Recall**	**Accuracy**
	Best Results	0.610	0.583	0.581
	Our participation	0.469	0.438	0.445

For the experiments of tweets' sentiment polarity classification in French language, the dataset is provided by DEFT-2018 workshop[36] [Claveau, Minard, Cellier, et al. 2018]. The dataset is a 10k records of tweets about french public transport, annotated as positive, negative, neutral and mixed polarity. But since our suggested method covers only three polarity classes, a set of 9335 tweets is extracted from DEFT-2018 dataset, with the polarities positive, negative and neutral.

Table 2.23 shows the results of sentiment polarity classification experiments, in French language tweets, using two seed-words lists: the semi-automatically created list and the translated from English seed-words list. The best results are achieved using the semi-automatically created list.

Table 2.23.: Sentiment Polarity results with French Language tweets using different seed-words Lists.

Seed-words List	**f1_measure**	**Recall**	**Accuracy**
Semi-Auto	**0.578**	**0.566**	**0.494**
Translated from English	0.485	0.433	0.427

For our participation in DEFT 2018 task 2 challenge [Htait 2018], we employed the following additional methods:

[35] https://archive.org/details/twitterstream
[36] https://deft.limsi.fr/2018/

- Extending the lists of adapted seed-words using NormAFE[37] (explained in Chapter 3), a tool that creates thesaurus for microblogs normalization, in a form of pairs of misspelled word with its standard-form word, in the languages: Arabic, French and English. Using NormAFE, we extracted the misspellings of our seed-words and we add them to the original list. For example some of the misspellings of the word $magnifique$ (as $magnificent$ in French) are : $magnfique$, $magnif$, $magnifi$, $magnifiiiique$, $magnifiiique$, $magnifiique$, $magnifik$, $magnifike$, $magnifiq$, etc. The result of this procedure is a 1358 positive seed-word and 1330 negative seed-words.

- Since DEFT 2018 task 2 included a fourth polarity class *MixPosNeg* (mixed polarity), which is not covered by our current system, we decided to consider the tweets of a certain polarity with an emoji of an opposite polarity as a mixed polarity tweets, or more specifically as $sarcasm$. An example in the following tweet where the person is mostly complaining about a negative event and then he adds a smiley at the end of his tweet which shows sarcasm and the expression of mixed sentiment : " ... des vieux types mn clc dans le métro et un pigeon s'est lacher sur ma veste ... ☺".

Table 2.24 shows the official results of DEFT 2018 task 2 challenge, where each of our runs is explained below:

- **Run_1**: We used the extended seed-words by NormAFE, but without adding the fourth class of *MixPosNeg* prediction. Therefore, the results contain only three classes *Positive*, *Negative* and *Neutral*.

- **Run_2**: We used the extended seed-words by NormAFE, with the fourth class of *MixPosNeg* prediction added to the results.

- **Run_3**: We used the semi-automatically created original list of seed-words, but without adding the fourth class of *MixPosNeg* prediction. Therefore, the results contain only three classes *Positive*, *Negative* and *Neutral*.

- **Run_4**: We used the semi-automatically created original list of seed-words, with the fourth class of *MixPosNeg* prediction added to the results.

The Run_3 achieved an F1-measure equals to 0.64, as the best result between our four runs. The results show that using an extended version of the seed-words decreased the F1-measure from 0.64 in Run_3 to 0.62 in Run_1. Also, our method to predict the fourth class $MixedPosNeg$ decreased the F1-measure to 0.63 in Run_4 and to 0.61 in Run_2.

[37] https://github.com/amalhtait/NormAFE

Table 2.24.: The results at DEFT-2018 Task 2 - for French Language.

	F1-measure
DEFT Best Result	0.82288
Run_1	0.62539
Run_2	0.61622
Run_3	**0.64524**
Run_4	0.63939

2.4.5.3. Sentiment Polarity prediction in Book reviews

The reviews have different characteristics than the tweets or the microblogs in general, since the tweets are often informal and unstructured. Therefore, in this section, we apply our method on the book reviews domain to test it capacity of sentiment polarity prediction in a different domain than microblogs.

But first, to test the polarity prediction method in book reviews, we created an annotated corpora by randomly extracting 100 positive reviews (of rate=5) and 100 negative reviews (of rate=1), from Amazon's book reviews [R. He and McAuley 2016]. The created dataset includes different size reviews, between 20 and 700 words, with an average of words number equals to 90 words, and a median equals to 53.5 words.

Then, due to the difference in writing style between reviews and tweets, instead of calculating the average of sentiment intensity of all the words, as in the method applied for tweets, the review is filtered by selecting only verbs, nouns and adjectives, and the average of their sentiment intensity is calculated. Then, the review is considered positive if its sentiment intensity is greater than 0.01, else it is considered a negative review. The results are presented in Table 2.25.

Table 2.25.: The results of book reviews adapted seed words in English language.

Results	**F1_measure**	**Recall**	**Accuracy**
Semi-Auto extracted seed-words	0.703	0.483	0.724

2.4.6. Discussion

The experiments, presented in the previous section, showed the capacity of our suggested method, based on adapted seed-words and word embeddings model, to predict sentiment intensity in phrases and terms, and sentiment polarity in sentences. For the English language experiments, the best results were achieved using the semi-automatically created adapted seed-words, even though the automatically created lists achieved also good results which can open the doors to

future improvements to the automatic method.

As for the Arabic language experiments, the best results were not achieved by the semi-automatically created list of seed-words, but by the translated from English language seed-words, and it can be caused by the method applied for the lists filtering. In the process of manual seed-words lists creation, as mentioned in Section 2.4.2.2, the lists of most common words in positive and negative tweets are filtered by a list of 240 positive and negative predefined words in standard Arabic language [Mohammad and Turney 2013]. Therefore, the extracted seed-words lists includes the words of intersection between a list of most common words in tweets and a standard Arabic list of words with strong sentiment orientation. The purpose was to make a seeds-words list adapted to tweets in standard Arabic language. Such filtering caused the elimination of common slang words carrying strong sentiment polarity, like تجنن that has the meaning of "*goes mad*" in standard Arabic language, but it is considered a slang word and has the meaning of "*amazing*" in Lebanese Arabic dialect. Therefore, the employment of strictly standard Arabic words into the prediction of sentiment intensity of tweets in different Arabic dialects reduced the effectiveness of the "*Adapted*" to domain and language seed-words. More tests and experiments would be necessary to prove that the filtering is the cause of decremented results.

For the French language experiments, the best results were achieved using the semi-automatically created adapted seed-words. As for our participation at DEFT-2018, our results are for predicting only three classes : *Positive*, *Negative* and *Neutral*, in a challenge where the prediction of four classes is required (*Positive*, *Negative*, *Neutral* and *MixedPosNeg*). Therefore, and as a future improvement plan, we will test a new method to predict the fourth class $MixedPosNeg$, by predicting the polarities of tweet segments and detecting opposite polarities in the same tweet.

Finally, a Software is created, based on our suggested method, and shared as open source for predicting sentiment intensity in words of:

- General tweets in English and Arabic languages.
- Public transport tweets in French language.
- Book reviews in English language.

The tool's name is Adapted Sentiment Intensity Detector (ASID) and it can be downloaded from github[38].

[38] https://github.com/amalhtait/ASID

2.5. Conclusion

In this chapter we have proposed several methods for sentiment intensity and sentiment polarity prediction, in tweets and book reviews, applied in English, French and Arabic languages. The experiments showed the effectiveness of the adapted to domain seed-words and word embeddings models method, also it proved its ability to be easily applied on different domains and in different languages. In addition, our participation in several workshops showed good results compared to other participants results, since our method is based on a semi-supervised approach requiring a very small amount of annotated data.

In the next chapter, we propose an unsupervised method to create normalisation thesaurus, for the purpose of increasing the effectiveness of the previously proposed sentiment analysis methods, since the bad quality of a text (typos, misspellings, informal words, etc) can create several obstacles in the way of text processing and therefore sentiment analysis.

Chapter 3: Automatic Creation of Thesaurus for Text Normalisation

Summary

3.1. Introduction

Twitter and other social web services are considered a source of large-volume real-time data, which make them highly attractive for information extraction and text mining. Unfortunately, the quality of their text, with the typos, misspellings, informal words, phonetic substitutions and word shortening creates huge obstacles in the way of text processing. Therefore, normalisation techniques are a necessity to correct and make more sense of the social web resources.

In the work presented in Chapter 2 regarding the sentiment analysis prediction in tweets and reviews, we faced several challenges due to informal language of the content. Therefore, in this chapter, we present a text normalisation proposal to solve the previously mentioned challenges. It is based on an unsupervised automatic creation of normalisation thesaurus, applied in several languages: Arabic, French and English [Htait, Sébastien Fournier, and Bellot 2018]. In general, a thesaurus is a dictionary of synonyms and antonyms, but in this chapter, we handle normalisation thesaurus, which are dictionaries of misspellings, that can be used in the process of bringing or returning words to their normal and standard state.

This work is inspired by [Sridhar 2015], an unsupervised method for text normalisation using distributed representations of words, or word embeddings, but we develop their approach with some improvements, and a variety of tested languages, in addition to a much larger datasets for the word embeddings models training. In this chapter, the method is first presented, followed by experiments and evaluations. In addition, a tool (NormAFE[39]) is built to create thesauruses for text normalisation, and is available as open source including the resources: three word embeddings models, and three normalisation thesauruses, for the three languages.

3.2. Related Work

The primary approach in text normalisation was the noisy channel model [Shannon 1948], the approach aims to find $argmax P(S|T)$ where the misspelled text is T and its corresponding standard form is S, and that is by computing $argmax P(T|S)P(S)$, in which $P(S)$ is a language model and $P(T|S)$ is an error model. For many applications, there was a considerable energy to improve both models, with a result of improvement in overall system accuracy. For example, some researchers worked on a new error model for spelling correction, based on generic string to string edits [Brill and Moore 2000]. And others expanded the

[39] https://github.com/amalhtait/NormAFE

error model by analyzing a sample of texting forms to define frequent word formation processes in creative texting language. The noisy channel model in text normalisation showed effectiveness, but its methods are based on the assumption that a token $t_i \in T$ only depends on $s_i \in S$, ignoring the context around the token, which can cause ambiguity between words (e.g. $gooood$ was meant to be $good$ or God?).

Statistical machine translation (SMT) has been also used as a method for text normalisation, by treating the misspelled text as the source language, and the standard form as the target language. Similar work is found on phrase-based SMT model for text normalisation with bootstrapping the phrase alignment [Aw, M. Zhang, Xiao, et al. 2006]. Unfortunately, SMT approaches require training data that are often unavailable. Some researchers used speech recognition to solve text normalisation issue [Kobus, Yvon, and Damnati 2008]. They converted the input text tokens into phonetic tokens, then restored them to words using phonetic thesaurus. Others used a classifier to detect misspelled words, and generated correction possibilities based on morphophonemic similarity [Han and Baldwin 2011]. But these methods need large-scale of annotated training data, which limits their adaptability to new domains and languages.

To overcome the limitations of previously cited methods, a technique is applied to learn distributed representation of words (known as word embeddings), and to capture distributional similarity between words in a unsupervised manner. As a result, each word will be represented by a numeric vector of high-dimensionality, encoding many linguistic regularities and patterns, also syntactic and semantic word relationships. Due to this representation, words with semantic similarity are represented by similar vectors, and a misspelled word might be also represented by a similar vector as its standard-form word.

Sridhar et al. [Sridhar 2015] were first to propose that method with a training dataset of 27356 English SMS phrases. Their research was the base of several similar work in Portuguese [Bertaglia and Nunes 2017], Turkish [Eryiğit and Torunoğlu-Selamet 2017] and Chinese [Yan, Y. Li, and Fan 2017], but never in Arabic nor French. In addition, none of these work is open source, and they did not share the word embeddings models, nor the lexicons or thesauruses. Also, all their work was based on relatively small datasets. For example, Bertaglia's work [Bertaglia and Nunes 2017] was focused on products reviews, that are slightly effected by the misspelling errors, the slang words and the typo errors, compared to the tweets, which leads to a much more effective work in micro-blogs' normalisation. Also Bertaglia's work [Bertaglia and Nunes 2017] was based on a dataset of only 86 thousand products reviews and an unknown small amount of tweets in Portuguese. And like the rest of the previously cited researchers, the datasets and word embeddings models were not publicly shared.

The proposed method, in this chapter, focuses on tweets as a dataset resource, for their richness in misspellings and slang words. In addition, the proposed method is tested with three languages: Arabic, French and English. And as for the dataset size, a large tweets corpora is employed in the training of word embeddings models: one billion tweets in English language, 238 million tweets in Arabic language and 48 million tweets in French language.

3.3. General overview of the proposed method

Word embeddings models allow to capture the nearest neighbors of a certain word X using the cosine distance between the dimensional vector of that word X and the dimensional vector of each word in the model. The example from a word embeddings model based on tweets in French language, presented by the t-Distributed Stochastic Neighbor Embedding (t-SNE) with two dimension (2D) in Figure 3.5, shows that most of the nearest neighbors of the word $alors$ ($then$ in French) are not real French words but the misspellings of the word $alors$, such as: $alrs$, $allrs$, alr, $alord$, $alirs$, $allors$ and $alorq$, in addition to some other words, such as: $sachant$ ($knowing$ in French). And another example, in Arabic language at Figure 3.6, where the word بارد ($cold$ in Arabic) has the nearest neighbors as its misspellings and not real Arabic words, like: باااااارد , باارد , بآرد and بارررد. And a similar example presented in Figure 3.7, in English language, for the word $will$ and its nearest neighbors: *wlll, wiill, wiil*, etc.

Therefore, and based on the previous observations, a method for creating normalisation thesaurus is suggested, and it follows the below steps:

1. First, the creation of a word embeddings model with micro-blogs text as training data, since the micro-blogs includes many typos, misspellings, informal words, etc, which gives better chances of including the maximum possible of misspellings in the created normalisation thesaurus.

2. Then, we specify a list of the most common words in a standard-form (without misspellings), lets call it *base-list*, to find their misspellings. The base-list can be chosen depending on the needs to the thesaurus, as it can be of general or specific domain. For example, for book-reviews domain the list would contain the words: book, read, recommend, etc. The shared software "NormAFE" offers the possibility of creating new normalisation thesaurus based on different base-list.

3. Next, we look for most similar words of each element of the *base-list* within the word embeddings model's space. The similarity between words in the word embeddings model is based on the cosine distance measure between

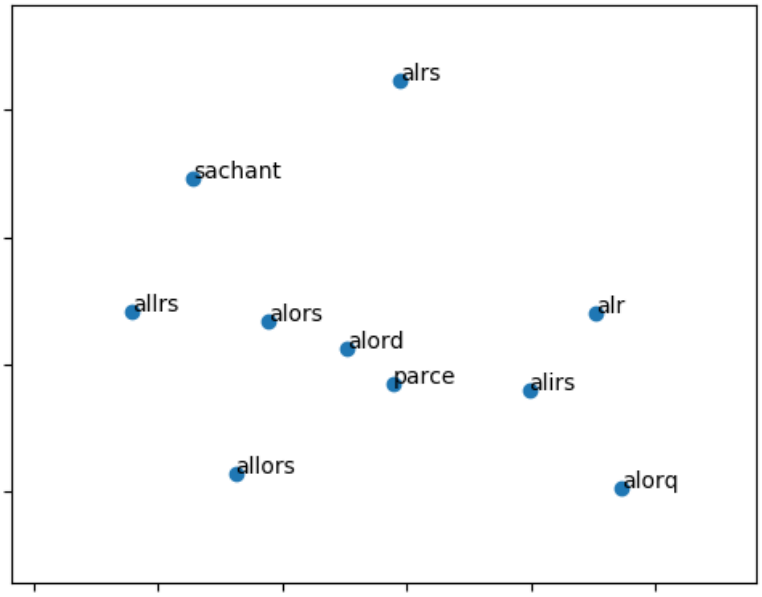


Figure 3.5.: The French word "*alors*" (then) with its nearest neighbors in the word embeddings model space, by t-SNE of 2D, based on tweets of French language

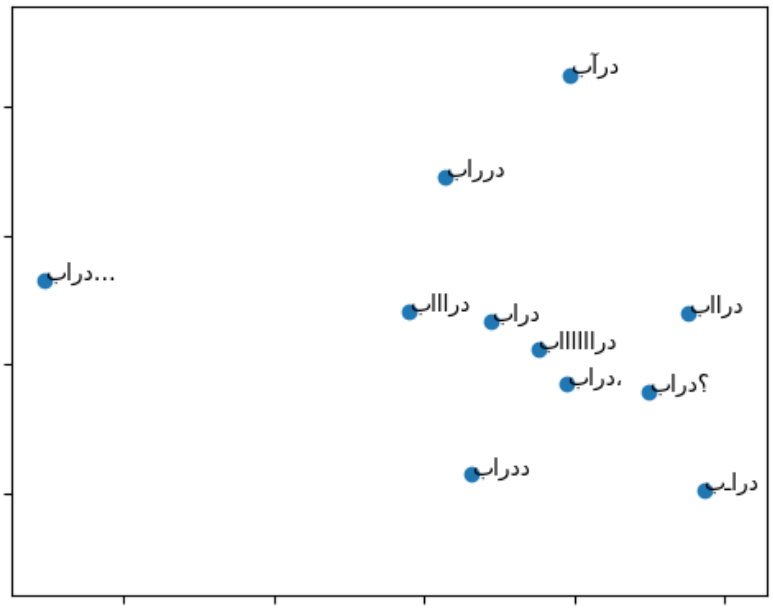


Figure 3.6.: The Arabic word (cold) with its nearest neighbors in the word embeddings model space, by t-SNE of 2D, based on tweets of Arabic language (letters showed from left to right).

the vectors of these words in the model. We suggest the use of the class *most_similar* (based on cosine distance measure), of Gensim framework,

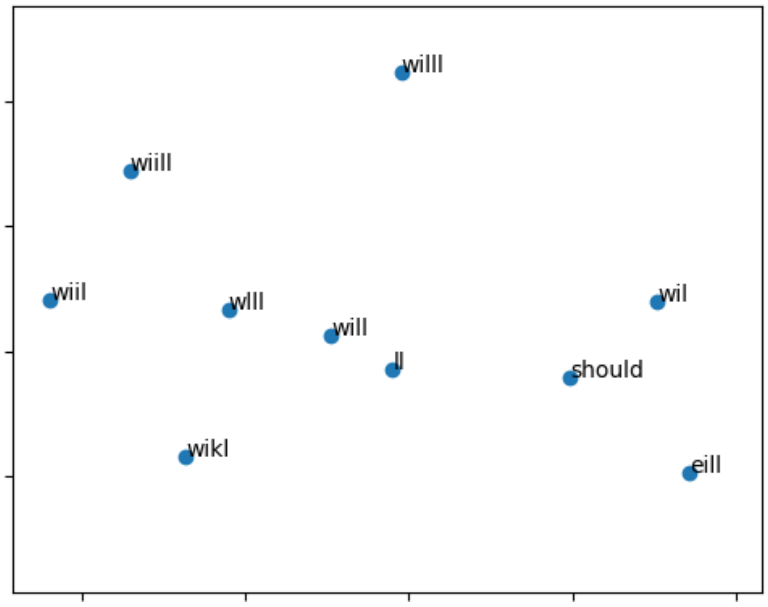


Figure 3.7.: The English word "*will*" with its nearest neighbors in the word embeddings model space, by t-SNE of 2D, based on tweets of English language

to give a list of most similar words to the standard-form word. To refine the results, *most_similar* class can be used with the antonym of the standard-form word as in the following example with the word *active*:

```
model.most_similar(positive=['active'], negative=['inactive'])
```

The antonym exclusion eliminates the possibility of extracting the word *inactive* as a similar word to the word *active*, since the list of the 5 most similar words of *active* is: *inactive*, *acitve*, *avtive*, *actuve* and *innactive*. For this step, we employ the Natural Language Toolkit[40] to find the antonyms of *base-list* words.

4. Finally, a filtering is necessary for the purpose of eliminating the errors in the list of possible misspellings. For example, the four most similar words of "*good*" in the word embeddings model space, are: *goood*, *goid*, *great* and *goos*. The word "*great*" should be eliminated, therefore, we suggest the use of Python's class *SequenceMatcher* to compare the previously collected similar words to the standard-form word. The idea of this method is to find the longest contiguous matching subsequence that contains no junk elements (or different elements). The same is then applied recursively to the pieces of the sequences to the left and to the right of the matching

[40] http://www.nltk.org/

subsequence. This method tend to give matches that *look right* to people[41]. An example of a normalisation thesaurus content is presented in Table 3.26, with the word *good*.

Table 3.26.: An example of a normalisation thesaurus content, in English language.

Misspelled	Standard-word
goood	good
goid	good
goos	good

3.4. Experiments

To proceed with the experiments, word embeddings models are required. In Chapter 2, Section 3, three sets of word embeddings models were created based on collected tweets from the archived twitter streams[42], in English, French and Arabic languages. The three models were created with a vocabulary size of 9 million words for Arabic model, 5 million words for English model and 683 thousand words for French model. These three word embeddings models are the base of our thesaurus creation in the following section.

3.4.1. Creating normalisation thesaurus in English, French and Arabic languages

To have as much similar evaluation as possible between the three languages, a sample of 50 standard-form words highly positive or negative is selected, as a $base - list$, for each language, since sentiment words are usually used in micro-blogs at same high frequencies, in most languages. The 50 chosen standard-form words, for each language, are chosen randomly from the semi-automatically created seed-words, presented in Chapter 2 at Tables 2.9, 2.10 and 2.12.

Then, the previously described method of similar words extraction and filtration is applied to our $base - lists$. The Gensim framework's class "$most\_similar$" extracts, by default, the first 10 most similar words, but we are able to change the default number 10, which can lead to the creation of smaller or larger thesauruses, like in the below example:

- The list of 5 similar words of *good* is: goood, goid, gooood, goooood, gud.
- The list of 15 similar words of *good* is: goood, goid, gooood, goooood, gud, goos, gooooood, gpod, great, gopd, giod, gooooooood, cargood, g00d, goooooooood.

[41] https://docs.python.org/2/library/difflib.html
[42] https://archive.org/details/twitterstream

The second list is larger and richer in words, but it includes unwanted words like *great* and *cargood* (since these words are not misspellings of the word *good*). Therefore, and as part of the evaluation, four normalisation thesauruses are created for each language, with the number of extracted similar words is equal to 5, 25, 50 and 100. As a result, the created thesaurus for the Arabic language reached the size of 2053 pairs (of misspelled words with their standard-form word) when selecting 100 most similar words, for the French language the size of 500 pairs, and for the English language the size of 2776 pairs.

3.4.2. Evaluating the thesaurus' content

To evaluate the proposed method by thesaurus's content, a manual annotation is applied by checking the correct pairing between the misspelled words and their assigned standard-form words, and the annotation differentiate between two types of evaluation: *Correction* (e.g. *graet* and *great*) and *Normalisation* which includes the correction and the lemmatization (e.g. *shows* and *show*). Table 3.27 shows an example of the method of annotation, where the check-mark is a right correction or normalisation, and the x-mark is a wrong one.

Table 3.27.: An example of thesauruses annotation, where three examples from each language is selected (English, French and Arabic), and where the check-mark is a right correction or normalisation, and the x-mark is a wrong one.

Misspelled	Standard-word	Correction	Normalisation
gladd	glad	✓	✓
hates	hate	✗	✓
horrific	horrible	✗	✗
aiiiiiime (looooooove)	aime (love)	✓	✓
decevra (will disappoint)	decevoir (disappoint)	✗	✓
deballer (unpack)	deprimer (depress)	✗	✗
ممتاذ (misspelled excellent)	ممتاز (excellent)	✓	✓
اكرهه (hate him)	اكره (hate)	✗	✓
اهبل (dump)	غبي (stupid)	✗	✗

First, a briefing of the evaluation results, with all created normalisation thesauruses, are presented below:

1. For English language, an average of 96% in *Normalisation* success, and of 86% in *Correction* success.

2. For Arabic language, an average of 89.5% in *Normalisation* success, and of 83.7% in *Correction* success.

3. For French language, an average of 85% in *Normalisation* success, and of 73.6% in *Correction* success.

Then, the results of the evaluation are presented with more details in the graphs of Figure 3.8, consequentially from left to right in English, French and Arabic languages, where the percentage of successful *Correction* is the line in dashed blue and the percentage of successful *Normalisation* is the line in solid red, both calculated relatively to the variation of most similar words number (as 5, 25, 50 and 100) and the variation of the thesaurus size (the bars in grey).

The results in Figure 3.8 shows that the percentage of successful *Normalisation* is always higher than the percentage of successful *Correction*. Also, for English (left graph) and French (middle graph) languages, an increase in the percentage of successful *Correction* and *Normalisation* appears when the number of similar words extracted is between 5 and 25, followed by a continuous decrease. And for Arabic language (right graph), a sharp decrease with the percentage of successful *Correction* and *Normalisation* is observed, with the increase of similar words extracted number.

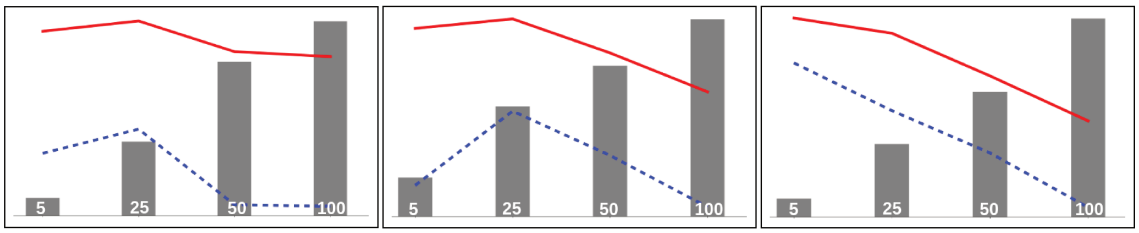


Figure 3.8.: The percentage of successful *Correction* (dashed blue) and *Normalisation* (solid red) in the test thesauruses, both calculated with a variation of most similar words size and a variation of the thesauruses size (the bars in grey), from left to right, in English, French and Arabic languages.

3.4.3. Evaluating the thesaurus' contribution in Sentiment Analysis prediction

The evaluation of the thesaurus by its content, in the previous section, shows promising results. But to prove the usefulness of the thesaurus in sentiment analysis, the English language normalisation thesaurus is evaluated by its influence on predicting sentiment polarity (positive, negative or neutral) in micro-blogs

messages.

The first experiments are performed on the method of Chapter 2, Section 3, with sentiment intensity prediction of tweets, based on adapted seed-words and word embeddings. Unfortunately, text normalisation did not change the prediction quality, and that can be justified by the following reasons:

- First, we apply the normalisation procedure on only the test data, therefore, the amount of normalisation in text can be insufficient to effect the sentiment intensity scores.
- Second, we employ word embeddings in the sentiment intensity prediction and in the normalisation thesaurus creation, which can neutralize the effect of the normalisation on the sentiment analysis procedure. For example the words *doctor* and *doctr* are neighbors in the word embeddings space, as explained in Section 3.3. To calculate the sentiment intensity of the word *doctor*, as presented in Chapter 2 Section 3, the average of cosine similarity between the word and the lists of positive and negative seed-words is calculated as $score_1$. And the fact that the word *doctr* is represented by a similar dimensional vector as *doctor*, therefore, the average of cosine similarity of the word *doctr* with the lists of positive and negative seed-words is very close to $score_1$. Therefore, replacing *doctr* by *doctor* would not have a big affect on sentiment intensity predicting scores.

Eventually, we decide to evaluate the English language normalisation thesaurus on a another sentiment analysis method, proposed by our team [Hamdan, Bellot, and Bechet 2015], not based on word embeddings models but on supervised machine learning (SVM), using the software Echo[43]. For Echo's training and testing, SemEval2014's datasets [Rosenthal, Nakov, Ritter, et al. 2014] are used: for training, the annotated (by sentiment polarity) training dataset of almost 10000 Tweets, and for testing the 1000 Live-Journal from 2014, the 2000 SMS from 2013 and the 3800 Tweets from 2013. The results[44] of Echo, predicting sentiment polarity of the testing data, are presented in Table 3.28. The first row is the baseline, where Echo runs without normalisation. Then, for the rest of the rows, the normalisation was applied, on the training and testing datasets, using four thesauruses, all based on the same list of 50 English sentiment standard-words, but differ in the number of most similar words chosen at the level of thesaurus creation (5, 25, 50 or 100), and as a result, these thesauruses differ in their size (since the size of a thesaurus increases with the increasing number of "most similar words").

[43] https://github.com/OpenEdition/echo

[44] The results are displayed with the f-measure value, a measure of a test's accuracy Powers 2011.

The results, in Table 3.28, show an increase in the capability of Echo to successfully predict sentiment polarity in micro-blogs messages, using the normalisation thesauruses. Also, they show that Echo achieves better results when increasing the thesaurus size, and best results in this evaluation is achieved with thesaurus size equals to 2776 pairs.

Table 3.28.: Results of Echo with SemEval2014's data, with a baseline of no normalisation, then with a normalisation applied using four thesauruses that differ in the number of most similar words and in their size.

Echo	**#Similar**	**DictSize**	**LiveJournal2014**	**SMS2013**	**Twitter2013**
baseline	-	-	0.58	0.55	0.55
+Dict_1	5	371	0.58	0.55	0.55
+Dict_2	25	1449	0.58	0.56	0.56
+Dict_3	50	2337	0.58	0.56	0.56
+Dict_4	100	2776	**0.59**	**0.56**	**0.56**

3.5. Discussion

Evaluating the method of thesauruses creation for text normalisation is applied by evaluating the thesauruses content and their effect when used with sentiment analysis procedures. The evaluation of the created thesaurus content showed an average in $Normalisation$ success of 96% in English language, 89.5% in Arabic Language and 85% in French Language. And the evaluation's results of employing these thesaurus in a sentiment analysis tool for micro-blogs messages, showed an increase in the tool's ability to predict the sentiment polarity of the messages.

An interesting observation is detected in the evaluation results in Section 3.4.2 where the percentage of $Normalisation$ and $Correction$ success decreases with the increasing size of the thesauruses. But, on the other hand, and based on the evaluation in Section 3.4.3, the effectiveness of the thesaurus (in sentiment analysis) increases with its size, independently from the percentage of success in $Normalisation$ and $Correction$.

In addition, the created thesaurus revealed other characteristics than the normalisation capacity. For example, by observing the Arabic language thesaurus, many pairs of dialect word with its standard-form word were found, some examples are in Table 3.29.

Table 3.29.: An example of Arabic language pairs of dialect word with its standard-form word in the normalisation thesaurus.

$DialectWord$ -	$DialectSource$ -	$StandardWord$
عبيط	Egypt Arabic	غبي (Stupid)
ضايقج	Gulf Arabic	ضايقك (bothered you)

3.6. Conclusion

In this chapter we presented an approach based on an unsupervised method for text normalisation using word embeddings, applied on Arabic, French and English languages. In addition, a tool (NormAFE) is built to create thesauruses for text normalisation, and is available as open source. The created thesauruses did not increase the effectiveness of the method applied in Chapter 2, but they proved their effectiveness with another sentiment analysis method, not based on word embeddings.

The next part of this manuscript concerns information retrieval and filtering, in the context of book search and book recommendation. It includes three chapters, where we start by presenting these domains backgrounds, followed by our contributions in these domains.

Part II.

Information Retrieval & Information Filtering

Chapter 4:
Background to Information Retrieval & Information Filtering

Summary

In this chapter, we briefly present the basic knowledge used over the second part of this manuscript, concerning Information Retrieval (IR) and Information Filtering (IF), two interfering domains. We start with the concepts, then, we present their main approaches by focusing on the ones used or mentioned in the following chapters. Finally, the evaluation measures employed in this part of the manuscript are presented and briefly explained.

4.1. Information Retrieval as Search systems

Definition: Information Retrieval (IR) is the field of dealing with unstructured data retrieval, mostly textual documents, in response to a query or a topic declaration, which may itself be unstructured (e.g. a natural language sentence) [Greengrass 2000]. The effectiveness of an IR system is based on its ability to retrieve a set of documents that answers the users need of information, which are labeled as relevant documents. The retrieved documents are often ranked by a relevance score.

The IR procedure uses the query, or the request of information, and matches it with the document representations. Then, the documents with best matching characteristics are considered relevant, and are proposed to the user. In brief, an IR system has to maintain three basic processes: the representation of the content of the documents, the representation of the user's information need, and the comparison of the two representations. As shown in Figure 4.9, the documents' representation passes through an off-line indexing process, that often includes parts of the documents. On the other side of the figure, there is the users' need of information or the query, which is the subject of a possible reformulation for a more effective information retrieval. Then a matching process is applied as a comparison between the query and the document representations for a ranked list retrieval of relevant documents, according to the IR system. The retrieved documents could then be used as an appliance of query formulation by feedback.

In Chapter 6, we present our suggested methods of sentiment analysis employment in the information retrieval field. For that purpose, various information retrieval concepts are adopted, and therefore briefly introduced in the following sections, like information retrieval models, methods for query formulation, and methods for retrieved rankings aggregation.

4.1.1. Information Retrieval models

Several models have been employed in IR systems to match documents and queries. The Boolean model was the first to be proposed in IR by [Lancaster

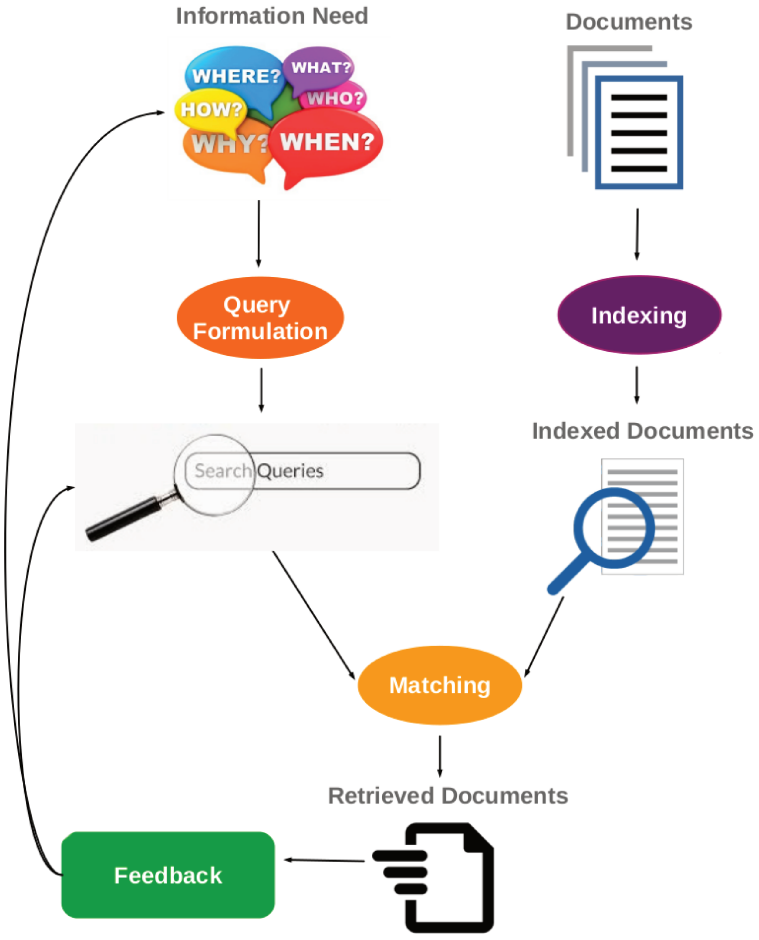


Figure 4.9.: Visualisation of the information retrieval procedure.

and Fayen 1973]. It is based on set theory and Boolean algebra, where the documents are sets of terms and the queries are Boolean expressions on terms joined by logical operators, such as "*OR*", "*AND*", and "*NOT*". In this model, the terms are evenly weighted and the retrieved documents are difficult to rank. Therefore, the pursuant models were oriented more toward ranking the retrieved documents, like Vector Space Model, Probabilistic Model, Divergence From Randomness Model and Language Model.

The **vector space model** [Salton, A. Wong, and C.-S. Yang 1975] is based on a cosine similarity measure between a query and each document, represented by their terms weight vectors. And the results of this measure expresses the ranking score of the documents. However, the vector model does not take into account the semantic relationships since a term can be expressed differently depending on the context.

The **probabilistic model** [Maron and Kuhns 1960][K. S. Jones, Walker, and Robertson 2000] is based on the probability of a document's relevance for a query calculation. And the matching probability score between the document d and the query Q is the ratio of the probability that a given document is relevant to a query Q, $p(d/Q)$, and the probability that it is irrelevant, $p(\bar{d}/Q)$, as presented in Equation 4.12.

$$RSV(Q,d) = \frac{p(d/Q)}{p(\bar{d}/Q)} = \sum_{i=1}^{t} log \frac{p_i \cdot (1-q_i)}{q_i \cdot (1-p_i)} \tag{4.12}$$

where p_i is the probability for a term t_i (of the query) to exist in d with d is a relevant document, and q_i is the probability for the term t_i to exist in d with d is an irrelevant document, and t is the total number of terms in the query.

The **Divergence from randomness model** (DFR) [Amati 2003] is based on term weights computing by measuring the divergence between a term distribution produced by a random process, within the collection, and the actual term distribution, within the document. Such measure gives different importance to words for describing the documents' content. The rank score of the document d with regard to a query Q is computed as:

$$RSV(Q,d) = \sum_{t_i \in Q} tf_{i,Q} \cdot w_i \tag{4.13}$$

where $tf_{i,Q}$ is the frequency of the term t_i in the query, and w_i is the weight of the term t_i calculated in the below Equation:

$$w_i = [-logP(t_i|C)] \cdot [1 - P(t_i|d)] \tag{4.14}$$

Where $P(t_i|C)$ is the distribution probability over the whole collection C, and $P(t_i|d)$ is the distribution probability of the term t_i in regard to document d.

In addition, a variety of term weighting models are based on the DFR model such as: Bo1 (Bose-Einstein 1), Bo2 (Bose-Einstein 2) and KL (Kullback-Leibler):

1. Bo1 and Bo2 Models are based on the Bose Einstein Statistic, and the weight of term t in the top ranked documents is calculated as shows in Equation 4.15 [Macdonald, B. He, Plachouras, et al. 2005].

$$w(t) = tf_x \cdot log_2 \frac{(1+P_n)}{P_n} + log_2(1+P_n) \tag{4.15}$$

   where tf_x is the frequency of the query term t in the top ranked documents, and P_n represents the probability of the term t in the collection.

2. KL divergence calculates the divergence between the probability distribu-

tions of terms in the whole collection and in the top ranked retrieved documents. For the term t this divergence is [Cover and Thomas 1991]:

$$KLD_{PR,PC}(t) = P_R(t) \cdot log\frac{P_R(t)}{P_C(t)} \tag{4.16}$$

where $P_R(t)$ is the probability of term t in the top retrieved documents, and $P_C(t)$ is the probability of term t in the total collection.

The previously presented IR models are based on the assumption that a document is only relevant if it looks like the query, by measuring the similarity between a document d and a query q, or by estimating the probability of a document to respond to the query. But the **Language model** [Ponte and Croft 1998] is based on calculating the capacity of each document of the collection to generate the query. The probability of generating the query $P(q/d)$ is calculated as below, with $q = t_1t_2...t_n$:

$$P(q/d) = P(t_1t_2t_3...t_n/d) = \prod_{t \epsilon q} P(t/d) \tag{4.17}$$

The **Sequential Dependence Model** (SDM) [Metzler and Croft 2005] is based on the Language Model concept, and relies on the idea of integrating multi word phrases by considering a combination of query terms with proximity constraints such as: single term features (standard unigram language model features, f_T), exact phrase features (words appearing in sequence, f_O) and unordered window features (require words to be close together, but not necessarily in an exact sequence order, f_U). And the document D is ranked, for a query Q, according to the below scoring equation:

$$SDM(Q,D) = \lambda_T \sum_{q \epsilon Q} f_T(q,D) + \lambda_O \sum_{i=1}^{|Q|-1} f_O(q_i, q_i+1, D) + \lambda_U \sum_{i=1}^{|Q|-1} f_U(q_i, q_i+1, D) \tag{4.18}$$

where λ_T, λ_O and λ_U are the feature's weights.

4.1.2. Query reformulation

The effectiveness of an IR search system is related to the accuracy in which a query is formed by, to reflect the actual user's need of information. Therefore, several techniques of query reformulation have been proposed to improve the performance of IR systems, and we present in this section the automatic query expansion and the query expansion by relevance feedback.

An **automatic query expansion** is accomplished by adding terms to the original. It can be applied using different techniques, we mention some:

- Expanding the query with terms which are related to its initial terms, for example their synonyms or definitions. These added terms are usually delivered by a linguistics resource, as the online lexical database *Wordnet* [Miller 1995]. To note that according to [Voorhees 1994], using the *Wordnet*'s synonyms for query expansion without a word sense disambiguation, can lead to a decrease in the IR performance in case the initial query reflected correctly the information's needs.

- Implying an analysis on the content of the full collection to generate correlations between pairs of terms by exploiting term co-occurrence or term clustering [Bast, Majumdar, and Weber 2007].

- Taking advantage of the query context by typically make use of top-ranked documents only, as a relevance feedback technique. For example, applying a text summarization over the top-ranked documents [Lam-Adesina and G. J. Jones 2001]. It can be more effective than the analysis of the full collection's technique that might be based on terms that are frequent in the collection but irrelevant for the query.

The **relevance feedback** approach corresponds to a query expansion technique. We can distinguish several approaches of relevance feedback like the interactive technique, where a user controls over the query modification, for example by using its responds to simple questions for an original query modification [Kumaran and Allan 2006], in addition to the pseudo relevance feedback approach (PRF), or blind relevance feedback, where no user intervention is required.

Among all query expansion approaches, PRF has been considered the most effective [Rocchio 1971][J. Xu and Croft 2017]. It offers an automatic process of query expansion, with the purpose of capturing the user's search intent, beyond the initial query content. Typical PRF methods presume the top N retrieved documents are relevant, therefore, they are used in the new query formulation. The subsequent process of term extraction from top retrieved documents relies, mostly, on the following features: terms frequency (tf) weighting with the most frequent terms selected, term frequency-inverse document frequency (tf-idf) weighting with the selection of most important words to a document in a collection [Ye and J. X. Huang 2014], and terms co-occurrence with query terms [Ye and J. X. Huang 2014].

4.1.3. Ranking aggregation

Rank aggregation is an approach for combining different rank orderings of the same set of candidates for the purpose of obtaining a better ordering. Previous

researches concluded an improvement in book retrieving performance when aggregating multiple ranks of retrieved books [Bartell, Cottrell, and Belew 1994] [Nicholas J. Belkin, Kantor, Fox, et al. 1995][Benkoussas, Ollagnier, and Bellot 2015]. In this manuscript, two methods of rank aggregation are presented: The classic Borda Count and Ordered Weighted Averaging.

Borda count is a voting system suggested by Jean-Charles de Borda around 1781 [Borda 1995]. With Borda's method, voters rank the list of candidates in order of preference. Then, on a particular ballot, the lowest ranking candidate is given 1 point, the second lowest is given 2 points, and so on, the top candidate receiving points equal to the number of candidates. The number of points given to each candidate is summed across all ballots.
In this thesis, Borda's method is employed to combine multiple book rankings. And by applying the Borda's method, the new relevance score of each book is computed by accumulating the numbers of books it exceeds in each ranking list, plus one. An example of the Borda count method is presented in Table 4.30, where the Book_A would have the score 5, equals to the number of books it exceeds in list X, plus one, which is 3, plus the number of books it exceeds in list Y, plus one, which is 2. And if a book does not exist in a list, its score in that list is considered zero (e.g. Book_D in list X). Borda's aggregation scores are presented in the two columns on the right.

Table 4.30.: Example of ranking lists aggregation using Borda's method.

Position	list X	list Y
1	Book_A	Book_B
2	Book_B	Book_D
3	Book_C	Book_A
4	-	Book_C

New Order	Score
Book_B	6
Book_A	5
Book_D	3
Book_C	2

Ordered Weighted Averaging (OWA) was introduced by [Yager 1988] as a new aggregation technique. An OWA operator of dimension n is a mapping $f : R^n \rightarrow R$, that has an associated weighting n vector :

$$w = (w1, w2, ..., wn)^T \tag{4.19}$$

such as $w_i \in [0, 1]$, 1 ≤ i ≤ n, and :

$$\sum_{i=1}^{n} w_i = w_1 + ... + w_n = 1 \tag{4.20}$$

And where :

$$f(a_1, a_2, ..., a_n) = \sum_{j=1}^{n} w_j b_j = w_1 b_1 + ... + w_n b_n \tag{4.21}$$

with b_j is the j-th largest element of the collection $a_1, a_2, ..., a_n$.

OWA operators have different behaviors bases on their associated weighting vector. In this thesis, we used the $dispersion$ measure defined as :

$$dispersion(W) = -\sum_{i=1}^{n} w_i \, ln(w_i) \tag{4.22}$$

where w_i is calculated as following, based on linguistic quantifiers [Yager 1988]:

$$w_i = Q(\frac{i}{n}) - Q(\frac{i-1}{n}), \quad i = 1, 2, ..., n \tag{4.23}$$

where n is the number of combined operands, and Q is a linguistic quantifier (regular increasing monotone). The following Q function is used as suggested by [Zadeh 1983] :

$$\begin{cases} 0 & if\ r < a \\ \frac{r-a}{b-a} & if\ a \leq r \leq b \\ 1 & if\ r > b \end{cases} \tag{4.24}$$

with a, b, $r \in$ [0,1].

4.2. Information Filtering as Recommendation systems

Definition: Information Filtering (IF) concerns the delivering of information presumably interesting or useful to the user. An IF system benefits users by filtering the data source to provide relevant information to the users. It is considered a recommending system once the provided information takes the form of suggestions. Also, a recommending system may be considered as an IR system but without a query.

In the absence of a query describing the users' needs, and with the difference in interests between users, the IF procedure requires gathering data about the user, in addition to feedback from the user, for the purpose of making a user profile of his preferences. The Figure 4.10, representing a glimpse of the IF procedure, shows a high similarity with the Figure 4.9. The users' need of information (the query) of the IR procedure is substituted with the profile of the

user, built by the collected data about its interests. The matching process is then applied as a comparison between the profile and the document representations. The documents of most similar characteristics are filtered and considered relevant, then recommended to the user. To match documents and profiles, similar models were employed in the IF systems as the ones applied in the IR systems, presented in Section 4.1.1. Note that the IF procedure is frequently applied as a consecutive step to IR, for the purpose of improving the retrieval quality. For example, the re-ranking procedure following documents' search retrieval, is actually a filtering of the retrieved set of documents, and it requires in general the practice of IF approaches like content-based and collaborative filtering, or graph-based approaches [Benkoussas, Ollagnier, and Bellot 2015].

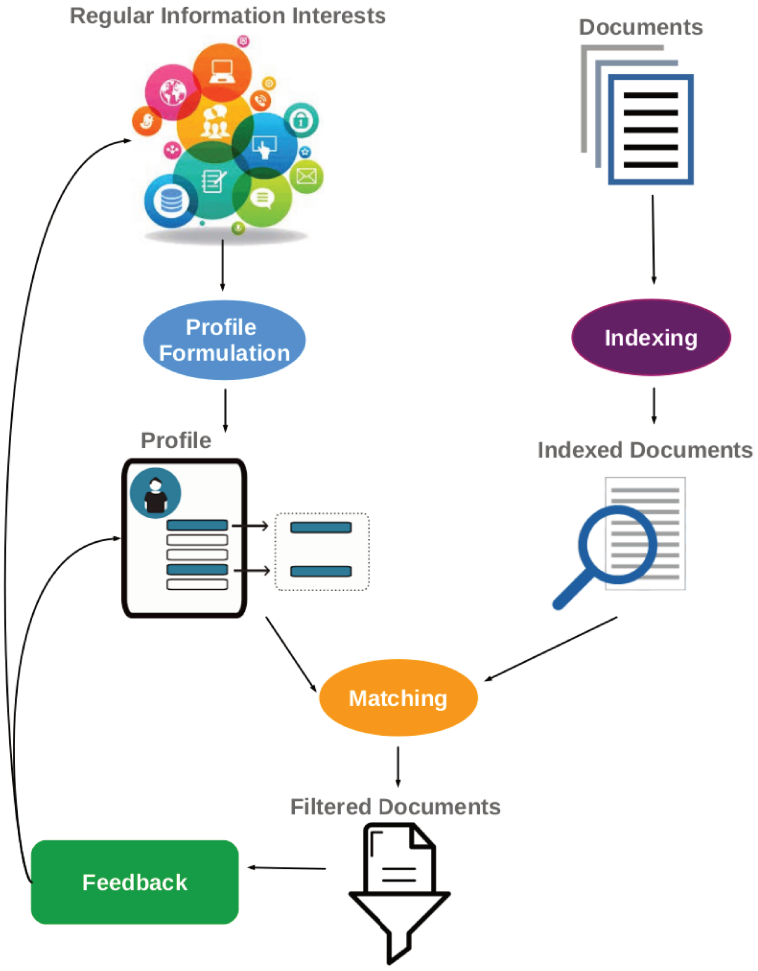


Figure 4.10.: Visualisation of the information filtering procedure.

In Chapter 5, we present our suggested methods to detect bibliographical zone in documents, which is the first step of a continuous work concerning a graph-based recommendation system, as it leads to an inter-citation relation between

the documents. For that purpose, some information filtering concepts are mentioned or used, and therefore are briefly introduced in the following sections.

4.2.1. Information Filtering approaches

The main approaches for IF are content-based filtering [Lops, De Gemmis, and Semeraro 2011] and collaborative filtering [Koren and Bell 2015][Balabanović and Shoham 1997]. The basic idea of content-based filtering is that users could be interested in items that are similar to previously liked items by these users. In other words, content-based filtering recommends items based on the correlation between the content of the items' documents, represented by a set of terms that occurs in the documents, and a user profile, represented by the terms appearing in the content of items' documents which have been seen by the user (or liked, bought, etc). Such approach is very similar to the IR procedure by giving the extracted terms of viewed items the role of a search query. And the main idea of collaborative filtering is that users like items that their analogue users liked. That is, collaborative filtering is based on filtering information by using other people's recommendations and ratings.

4.2.2. Graphs-based recommendation

The collaborative filtering leaded to several graph-based approaches, since links between users or items can minimize the sparsity of information [Gu, Zhou, and Ding 2010]. The users graph is usually based on the neighborhood information of users in the user-item rating matrix, considering that if two users have identical ratings on mutual items, then they likely to have identical ratings on other items, therefore they are considered linked neighbor-nodes in the graph of users [Jin, Chai, and Si 2004], and a user is recommended the items high rated by his neighboring users. Other information can also be used in building the graph, like the user's demographic information and the social interactions or relationship between users.

The items graph can also be based on the neighborhood information of items in the user-item rating matrix [Fei Wang, Ma, L. Yang, et al. 2006], considering that if two items have identical ratings by mutual users, then they likely to have identical ratings by the other users, therefore they are considered linked neighbor-nodes in the graph of items, and a user is recommended the items neighboring the item he high rated. The genre information of items can also be employed in connecting items in the graph, for example books of same category (e.g. Science Fiction, Romance). Other types of information can also be part of the items graph creation, depending on the items characteristics. For example, articles and books, holding bibliographical references within, can have an inter-linkage based on these citations, and we can cite the work of [Ollagnier,

Sébastien Fournier, and Bellot 2018], where they suggested the creation of a graph-based recommender with a connection type of "citing-cited" between the scientific papers.

4.3. Evaluation measures

The evaluation of IR and IF systems is essential to estimate their performance to extract relevant documents, and also to be able to compare between systems of different methods and models. The performance measures can be calculated as below, with some measures from the first chapter of this manuscript re-explained for the IR and IF fields:

- The *Precision* is used to test the system capacity to eliminate irrelevant documents, and it is equal to the ratio of the number of retrieved relevant documents to the total number of retrieved documents.
- The *Recall* is used to test the system capacity to retrieve relevant documents, and it is equal to the ratio of the number of retrieved relevant documents to the total number of relevant documents.
- The *F-measure* combines the two previous measures in the following Equation:

$$F - Measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{4.25}$$

  And to evaluate the system's performance with a set of queries, we calculate the mean of the average precision scores for each query, called *MAP* (Mean Average Precision).
- The $NDCG@k(q)$ (Normalized Discounted Cumulative Gain) is used to measure the ranking quality of the first k retrieved documents of the query q. It is the normalised score, in the interval [0,1], of DCG which is calculated in Equation 4.26:

$$DCG@k(q) = \sum_{i=1}^{k} \frac{rel_i}{log_2(i+1)} \tag{4.26}$$

  with rel_i is the relevance level of the document based on its position i. And the $NDCG@k$ of a set of queries is equals to the average $NDCG@k(q)$ of each query.

4.4. Conclusion

In this chapter we presented the notions used throughout the second part of this manuscript, including the concepts of IR and IF. Then, we presented their main

approaches and finally, we presented the employed evaluation measures.

It is important to note that the two fields of IR and IF are often considered two sides of the same coin [Nicholas J Belkin and Croft 1992]. They interfere in many tasks, for example when the filtering is used to re-rank the retrieved documents by an IR system, using users' profile. Also, once the user profile takes the shape of a set of terms, the recommendation (IF) takes the form of a search procedure (IR) using the user profile as a search query.

In this thesis, we pursue IR in books context, and we consider the **book search** field as a subsection of IR, where users would be searching for a specific book, or seeking books' suggestions, by a query of natural language text form. Also, we interpret briefly the **book recommendation**, as a subsection of IF, where books are suggested to users without a query but with a graph-based recommender of inter connectivity between books.

In the following chapters, we present our contributions over the second part of this manuscript, we start with our suggested method of bibliographical zone detection in articles and book, for the purpose of providing a linking source between the books, to be then used in a future work related to IF field (book recommendation). Then, in the next chapter we present our proposed methods to benefit from sentiment analysis in the IR field (book search), with a sentiment oriented pseudo relevance feedback method in book retrieval, and a study of sentiment analysis role in sentence classification of long and multi-topic book-search queries.

Chapter 5: Automatic Detection of Bibliographical Zone for Inter Citation Linkage

Summary

5.1. Introduction

The ability of linking documents is essential in recommendation approaches, and bibliographic references can provide a major link source. The purpose of linking documents is to construct a graph, each node in the graph would represent a document and neighboring nodes would be its citing documents or cited by documents, which reflect a certain similarity.

The work presented in this chapter is part of a research project concerning the bibliographic references in Digital Humanities (DH) data, which included: an automatic annotation of bibliographic references [Y.-M. Kim, Bellot, Tavernier, et al. 2012a][Ollagnier, Sébastien Fournier, and Bellot 2016], an identification of bibliographical zones [Htait, Sébastien Fournier, and Bellot 2016a], and an extraction of the bibliographical references to establish the links between contents and create a graph referred to as "Directed Graph of Citations" [Ollagnier 2017].

To identify documents' parts that contain references, tested on papers and articles of XML/TEI format, as a first approach we used a finite-state automaton that can detect patterns of consecutive references and annotate them as the article's bibliography, and it is performed by Unitex 3.0[45]. On the testing level, we are not capable of detecting long patterns such as bibliographical references' zones using Unitex 3.0, due to technical limitation of Unitex. Therefore, we suggest the use of machine learning technique for the annotation of references, so we can treat each reference apart.

We present our contribution in two sub-tasks:

- First Sub-Task: Retrieving references using Support Vector Machines (SVM), due to a model initially created to differentiate between the footnotes containing or not containing bibliographical references.
- Second Sub-Task: Detecting references' zone of the document, if it exists, as the largest list of consecutive references detected on the first sub-task.

5.2. Proposed method

BILBO [Y.-M. Kim, Bellot, Tavernier, et al. 2012a] is an open source software for automatic annotation of bibliographic reference. It labels the words according to their type (author, title, date, etc) as the example in Figure 5.11. Written in Python programming language, it is principally based on Conditional Random

[45] http://www-igm.uniov-mlv.fr/ unitex/

Fields (CRF), machine learning technique to segment and label sequence data. As external software, Wapiti[46] is used for CRF learning and inference and SVM-light[47] is used for sequence classification.

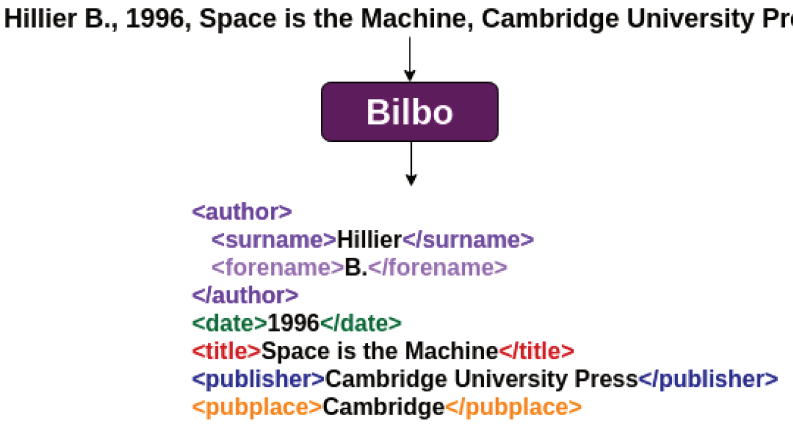


Figure 5.11.: Example of reference annotation using BILBO.

BILBO's automatic annotation includes the bibliographical references in bibliographical zones, in footnotes and in text. To annotate bibliographical references in footnotes, we should first identify bibliographical parts, because the footnotes include both bibliographical and non-bibliographical information. SVM is chosen for the classification between bibliographical and non-bibliographical information.

To build BILBO's SVM annotated corpora, Journals catalogs' articles references were used, in Figure 5.12 an example of these references [Y.-M. Kim, Bellot, Tavernier, et al. 2012b]. That corpora contained 1147 annotated bibliographical footnotes references and 385 non-bibliographical footnotes that do not contain any reference.

In the work of [Y.-M. Kim, Bellot, Tavernier, et al. 2012b], and for testing purposes, 1532 footnote instances were randomly divided into learning and test sets (70% and 30% respectively). It was tested for more than 20 different feature selection strategies. The best results, in Table 5.31, were achieved with the combination of the features, input words, punctuation marks and four different local features (posspage indicating page expressions such as 'p.', weblink, posseditor indicating editor expressions such as 'Ed.', and italic).

We should note that positive precision (Precision_positive) and positive recall (Recall_positive) measure the performance of the system to annotate correctly footnotes which contain references. And that negative precision (Precision_neg)

[46] https://wapiti.limsi.fr/
[47] http://svmlight.joachims.org/

Figure 5.12.: Example of Footnotes from Revues.org papers as references and texts.

Table 5.31.: Previous results for identifying references in Footnotes [Y.-M. Kim, Bellot, Tavernier, et al. 2012b].

Accuracy	Precision_pos	Recall_pos	Precision_neg	Recall_neg
94.78%	95.77%	97.42%	91.43%	86.49%

and negative recall (Recall_neg) measure the performance of the system to annotate correctly footnotes which do not contain any references.

BILBO SVM model was basically oriented to work with footnotes, applying the knowledge gained on texts anywhere in the body of the article will be considered as Transfer Learning [Pan, Q. Yang, et al. 2010] technique. Although the high performance of BILBO in the bibliographical footnote field annotation, the transfer learning technique might decrease its performance. Therefore, additional procedures are applied for the purpose of increasing BILBO's performance in the current task of identifying bibliographical zone, and we divide our work into two sub-tasks.

For the first sub-task, we propose a strategy of three steps, as in Figure 5.13:

- The first step, we apply a possible filtering on paragraphs. We consider the length of a reference between 20 and 500 characters, based on an observation of 100 bibliographical references.
- The second step, we use BILBO SVM model to identify references.
- The third step, since our target is to detect bibliographical references' zone which is a list of consecutive references, we consider a non-bibliographical paragraph preceded and followed by bibliographical paragraph is most probably a bibliographical paragraph. And the opposite is also true. For example, if we consider the consecutive paragraphs: parag_1, parag_2 and parag_3. If parag_1 and parag_3 are bibliographical references, we have a

good chance that the three paragraphs are part of the bibliographical zone, therefore parag_2 is considered a bibliographical reference.

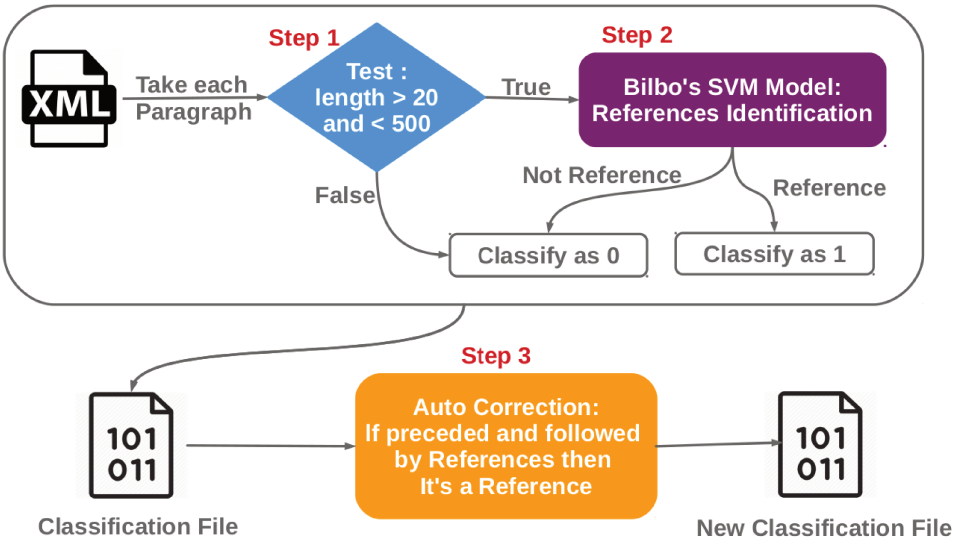


Figure 5.13.: Subtask 1: The steps to find references in text.

For the second sub-task, we search for the largest list of consecutive references. Figure 5.14 explains the algorithm used to detect the bibliographical references' zone. The file is treated by paragraphs, where each paragraph is classified as reference or not reference by BILBO's SVM model. Then the list of classified paragraphs is analysed: the first reference found is marked as the start of the zone, and with every new reference found we increment the size of the zone and mark it as the end of the zone. But once a non-bibliographical reference is found, in case of first appearance we ignore it and consider it an error by the SVM model, but in case of second appearance, we reset our zone's variables (start, end and size) to zero, in the purpose of triggering a new search for another larger references' zone. And at the end of the list, we return the positions of the largest bibliographical references' zone found.

5.3. Evaluation

5.3.1. Testing of reference identification

Our system works with semi-structured documents, since we only need to distinguish the paragraphs in the paper, but we served of the available corpora based on papers provided as structured files XML/Text Encoding Initiative[48] (TEI) by

[48] www.tei-c.org/

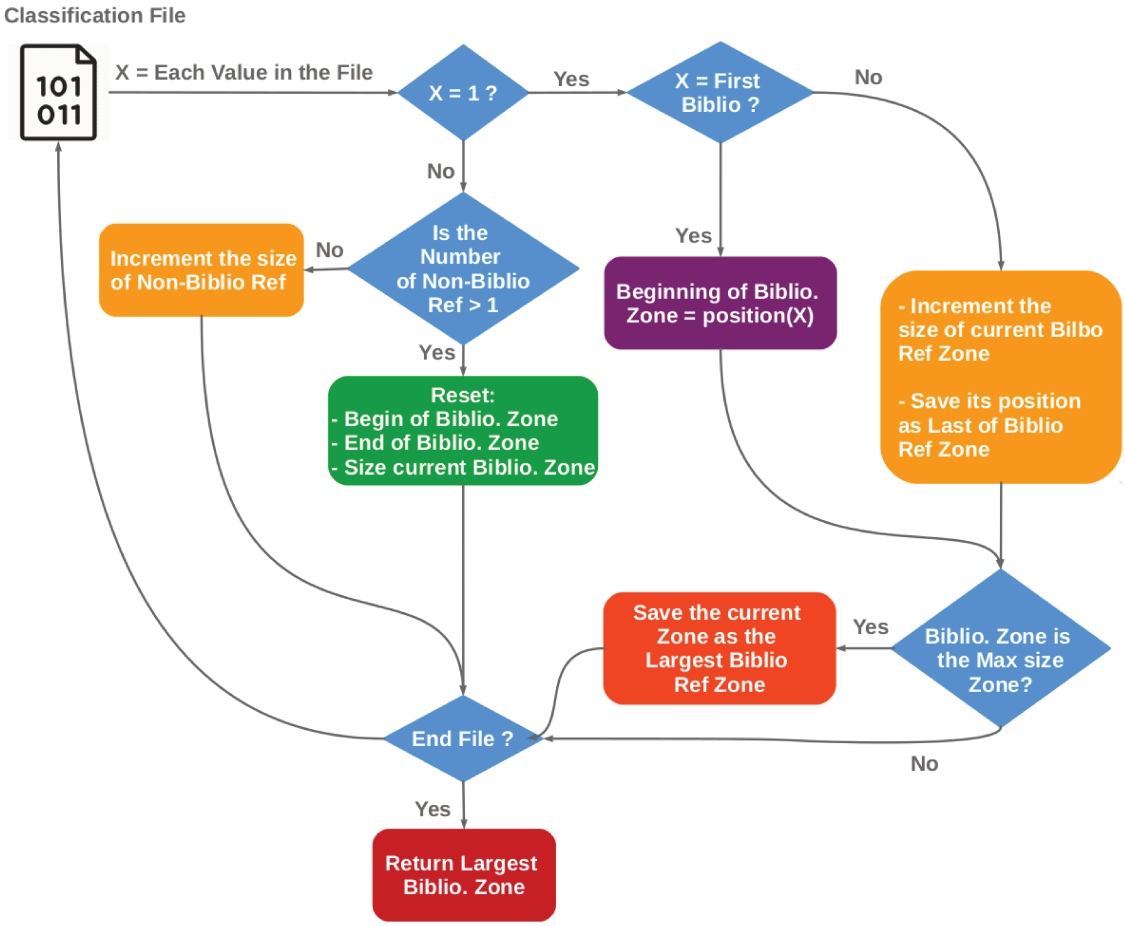


Figure 5.14.: Subtask 2: Algorithm to detect the bibliographical references' zone.

the OpenEdition's Journals catalog platform [49].

For testing purposes, we built an annotated artificial document of 1411 paragraphs, of which 275 are bibliographical references and 1136 are not bibliographical references, extracted from 10 papers of the OpenEdition's Journals catalog platform (5 French papers, 3 English papers, 1 Italian paper and 1 Spanish paper). An extract of the file is in Figure 5.15.

The prediction of SVM model, as shown in the first line of Table 5.32, results an $Accuracy$ equals to 0.80, $PrecisionPositive$ equals to 0.59, $RecallPositive$ equals to 0.50, $PrecisionNegative$ equals to 0.85, $RecallNegative$ equals to 0.89, $f\_measure$ positive[50] equals to 0.54 and $f\_measure$ negative equals to 0.87. By adding step 1 from Figure 5.15, the results, as shown in the second line

[49]https://www.openedition.org/catalogue-journals
[50]The $f\_measure$ used is the harmonic mean of precision and recall.

```
<p>The challenge is therefore to conserve a unique patrimony through
     history that has been present in this territory throughout the c
<p>Figure 14: Cugnano: vestiges d'une structure métallurgique. </p>
<p> <hi rendition="#italic">Figure 14 : Cugnano : remains of a metall
<p> <hi rendition="#bold">Bibliography </hi> </p>
<p><bibl> <hi rendition="#bold">Arangure, B., Bagnol, P., Dalla, L.,
               Archeologia Medievale</hi>, XXXIV, 79-113. </bibl></p>
<p><bibl> <hi rendition="#bold">Belli, M., De Luca, D. and Grassi, F.
          P. (dir.), <hi rendition="#italic">III Congresso Nazionale d
```

Figure 5.15.: Example of the testing set for reference identification.

of table 5.32, reflect an improvement of 2.76 points in the $Accuracy$ and 2.7 points in the $f\_measure$ positive. The most important improvement shown in our results is in the value of $recall\_positive$, and that can be be explained by the following: our method excludes the ambiguous non-bibliographical paragraphs from being mistaken for a bibliographical and by that we are increasing the number of the true positives (TP) in the Equation 5.27 of $recall\_positive$, where TP are examples correctly labeled as positives and false negatives (FN) refer to positive examples incorrectly labeled as negative [Davis and Goadrich 2006].

$$recall\_positive = \frac{TP}{TP + FN} \tag{5.27}$$

An example of similar mistakes is "$< p > Figure13 :< /p >$". First, during the conversion from PDF to XML and since the concept of paragraph is based on space between lines, the label of the image (here the example of "Figure 13") can be considered as a paragraph. Then, since this label contains: a word that starts with a capital letter, a number and a punctuation, this label may be detected as a part of a reference. This can be explained by the fact that scholarly papers used for learning include a lot of bibliographic references that are very short and incomplete.

And by adding step 3 from Figure 5.15, we can detect, as in the third line of table 5.32, an improvement on all the levels of measurement, since we seek for the consecutive bibliographical references, and that method serves greatly our purpose.

Using step 1 and step 3, as in fourth line of Tables 5.32 and 5.33, leads to an improvement of accuracy and $f\_measure$ positive and negative by almost 1 point, but a decrease in $Precision$ positive by 7 points, explained by the increase in classifying paragraphs as references while mistakenly classifying some non references paragraphs as references. Although this decrease, we decided to use both methods due to their positive effect on $Accuracy$ and $f\_measure$.

Table 5.32.: Results of references' detection steps to annotate correctly footnotes which contain references.

	Accuracy	Precision_pos	Recall_pos	f_mesure_pos
Step 2 alone	0.80	0.59	0.50	0.54
Step1 + Step 2	0.83	0.57	0.57	0.57
Step3 + Step 2	0.84	0.63	0.59	0.61
Step1 + 2 + 3	0.85	0.60	0.64	0.62

Table 5.33.: Results of references' detection steps to annotate correctly footnotes which does not contain references.

	Precision_neg	Recall_neg	f_mesure_neg
Step 2 alone	0.85	0.89	0.87
Step1 + Step 2	0.89	0.89	0.89
Step3 + Step 2	0.89	0.90	0.90
Step1 + 2 + 3	0.91	0.90	0.91

5.3.2. Testing of reference's zone identification

For testing both sub-tasks, the detection of references and the detection of references' zone, we use 20 papers in XML/TEI format from the journals of OpenEdition.org. An extract of the expected result file is in Figure 5.16, with an annotation of the references by the tag < $bibl$ >, and of the references' zone by the tags < $firstBibl$ > to show the beginning of the zone, and < $lastBibl$ > to show the end of the zone.

The below numbers show the results of our test, grouped by the level of correct bibliographical zone detection:

- 2 articles with a correct detection of the bibliographical zone, where the beginning and the end of the bibliography in the articles were marked correctly.
- 17 articles with a partially correct detection, where we have a detection of a major part of the bibliography, but not the complete zone is detected. An example is in Figure 5.17, the annotation skipped the first reference since our SVM model considered it not a bibliographical reference paragraph.
- 1 article with wrong detection of bibliographical zone. An isolated reference in the middle of the article was annotated as bibliographical zone, as shown in Figure 5.18. That's a result of not detecting any other reference in the bibliography of the article by the SVM model.

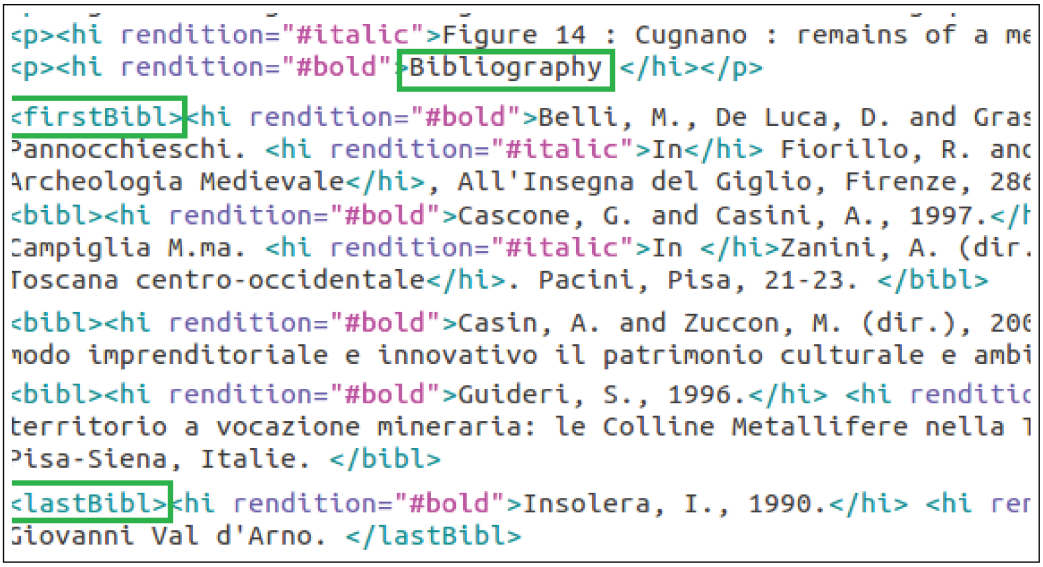


Figure 5.16.: An example of a result file after bibliographical zone detection.

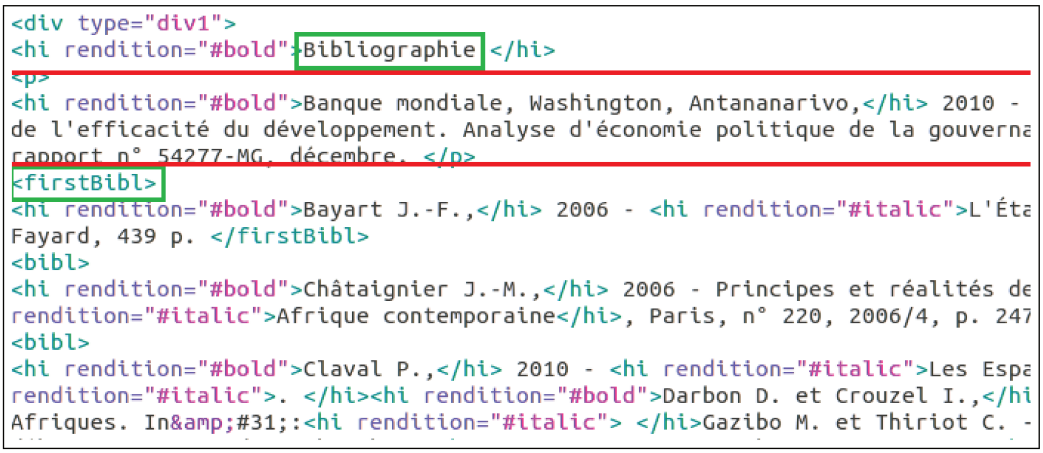


Figure 5.17.: An example of a partially correct zone detection, in the bibliographical zone detection.

In Table 5.34, based on the previous results, we are able to calculate the percentage of success in the detection of references' zone, Equation 5.28. For example, in the second line of the Table 5.34, paper_2 have a bibliographical zone formed of 8 references, 7 are detected as references' zone and 1 is not considered in the zone. That would result a percentage of success equals to 87.5%. As an average for the set of 20 papers tested, we achieved 72.23%.

$$Percentage\_of\_Success = \frac{Nb\_of\_Detected\_References}{Nb\_of\_Total\_References} \tag{5.28}$$

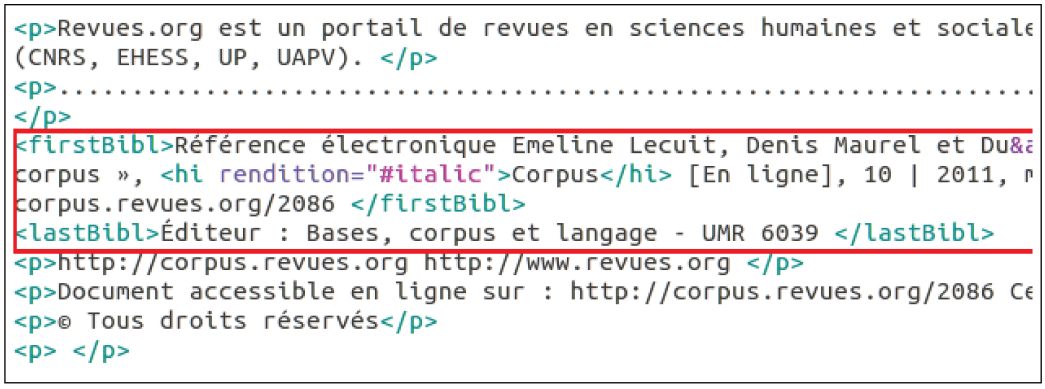


Figure 5.18.: An example of a wrong zone detection.

We notice that with 15 out of 20 papers we achieve a percentage of success higher than 70%, and for the rest of the papers the SVM had some limitation with the detection of references.

5.4. Conclusion

In this chapter, we presented our contribution to detect bibliographical zones in documents, for the purpose of creating references connections between these documents to improve their recommendation. We served first of an SVM model, BILBO, created to differentiate between bibliographical references and non bibliographical references in footnotes, to identify bibliographical references in the text of the papers body. Then, to improve the system performance, we took into consideration that the bibliographical references in papers have an average number of characters that we can be limited into an interval of maximum and minimum. In addition, we considered that bibliographical zones contain consecutive references, and therefore any non-bibliographical reference detected while surrounded by bibliographical reference is considered a bibliographical reference. We were able to achieved a $f\_measure$ equals to 0.62 in bibliographical references detection. Then, as a second step, we searched for the largest list of bibliographical references, and with a test of 20 papers, we achieve an average for the percentage of success equals to 72.23%.

In the next chapter, we proceed in exploring the other aspect of information filtering (or Recommendation), which is the information retrieval (or Search). We present our suggested methods of sentiment analysis employment in the information retrieval field, by a sentiment oriented pseudo relevance feedback method, and an analysis of the correlation between sentiment analysis and sentence's topic in a multi-topic book-search query.

Table 5.34.: Results for the percentage of success on a set of 20 Articles.

	Total_Ref	Skipped_Ref	Detected_Ref	Percentage_of_Success
Paper_1	16	0	16	100%
Paper_2	8	1	7	87.50%
Paper_3	12	1	11	91.67%
Paper_4	56	1	55	98.21%
Paper_5	34	1	33	97.06%
Paper_6	58	1	57	98.28%
Paper_7	24	1	23	95.83%
Paper_8	19	1	18	94.74%
Paper_9	14	1	13	92.86%
Paper_10	17	2	15	88.24%
Paper_11	14	9	5	35.71%
Paper_12	41	9	32	78.05%
Paper_13	17	17	0	0.00%
Paper_14	25	18	7	28.00%
Paper_15	34	22	12	35.29%
Paper_16	74	22	52	70.27%
Paper_17	15	1	17	88.23%
Paper_18	28	20	8	28.57%
Paper_19	11	1	10	90.9%
Paper_20	62	34	28	45.16%
Average				72.23%

Chapter 6: Sentiment Analysis for Book Retrieval

Summary

6.1. Introduction

Digital libraries, online bookshops and Books' social cataloging applications have increased in popularity during recent years with a continuous growth of catalogs' content (e.g. LibraryThing [51], Amazon Books). As a consequence, the books seekers are confronted with a progressive level of difficulty to manage massive catalogs, and to find books relevant to their expectations. Information Retrieval (IR) systems help assure the fast access and the accurate retrieval of books.

In Figure 6.19, we present a simplified illustration of the book retrieval process, where a user would be looking for a book, seeking suggestions or recommendations by a request of natural language text form (user query). Then, the user query is processed by a search engine, based on a specific IR model and characteristics, to search a books' collection for most relevant books. The response to the user query would be a list of most relevant books, ranked from most to least relevant.

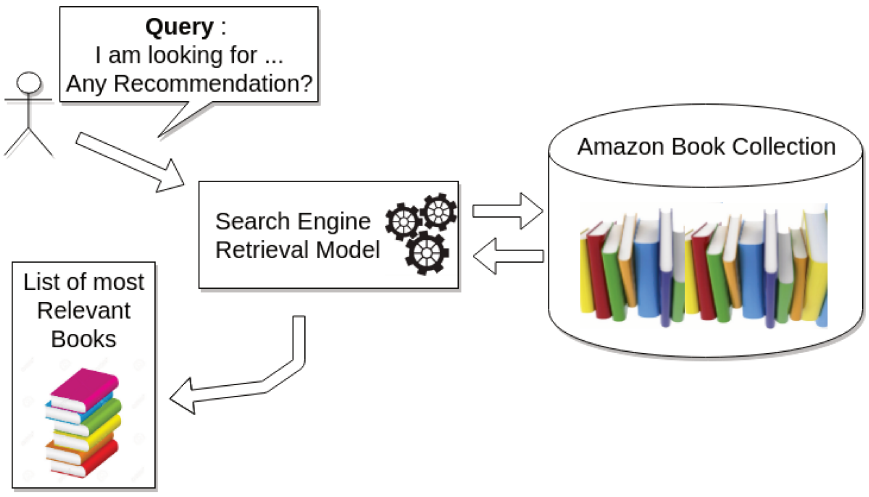


Figure 6.19.: Work Flow: Book search and recommendation.

Sentiment Analysis (SA) had been successfully applied in different fields, but it had limited contribution in the IR field. The common logic behind using SA in IR is its ability to enhance the capabilities of retrieval by allowing, for example, to find the features that customers are particularly interested in or to exclude items that have received negative evaluations. In addition, SA had even more limited participation in the book retrieval field. In this chapter, we propose new uses of SA in IR and in the book retrieval field, first, by introducing SA in a pseudo relevance feedback method for book retrieval, and second, by studying the role of sentiment analysis in sentence classification of long and multi-topic book-search queries.

[51] https://www.librarything.com/

Throughout this chapter, many experiments are done, and they employ datasets provided by Social Book Search (SBS) Lab[52] of CLEF (Conference and Labs of the Evalution Forum)[53]. SBS Lab includes the following tracks: Suggestion Track, Interactive Track and Mining Track. The Suggestion Track suggests a list of the most relevant books according to the request provided by the user, and for that task, SBS provides a collection of 2.8 million records containing professional metadata from Amazon[54](e.g. publisher, numberofpages), extended with user-generated content and social metadata (reviews, tags). An example of the collection's records is presented in Figure 6.22, as a book's metadata in an XML file.

In addition, the SBS - Suggestion Track provides realistic search requests (also called topics or queries, originally posted as questions on an online forum). A set of 208 search queries of the 2015's Suggestion Track, and a set of 120 search queries of the 2016's Suggestion Track are used in this chapter [Koolen, Bogers, Gäde, et al. 2015][Koolen, Bogers, and Kamps 2016]. And an example of a user request is presented in Figure 6.20.

```
<topic>
  <topicid>121591</topicid>
  <title>Help! i Need more books</title>
  <request>I have a Bad book a day habit and i need new
      books i love books in series (i get more books that
       way) any recommendations?</request>
  <group>Vampire Fiction</group>
  <examples/>
  <catalogue/>
</topic>
```

Figure 6.20.: An example topic in Social Book Search - 2016.

Also, a users' profiles collection is provided by SBS containing the cataloging transactions of 113,490 users. The cataloging transactions of a user is a list of information concerning the books read by the user. Each transaction is represented by a row, and each row contains eight columns; user, book, author, book title, publication year, month in which the user added that book, rating and a set of tags assigned by this user to this book. From the users profiles, we were able to create for each book an XML file with all its information. An example is illustrated in the following XML code of Figure 6.21.

[52]http://social-book-search.humanities.uva.nl
[53]http://www.clef-initiative.eu/
[54]https://www.amazon.com/

```
<book>
   <bookId>99</bookId>
   <author>A. C. Weisbecker</author>
   <title>Cosmic Banditos</title>
   <publicationYear>1988</publicationYear>
   <users>
      <user>
         <userId>u1936734</userId>
         <catalogueDate>2009-06</catalogueDate>
         <rating>0.0</rating>
         <tags>Literature, American Literature</tags>
      </user>
      <user>
         <userId>u0871476</userId>
         <catalogueDate>2008-12</catalogueDate>
         <rating>0.0</rating>
         <tags>Fiction, Humor</tags>
      </user>
   </users>
</book>
```

Figure 6.21.: An example of book XML files from users profiles collection.

For evaluation purposes, the SBS - Suggestion Track also shares a $qrels$ file. $Qrels$ are the recommendations to the search requests, retrieved from users' answers in the LibraryThing forum.

In this chapter, we examine two propositions for SA employment in book search, allocated in two main sections:

- Introducing Sentiment analysis in pseudo relevance feedback for book search.
- Sentiment analysis role in sentence's classification by topic in long book-search queries.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><!--
      version 1.0 / 2009-11-07T06:36:19+01:00 -->
<book>
    <isbn>1886330999</isbn>
    <title>Women & Anxiety: A Step-by-Step Program for
        Managing Anxiety and Depression</title>
    <ean>9781886330993</ean>
    <listprice>$14.95</listprice>
    <manufacturer>Hatherleigh Press</manufacturer>
    <publisher>Hatherleigh Press</publisher>
    <publicationdate>1998-06-17</publicationdate>
    <numberofpages>304</numberofpages>
    <dimensions>
        <height>78</height>
        <width>601</width>
        <length>901</length>
        <weight>91</weight>
    </dimensions>
    <reviews>
        <review>
            <date>1998-08-13</date>
            <summary>Stand's On It's Own </summary>
            <content>
                If you have anxiety or panic disorder, this
                    book is definately worthwhile.  Written
                    specifically for women, it provides step by
                     step guides on how to  get better and back
                     to your life.  It stands on it's own.
                    Highly  recommended.
            </content>
            <rating>5</rating>
            <totalvotes>10</totalvotes>
            <helpfulvotes>8</helpfulvotes>
        </review>
    <creators>
        <creator>
            <name>Helen Md Derosis</name>
            <role>Author</role>
        </creator>
    </creators>
    <blurbers/>
    <tags><tag count="2">women
    </tag><tag count="2">depression
    </tag><tag count="1">tr
    </tag>
    </tags>
</book>
```

Figure 6.22.: An example of book XML files in Amazon's Collection.

6.2. Related Work

Sentiment analysis had limited contribution in the information retrieval field, but we can find few interesting researches in this area; [I. H. Jensen 2012] worked on extracting sentiment from the documents' content, and they proposed an information retrieval model that ranks documents according to the expressions of sentiment in the document in addition to their topical relevance. And [Imhof, Badache, and Boughanem 2015] focused on the opinion about a mentioned book within the query content, and they expanded the user query with the information of the positively mentioned books in that query.

As for [Benkoussas and Bellot 2013], in their proposed book search method, they chose to measure the popularity of a book by the number its reviews instead of extracting the sentiment and the opinions about the book in those reviews. And we would like also to mention the work of [Y. Zhang, Lai, M. Zhang, et al. 2014], which was more oriented toward the recommendation field than the information retrieval field, but they used sentiment analysis for extracting products features with the users opinions from user reviews for the purpose of matching the specific product features to the user's interests.

The limited previous work covering the interference of sentiment analysis in information retrieval does not reflect all of what sentiment analysis is capable to offer for the information retrieval improvement. Therefore, we are presenting, in the following sections, new uses of sentiment analysis in information retrieval and in the book retrieval field.

6.3. Introducing Sentiment analysis in pseudo relevance feedback for book search

6.3.1. Introduction

Traditional applications of Information Retrieval (IR) to book search apply a full text indexing of books and match the user's query with the books' representation to retrieve a ranked list of books ordered on the basis of their relevance to the query. However, as in Web search, after examining the results produced by their first query, users generally engage in a process of query reformulation to the aim of making the query more adherent to their real needs, and to locate truly relevant books.

In the literature, several techniques of query reformulation have been proposed to improve the performance of IR systems; pseudo-relevance feedback (PRF), also known as blind relevance feedback, has been considered as one of the most effective techniques for improving retrieval accuracy by query expan-

sion. Typical PRF methods assume the top retrieved documents (books in our case), in the initial retrieval outcome, as relevant. As such, useful terms are extracted from their content to be employed in the query reformulation process to the purpose of retrieving new, better books. However, usual PRF algorithms [Ye and J. X. Huang 2014][[Bai, Song, Bruza, et al. 2005] make only use of the content of the top retrieved documents, disregarding any other type of information connected to the considered documents. Books social applications, such as OpenEdition Books and LibraryThing [55], offer, in addition to the catalogue of books' characteristics or partial content, the information generated by users about the books; these information are typically constituted by reviews, which include opinions/sentiments and personal descriptions about books, that can highlight certain aspects not included in the content of the books representation. Therefore, once extracted, these information can disclose certain aspects in books that can enrich the extended query with valuable information.

The difficulty of extracting information from book reviews that can be useful in query expansion is that the quantity of reviews associated with a book can be enormous; also, they can include a great amount of "noise". We propose an approach aimed at exploiting sentiment analysis in filtering information from large amount of reviews, followed by terms extraction to enhance query reformulation. The intuition at the basis of the proposed approach is that the writers of book reviews are often guided by the emotions provoked by the books content or characteristics, which they express in the description of the books they read. Therefore, locating emotional sentences in books reviews (e.g. $love$, $hate$), in addition to sentences with terms of strong sentiment polarity (e.g. $romance$, $crime$), can help in locating within these sentences some information useful to query expansion. The proposed approach is inspired by similar previous research exploiting sentiment analysis in filtering information. For example, [H.-L. Yang and Chao 2018] use sentiment analysis to highlight sentences with positive or negative sentiment polarity in reviews, as a selection of important information in reviews, to reduce information overloading while reading. Also, for the purpose of aspects extraction in user's comments, [Badache, Sébastien Fournier, and Chifu 2018] made use of sentiment analysis to locate aspects as nominal entities frequent and surrounded by emotional terms (e.g. $love$, $hate$). An example extracted from a book review (of Amazon Books), is the following: *[... From the first reading, my son sat still, absorbing every word.* ***The book has awesome pictures and most importantly it delivers a superb message at the end*** *...]*, where the user expresses admiration for the book's content with a strongly positive sentence (in bold), while sharing information about the book. The underlined terms, within the strongly positive sentence, would be the target to expand the initial query.

[55] https://www.librarything.com/

Based on the previous analysis, the presented new approach of query expansion combines two concepts: pseudo-relevance feedback and information filtering by sentiment analysis, where the query expansion by pseudo-relevance feedback employs the sentiment analysis as a method to filter important information from large amount of user generated content (aka reviews) associated with the first retrieved book, therefore the most relevant to the query, followed by terms and features extraction. It has to be outlined that although our proposal is not an interactive technique of relevance feedback, where a user controls the query modification [Kumaran and Allan 2006], but it offers an indirect human intervention in the query reformulation, since it identifies the expansion terms from descriptions related to the book itself by other users.

6.3.2. General overview of the proposed method: Books reviews' terms extraction

In this Section we describe the method aimed at extracting from all associated reviews of the top ranked book, which is considered the most relevant to the query, the terms to be employed in query expansion. The rationale behind the method is that the sentences that express a strong sentiment in a review contain words that characterise the book to which the review is referred. The method is organised into two main phases: the first phase is aimed at identifying, in a review, the sentences that are characterised by a strong polarity. The second phase is finalised at extracting form the sentences, previously selected, the words that will be employed in the query expansion phase.

The first phase of sentence selection is guided by sentiment analysis, and it is intended as an information filter aimed at reducing the information that can be useful for query expansion from a huge amount of reviews. The basic idea is to reduce or eliminate the less important parts of the reviews, by assuming that the sentences expressing strong sentiments, in the reviews, can be considered as carriers of useful information. The selection of sentences from the book reviews relies on several phases:

- First, each review is segmented into sentences, which are subsequently tagged by using the Stanford POS tagger [Toutanova, Klein, Manning, et al. 2003a].
- The sentiment intensity of adjectives, nouns and verbs is then calculated the tool created and shared for sentiment intensity classification: Adapted Sentiment Intensity Detector (ASID), presented in Chapter 2.
- Then, the sentences including adjectives, nouns or verbs with very high or very low sentiment intensity score (strongly positive or strongly negative),

are selected for the following phase of terms extraction.

After the phase of sentence selection, the second phase is applied for terms extraction, or what can be also described as an elimination of the overly repeated terms in the domain of book reviews, what makes them neutral for the search, as they do not add any beneficial contribution for the retrieval.

For that purpose, we have first computed the terms' frequency in the Amazon's book reviews dataset of 22M reviews [R. He and McAuley 2016]. Then, by analyzing the terms frequency, we have verified that the most frequent terms are mainly generic terms related to books (e.g. $book$, $read$, $story$, $recommand$) and general descriptive adjectives (e.g. $great$, $good$, bad). These most frequent terms are not meaningful for query expansion, and they are used as stopwords. In particular, we have generated a stoplist composed of 500 words, from the most frequent words. This list has been manually validated to exclude those terms that can be helpful in the search like: $children$, $fiction$, war, etc. The final list we have produced includes 454 empty terms (or new stopwords).

Eventually, the terms added to the initial query for the query expansion are the persisting terms from the selected sentences in the previous phase after removing all stopwords (including the previously created stoplist) and all terms starting with a capital letter (like books titles, names of people and places, etc).

We present in Figure 6.23 an example of both sentence selection and terms extraction for query expansion, starting by the following user query: *"I recently sponsored a child in El Salvador, and would like to read more about the country. I'm open to both fiction and non-fiction."*. After the first retrieval run, the top ranked book is considered as the most relevant to the query; therefore all its reviews are processed as we previously explained. Then, the sentences (in bold) with highly sentimental terms (underlined) are selected. Next, the terms selection phase is applied to extract the terms to be employed in the expansion of the initial query.

6.3.3. Experiments

The proposed method has been tested in a book search context, by considering that the book search system accepts the queries in the form of natural language text. The search is done through the meta-data and the users-generated content of books, instead of searching through the whole content of the books, as imposed by the employed collection. Then, the book search system produces a retrieved list of the most relevant books according to the query. This retrieval is required as a step preceding the query expansion by PRF, as an initial retrieval. In this section, we experiment with two proposed initial retrieval methods described in the following sections.

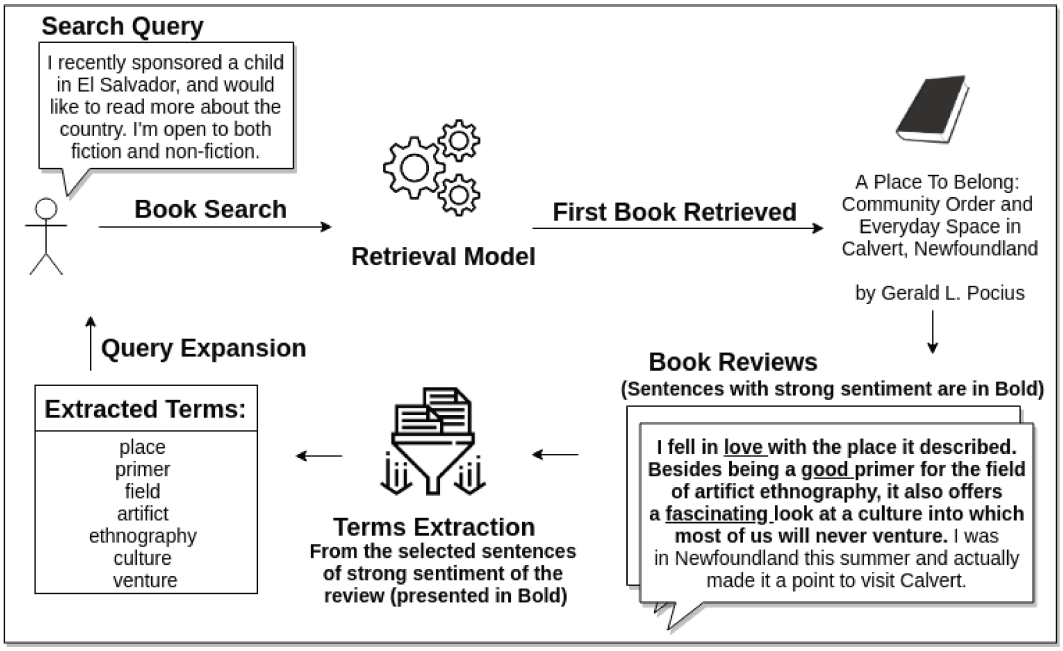


Figure 6.23.: An example of sentences selection and terms extraction from book reviews for the purpose of query expansion.

6.3.3.1. Initial Retrieval: First proposed method of SDM model and re-ranking

Our first experiment [Htait, Sébastien Fournier, and Bellot 2016c] to create an initial retrieval operation, included a query expansion method, followed by a re-ranking procedure on the retrieved books' list. For the proposed method, the Sequential Dependence Model (SDM) is used as a retrieval method. And the feature weights of the Equation 4.18 are set to $\lambda_T = 0.85$, $\lambda_O = 0.1$ and $\lambda_U = 0.05$, based on previous experiments [Bonnefoy, Deveaud, Bellot, et al. 2012].

The search queries are built by combining the title and the text of the user's query, extended by the tags extracted from the users profiles collection of example books mentioned by the user in the request. The framework *Indri*[56] is used, in this experiment, with SDM as an implemented retrieval model. *Porter stemmer* and Bayesian smoothing with *Dirichlet* priors (with $\mu = 1500$) are used in these experiments.

The retrieved results of relevant books are then re-ranked using scores calculated of the following books' information : $price$, $publication\ Date$ and $number\ Of\ Pages$ (extracted from Amazon's book collection, Figure 6.22). And since the combined values are of different weighting, the scores are normalised by applying the Equation 6.29 [J. H. Lee 1995].

[56] http://www.lemurproject.org/

$$normalized\_Score = \frac{old\_Score - min\_Score}{max\_Score - min\_Score} \tag{6.29}$$

The re-ranking is employed through a linear aggregation between the scores of SDM model and the books information. But since they have different levels of retrieval effectiveness, it is necessary to weigh scores depending on their overall performance. We use an interpolation parameter (α) that varies in testing for the goal of achieving the best interpolation that provides better retrieval effectiveness. The linear aggregation is shown in the Equation 6.30.

$$SDM\_bookInfo = \alpha.(SDM(Q, D)) + (1 - \alpha).(bookInfo(D)) \tag{6.30}$$

After several testings on SBS topics of 2015, α is set to 0.55 with the best result. Then, $bookInfo(D)$ is calculated by a normalized score of only the *book price*, since the price alone obtained the best result on SBS topics of 2015 compared to the combined values of $price$, $publication\ date$ and $number\ of\ pages$. Table 6.35 shows an example of the tests with a modest but still an increase in the results when combining books prices to the SDM, with α = 0.55. The results are presented with the measure of ranking quality $NDCG@10$[57].

Table 6.35.: Results of book information's aggregation with the SDM model, in first experiments for the initial retrieval operation, applied on SBS 2015 Topics.

Method	$nDCG@10$
SDM(Q, D)	0.1278
SDM_bookInfo_all	0.1251
SDM_bookInfo_price_0.4	0.1275
SDM_bookInfo_price_0.6	0.1267
SDM_bookInfo_price_0.55	**0.129**

Table 6.36 shows our system's official participation results in SBS Suggestion Track of 2016. The system shows a decrease in performance compared to the tests on SBS topics of 2015, such decrease can be related to the fact that our suggested method is based on the tags of the example books mentioned in the queries. Unfortunately, around 30% only of the 2016 users' queries included example books. Also, there was no tags to extract for almost half of the example books collected.

Based on these results, we propose a second method for the initial retrieval, presented in the next section, without any query expansion to avoid the none availability of data source we faced in the current proposed method.

[57]Normalised Discounted Cumulative Gain" of the first 10 elements of the list.

Table 6.36.: Our official participation results at SBS 2016. The runs are ranked according to $nDCG@10$.

Run	$nDCG@10$
SDM	0.0450
SDM + Book Price Score	0.0177

6.3.3.2. Initial retrieval: Second proposed method of multiple retrieval aggregation

The second experiment to create an initial retrieval is not based on any query modification method (e.g. expansion, reduction), but on a combination of multiple retrieval models, and Figure 6.24 illustrates this second experiment workflow.

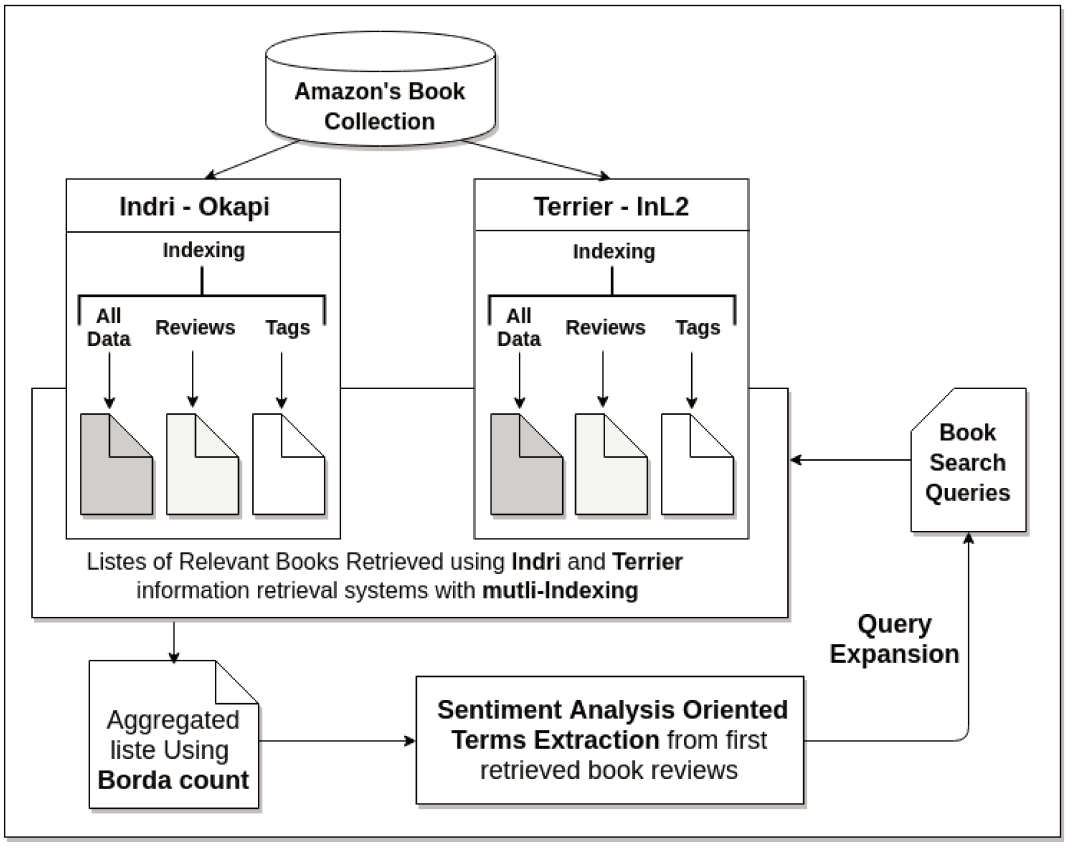


Figure 6.24.: Workflow of sentiment oriented pseudo relevance feedback's experiments, with a combination of multiple retrieval models for the initial retrieval.

We experimented with several retrieval models implemented in Indri[56] and Ter-

rier's[58] frameworks, before choosing the two with best results for our rank aggregation: the previously used SDM retrieval model (with Jelinek-Mercer smoothing), the probabilistic retrieval models tf-idf, Okapi, BM25, PL2 and InL2 of the DFR retrieval model, which is based on tf-idf measure with L2 term frequency normalization [Amati 2003].

These different retrieval models are tested seeking best retrieval performance, presented in Table 6.37, using user's queries of SBS Suggesting track of 2016, with $nDCG@10$ as a the measure of ranking quality.

Table 6.37.: Testing the performance of multiple retrieval models in $nDCG@10$, using the 120 search queries of the 2016's Suggestion Track.

Retrieval model	$nDCG@10$
Indri SDM	0.0857
Indri tf-idf	0.0952
Indri Okapi	**0.1048**
Terrier BM25	0.0786
Terrier PL2	0.0482
Terrier InL2	**0.1008**

We were able to achieve the best results by using Okapi similarity computation model by Indri, and as a second best results, by using InL2 model implemented in Terrier. Therefore, these two retrieval models have been selected for the next step of our experiments. Okapi's indexing is applied with the *krovetz stemmer*, in addition to a stop-words removal. And the retrieval is performed with the parameters *k1=3.0, b=0.3, k3=2*. InL2's indexing is applied with the default stop-words list of Terrier and the *porter stemmer*, and the retrieval is applied with parameter *c=1.1*. Note that for both search models, the set of queries is built based on the queries fields: <title> and <request>.
Furthermore, three different indexing strategies are applied with each selected search model: an indexing of all books' meta-data in the book collection, an indexing of only the reviews (Opinions of readers or editors about the book), and an indexing of only the tags (Single terms added by users describing the book). Thus, the initial retrieval operation is based on the combination of two retrieval models, InL2 and Okapi, with three different indexing strategies, presented in Table 6.38.

The OWA method is tested for the rank aggregation procedure, by adding up the position of each book, after associating a weight to each position ranking value. Also, the test was applied with the values a and b, of Equation 4.24, varying between 0.0 and 1.0, with a sequence of +0.05. The testing results,

[58] http://terrier.org/

Table 6.38.: The $nDCG$@10 of book retrieval, with the 120 search queries, for each retrieval model with each indexing strategy.

Models	Indexing	Initial Query ($nDCG$@10)
Indri - Okapi	Meta-data	0.1048
Indri - Okapi	Reviews	0.1079
Indri - Okapi	Tags	0.0707
Terrier - InL2	Meta-data	0.1008
Terrier - InL2	Reviews	0.0931
Terrier - InL2	Tags	0.0541

presented in Figure 6.25, shows a decrease in ranking quality, since the best results in that graph achieve a $nDCG$@10 equals to only 0.032, with a in the interval [0.65, 0.8] and b in the interval [0.85, 0.95].

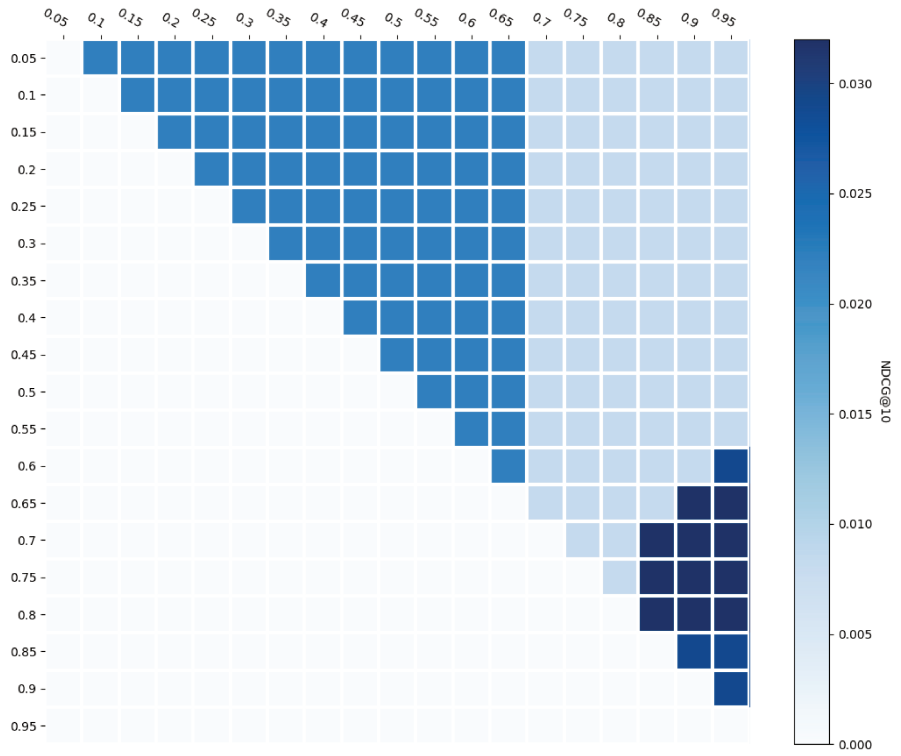


Figure 6.25.: The Ordered Weighted Averaging (OWA) method results with the values a and b, of Equation 4.24, varying between 0.0 and 1.0, with a sequence of +0.05.

Then, the Borda count method is tested for the rank aggregation process, and as shown in the last row of Table 6.39, it improved the retrieval performance by increasing the $NDCG$@10 value.

Table 6.39.: The $nDCG$@10 of book retrieval, with the 120 search queries, for each retrieval model with each indexing strategy, then combined with Borda Count method.

Models	Indexing	Initial Query ($nDCG$@10)
Indri - Okapi	Meta-data	0.1048
Indri - Okapi	Reviews	0.1079
Indri - Okapi	Tags	0.0707
Terrier - InL2	Meta-data	0.1008
Terrier - InL2	Reviews	0.0931
Terrier - InL2	Tags	0.0541
Borda Count Aggregation		**0.1205**

6.3.3.3. Final results with the suggested method of pseudo relevance feedback

Once we achieved good retrieval results by the second proposed method of multiple retrieval aggregation for the initial retrieval, we proceeded by testing the proposed method of sentiment oriented PRF.

For each search query, terms are extracted from the top N ranked (of Borda count aggregation) books' reviews, as explained in Section 6.3.2, and added to the original query. The best results are achieved with N equal to one and a maximum number of extracted terms equal to 40.

When applying our proposed method, we noticed that the query expansion can change the sentiment polarity of the query, which can redirect the search away from the user's intent. For example, with a query seeking books about "*tourism in Vietnam*", we can encounter reviews mentioning a war that happened years ago in that country. By consequence, terms like *war*, *crimes* and *death* could be added to the original query. Therefore, we decide, for the user's queries not containing any strongly negative words (e.g. war, $crime$), to extract only the highly positive sentences from the reviews. And for the user's queries containing strongly negative words, we extract the highly positive and highly negative sentences from the reviews.

After re-running the previous systems with the new extended query, the results in column $NDCG@10 + QE$ in Table 6.40 show a remarkable improvement with all the search models and indexing strategies used in this work (with (*) for p-value < 0.01). Note that our proposed method improved the results of 21% of the total number of tested queries.

Table 6.40.: The $nDCG$@10 of book retrieval, with the 120 search queries, for each retrieval model with each indexing strategy, before and after Query Expansion (with (*) for p-value <0.01).

Models & Indexing	Initial Query	Extended Query
Okapi - Meta-data	0.105	**0.141**
Okapi - Reviews	0.108	**0.134**
Okapi - Tags	0.071	**0.107**
InL2 - Meta-data	0.101	**0.144** *
InL2 - Reviews	0.093	**0.118**
InL2 - Tags	0.054	**0.085**
Borda Count Aggregation	0.121	**0.159** *

For comparison purpose, and to highlight on the importance of sentiment analysis role in information filtering for the query expansion, we apply the classic tf-idf[59] words weighting method, to extract terms based on their importance to the review in a total collection of 22M reviews from Amazon's book reviews [R. He and McAuley 2016]. Since we are comparing both methods, it is important for these methods to extract approximately the same number of terms for the query expansion. By observing the extracted terms guided by sentiment analysis, we notice that, for most queries, the number of these added terms is between 1 and 18 terms, with an average of 5 and a median of 5.6 terms by query. Therefore, in the tf-idf terms extraction method, we limit the terms extraction to 6 terms by query. The results are shown in Table 6.41 with a comparison between our sentiment based method and the tf-idf method, where the tf-idf method was not even able to improve the initial retrieval results, but the sentiment based method achieved much better results.

In a more detailed observation of the results, the sentiment based method was able to improve the retrieval performance of 27 queries, but also decreased the retrieval performance of 14 queries, out of the 120 SBS queries (the rest of the queries' results were not affected by the the query expansion). And as for the tf-idf based method, it improved the retrieval performance of 18 queries, but decreased the retrieval performance of 25 queries. Also, the sentiment based method was able to achieve higher scores in the retrieval performance. For example, the initial query presented in Figure 6.23 had a $nDCG$@10 equals to 0.000, then after expanding that query by the tf-idf method it achieved a $nDCG$@10 equals to 0.184, but it did not surpass the sentiment analysis based method improvement since it reached an $nDCG$@10 equals to 0.218.

Table 6.42 presents the official SBS Suggestion Track evaluation results of

[59] term frequency-inverse document frequency

Table 6.41.: The $nDCG$@10 of book retrieval, using the 120 search queries of the 2016's SBS Suggestion Track, for each retrieval model with each indexing strategy, by the tf-idf method and by our sentiment based method for Query Expansion (QE).

Models & Indexing	QE by Sentiment ($nDCG$@10)	QE by tf-idf ($nDCG$@10)
Okapi - Meta-data	**0.141**	0.072
Okapi - Reviews	**0.134**	0.085
Okapi - Tags	**0.107**	0.047
InL2 - Meta-data	**0.144**	0.079
InL2 - Reviews	**0.118**	0.078
InL2 - Tags	**0.085**	0.047

2016[60], with the two top teams' results, in comparison to our results achieved by applying the proposed method of PRF based on a sentiment analysis information filtering technique. We did not exceed the best result, but we were able to slightly surpass the second best team's result.

Table 6.42.: The official Suggestion Track evaluation results 2016 of the two top teams' results, via $nDCG$@10

Team	$nDCG$@10
Best score (Team PRIR)	0.215
Our results	**0.159**
Second Best score (Team CERIST)	0.156

6.3.4. Discussion

In this section a new approach of query expansion by pseudo-relevance feedback, based on sentiment-oriented terms extraction from user generated content, has been presented and tested in a book search context. The proposed method requires an initial retrieval, which is based, in this work, on a combination of multiple retrieval models' results: Indri's Okapi and Terrier's InL2, applied with three different indexing strategies (all books meta-data, books reviews and books tags). Following the initial retrieval, the sentiment intensity classification was used to filter information, by sentences selection from the reviews of the first initially retrieved book, for extracting terms to be exploited in the query formulation. The experiments showed a ranking quality improvement, of $nDCG$@10, between 25% and 56% after query expansion with our proposed approach compared to the initial query results. In addition, a comparison with the classic tf-idf terms retrieval method is applied, and it showed the effectiveness of introducing

[60] http://social-book-search.humanities.uva.nl/#/suggestion16

sentiment analysis in information filtering for PRF purposes.

Note that this approach is the subject of a future work where we plan on testing a combination of our sentiment based information filtering method with several term weighting method for the purpose of a better, and more competitive, retrieval results. In addition, such combination might allow the covering of a larger number of reviews related to the N top retrieved books, and not only the first retrieved book.

In the next section, we present another suggestion for sentiment analysis employment in book search; An analysis of the correlation between the sentiment and the information in the sentences of long book-search queries is presented, for the purpose of a sub-queries classification by topic.

6.4. Sentiment Analysis and Sentence Classification in Long Book-Search Queries

6.4.1. Introduction

Social cataloging web applications store and share book catalogs and various types of book metadata, while allowing users to search for books or seek recommendations. Its recommendation and search queries are usually destined to humans [61], what makes them often long, descriptive, and even narrative. Users may express their needs for a book, preferences in a type or genre of books, opinions toward certain books, describe content or event in a book, and even sometimes share personal information (e.g. *I am a teacher*).

Being able to differentiate the topic of sentences, in previously described long multi-topics queries, can improve in different ways the automation of such book-search tasks. Detecting unhelpful to search sentences in the query (e.g. *Thanks for any and all help.*), can help in query reduction by eliminating those sentences as they are not considered as carriers of useful information. And classifying sentences by their topic, can be used for an adapted search. For example, sentences including good read experience, with a book title, can be oriented to a book similarity search, but sentences including a certain topic preference should be focusing on a topic search. And also, sentences including personal information can be used for a personalised search.

In this section, sentence classification is studied on two levels: the helpfulness of the sentence with meaningful information for the search, and the topics or

[61] An example of a query in LibraryThing: https://www.librarything.com/topic/4920

type of information provided by the sentence. Three topics are highlighted on: book titles and author names (e.g. *I read "Peter the Great His Life and World" by Robert K. Massie.*), personal information (e.g. *I live in a very conservative area*), and narration of book content or story (e.g *The story opens on Elyse overseeing the wedding preparation of her female cousin.*).

The default in text classification is using terms as features, which is applied also in sentence classification. In this work, the possibility of introducing new features is tested. Since "Different types of sentences express sentiment in very different ways" [T. Chen, R. Xu, Y. He, et al. 2017], the correlation between sentiment in a sentence and its topic is studied to test the possibility of introducing sentiment as a feature. For this task, sentiment intensity is calculated, for its capacity to distinguish between sentences of same polarity, using the semi-supervised method in the first part of this manuscript, re-explained briefly in Section 6.4.4.

In addition, sentences in a query can share similar writing style and subjects with book reviews. Below is a part of a long book-search query:
[I just got engaged about a week and a half ago and I'm looking for recommendations on books about marriage. I've already read a couple of books on marriage that were interesting. ***Marriage A History talks about how marriage went from being all about property and obedience to being about love and how the divorce rate reflects this. The Other Woman: Twenty-one Wives, Lovers, and Others Talk Openly About Sex, Deception, Love, and Betrayal not the most positive book to read but definitely interesting. Dupont Circle A Novel I came across at Kramerbooks in DC and picked it up. The book focuses on three different couples including one gay couple and the laws issues regarding gay marriage*** *...]*
In the query example, the part in bold represent a description of specific books content with books titles, e.g. "*Marriage A History*", and interpretations or personal point of view about the book with expressions like "*not the most positive book ... but definitely interesting*". These sentences are similar to book reviews sentences. Therefore, calculating the similarity between sentences in a query and books reviews can be a possible feature for sentence classification, as it can help classifying sentences with book titles. To calculate that similarity in a general form, a reviews statistical language model is used to find, for each sentence in the query, the probability of being generated from that model (and therefore its similarity to that model's training dataset of reviews).

In the following sections, we present in more details the analysis of sentence's topic correlation with its sentiment intensity and its similarity to reviews.

6.4.2. Related Work

Many machine learning techniques have been applied, for the purpose of query classification, some as supervised [Kang and G. Kim 2003], some as unsupervised [Diemert and Vandelle 2009] and others as semi-supervised machine learning techniques [Beitzel, E. C. Jensen, Frieder, et al. 2005]. In book-search field, fewer studies covered query classification. [Ollagnier, Sébastien Fournier, and Bellot 2015] worked on a supervised machine learning method (Support Vector Machine) for classifying queries into the following classes: **oriented** (a search on a certain subject with orienting terms), **non-oriented** (a search on a theme in general), **specific** (a search for a specific book with an unknown title), and **non-comparable** (when the search do not belong to any of the previous classes). Their work was based on 300 annotated query from INEX SBS 2014[62]. The previously mentioned work covered the query classification by its type and not the classification of the sentences within the query by their topic. But the length of book-search queries creates new obstacles to defeat, and the most difficult obstacle is the variety of information in its long content, which require a classification at the sentence level.

Sentences in general, based on their topic, reveal sentiment in different ways, therefore, [T. Chen, R. Xu, Y. He, et al. 2017] focused on using classified sentences to improve sentiment analysis with deep machine learning. In this work, the possibility of a reverse perspective is studied, which is the improvement of sentence classification using sentiment analysis.

In addition, this section is studying the improvement of sentence classification using language model technique. Language models have been successfully applied to text classification. In [Bai, Nie, and Paradis 2004], models were created using training annotated datasets and then used to compute the likelihood of generating the test sentences. In this work, a new model is created based on book reviews and used to compute the likelihood of generating query sentences, as a similarity measurement between book reviews style and book-search query sentences topic.

6.4.3. Book-search queries' annotation

Out of 680 user queries, from the 2014's dataset of Social Book Search Lab, 43 queries are randomly selected based on their length, since this work focus on long queries. These 43 queries have more than 55 words, stop-words excluded. Then, each query is segmented into sentences, which results a total of 528 sentences. These sentences are manually annotated, for this study, based on their helpfulness to the search, and on the information they provide as: book titles

[62] https://inex.mmci.uni-saarland.de/data/documentcollection.html

and authors names, personal information, and narration of book content. An example is shown in Figure 6.26.

```
<sentences>
 <sentence relevant="False" info="Null" >
  Where is the time to go online and talk?<\sentence>
 <sentence relevant="True" info="General">
  No sappy romance involved<\sentence>
 <sentence relevant="True" info="Personal_Info">
  I am a sixth grade science teacher<\sentence>
 <sentence relevant="True" info="Book_Content">
  Pierre becomes for a while a Mason<\sentence>
 <sentence relevant="True" info="Book_Author">
  I have one by Robert Fitzgerald Peter<\sentence>
<\sentences>
```

Figure 6.26.: An example of annotated sentences in book-search queries for the purpose of sentence classification.

6.4.4. Sentiment Intensity prediction for sentences

As part of this work, sentiment intensity is calculated for each sentence of the queries. We use the sentiment intensity prediction method applied in the first part of this manuscript, using the tool created and shared Adapted Sentiment Intensity Detector (ASID). The semi-supervised method, applied on book reviews, predicts sentiment intensity of words. To predict the sentiment intensity of the entire sentence, first the adjectives, nouns and verbs are selected from the sentence using Stanford POS tagger [Toutanova, Klein, Manning, et al. 2003b], then the ones with very high or very low sentiment intensity are used by adding up their score to have a total score for the sentence. Figure 6.27 presents an example of sentiment intensity calculation for a query sentence. The sentiment intensity is calculated for the verbs in the sentence (since this sentence does not contain any nouns or adjectives), then the very high and very low sentiment intensity scores are selected, which is in that sentence one value only, of the verb *blaming*, equals to -0.30.

6.4.5. Reviews' language model

Since a similarity in writing style is noticed between certain sentences of user queries and the book reviews, such similarity is considered a sentence's characteristic to be detected. Therefore, to detect this similarity in style, a statistical language modeling approach is used to compute the likelihood of generating a

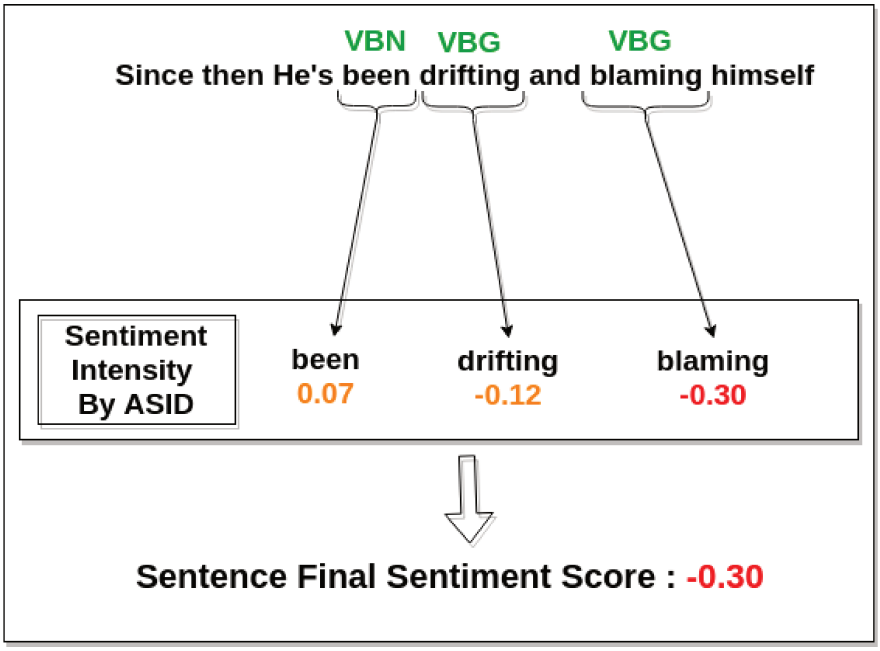


Figure 6.27.: An example of sentiment intensity calculation for query sentences using the tool ASID.

sentence of a query from a book reviews language model. Such method is unsupervised and does not require an annotated dataset.

The statistical language modeling were originally introduced by Collins in [Collins 1997], and it is the science of building models that estimate the prior probabilities of various linguistic units [Stolcke 2002]. It makes it possible to easily consider taking into account large linguistic units, like bigrams and trigrams.

The tool SRILM[63] [Stolcke 2002] is used to create the model from book reviews dataset (as training data), and to compute the probability of sentences in queries to be generated from the model (as test data). The language model is created as a standard language model of trigram and Good-Turing discounting (or Katz) for smoothing, based on 22 million of Amazon's book reviews [R. He and McAuley 2016], as training dataset.

The tool SRILM offers details in the diagnostic output like the number of words in the sentence, the sentence likelihood to model or the logarithm of likelihood by $logP(W|\theta_R)$, and the perplexity which is the inverse probability of the sentence normalized by the number of words, as shown in Equation 6.31. In this paper, the length of sentences vary from one word to almost 100 words, therefore the score of perplexity seems more reliable for a comparison between sentences. Note that minimizing perplexity is the same as maximizing probability of likelihood, and a low perplexity indicates the probability distribution is good at

[63] http://www.speech.sri.com/projects/srilm/

predicting the sample.

$$perplexity = \sqrt[m]{\frac{1}{P(W|\theta_R)}} \tag{6.31}$$

with m as the number of words in the sentence, and $P(W|\theta_R)$ is the probability of the sentence W likelihood to the model θ_R.

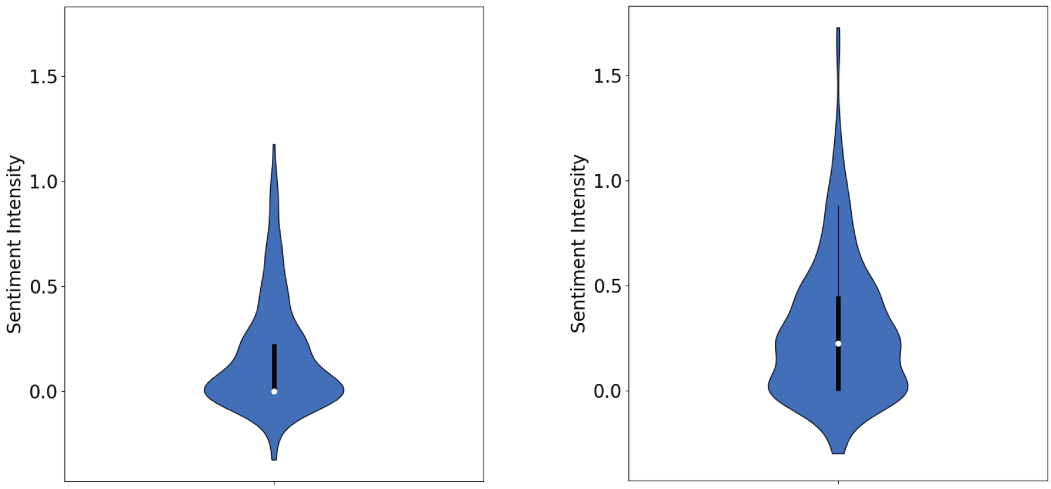
6.4.6. Displaying data in graphs

As previously explained, a corpora of 528 sentences from user queries is created and annotated as the examples in Figure 3.5. Then, for each sentence the sentiment intensity score and the perplexity score are calculated following the methods explained in the previous sections. To present the scores, Violin plots are used for their ability to show the probability density of the data at different values. Also, they include a marker (white dot) for the median of the data and a box (black rectangle) indicating the interquartile range.

6.4.6.1. Correlation between sentiment intensity, perplexity and sentences' usefulness

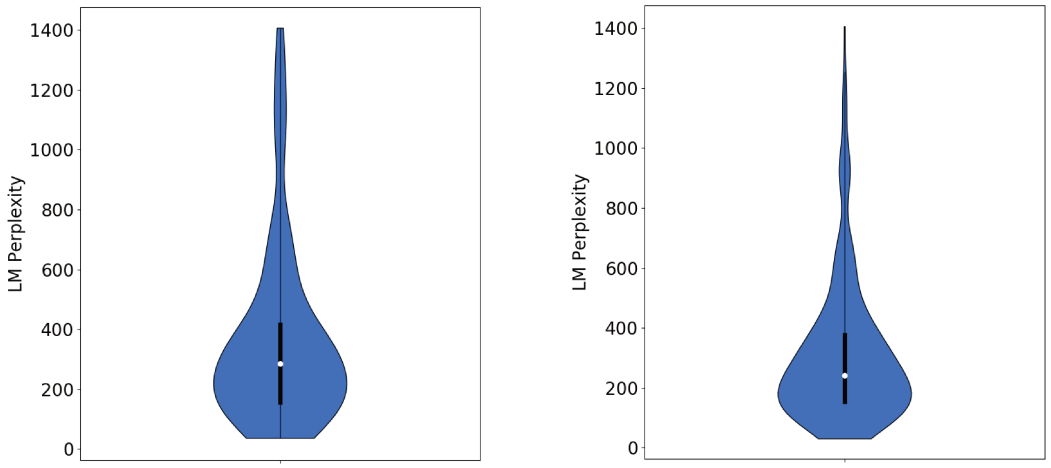
The graph in Figure 6.28 shows the distribution (or probability density) of **sentiment intensity** between two categories of sentences: on the right the sentences which are helpful to the search and on the left the sentences which are unhelpful to the search (noise). The shape on the left is horizontally stretched compared to the right one, and mostly dilated over the area of neutral sentiment intensity (sentiment score = 0), where also exist the median of the data. On the other hand, the shape on the right is vertically stretched, showing the diversity in sentiment intensity in the helpful to search sentences, but concentrated mostly in the positive area, at sentiment score higher than zero but lower than 0.5.

The graph in Figure 6.29 represent the distribution of **perplexity** between two categories of sentences: on the right the sentences which are helpful to the search and on the left the sentences which are unhelpful to the search (noise). Both shapes are vertically compressed and dilated over the area of low perplexity. The graph on the right in Figure 6.29, representing the distribution of helpful sentences, shows the median of the data distribution (the white dot) on a lower level of score of perplexity, than the left graph. Explained by the slightly horisontal dilation of the left graph above the median level.



(a) Not Useful (b) Useful

Figure 6.28.: The distribution of sentiment intensity between two categories of sentences: on the right the sentences which are helpful to the search and on the left the sentences which are unhelpful to the search.



(a) Not Useful (b) Useful

Figure 6.29.: The distribution of perplexity between two categories of sentences: on the right the sentences which are helpful to the search and on the left the sentences which are unhelpful to the search.

6.4.6.2. Correlation between sentiment intensity, perplexity and topics

The graphs in Figure 6.30 show the distribution of sentiment on sentences based on their topic. The graphs are described below consecutively, from top to bottom,

by topic:

- Book titles and authors names: on the right the sentences with books titles or authors names, and on the left the sentences without books titles and authors names. The graph on the right shows a high distribution of positive sentiment, but the left graph shows a high concentration on neutral sentiment with a small distribution for positive and negative sentiment. Also, It is noticed the lack of negative sentiment in sentences with books titles or authors names.

- Personal information: on the right the sentences containing personal information about the user, and on the left the sentences without personal information. The graph on the right shows a high concentration on neutral sentiment, where also exist the median of the data, and then a smaller distribution in positive sentiment. On the left, the graph shows a lower concentration on neutral sentiment, but it is noticeable the presence of strongly positive sentences.

- Narration of book content: on the right the sentences containing book content or events, and on the left the sentences without book content. Both graphs are vertically stretched but have different shapes. The graph on the right shows a higher distribution of negative sentiment as for sentences with book content, and the graph on the left shows higher positive values.

The graphs in Figure 6.31 shows the distribution of perplexity between the informational sentences, consecutively from top to bottom: Book titles and authors names, Personal information and Narration of book content. When comparing the first set of graphs, of book titles and authors names, the left graph has its median of data on a lower perplexity level than the right graph, with a higher concentration of data in a tighter interval of perplexity. For the second sets of graphs, of personal information, the right graph shows a lower interquartile range than the left graph. As for the third set of graphs, of book content, a slight difference can be detected between the two graphs, where the left graph is more stretched vertically.

6.4.7. Graphs interpretation

Observing the distribution of data in the graphs of the previous sections, many conclusions can be extracted:

- In Figure 6.28, it is clear that unhelpful sentences tend to have high level of emotions (positive or negative), but unhelpful sentences (noise) are more probable to be neutral.

- The Figure 6.29 shows that sentences with high perplexity, which means they are not similar to book reviews sentences, have a higher probability of being unhelpful sentence than helpful.
- The Figure 6.30 gives an idea of sentiment correlation with sentences topics: sentences with book titles or author names have a high level of positive emotions, but sentences with personal information tend to be neutral. And sentences with book content narration are distributed over the area of emotional moderate level, with a higher probability of positive than negative.
- The Figure 6.31 gives an idea of the correlation between the sentences topics and their similarity to reviews: sentences with no book titles are more similar to reviews than the ones with book titles. Also, sentences with personal information tend to be similar to reviews. And sentences with book content narration show a slight more similarity with reviews sentences style than the sentences with no book content narration.

6.4.8. Discussion

This section analysed the correlation between sentiment intensity and reviews similarity toward sentences topics in long book-search queries, as they are considered multi-topic queries. First, by presenting the user queries and books collections, then extracting the sentiment intensity of each sentence of the queries. Followed by calculating the likelihood of each sentence being generated from a statistical language model based on reviews. And finally by presenting, in graphs, the relation between sentiment intensity score, language model score, and the topics of the sentences.

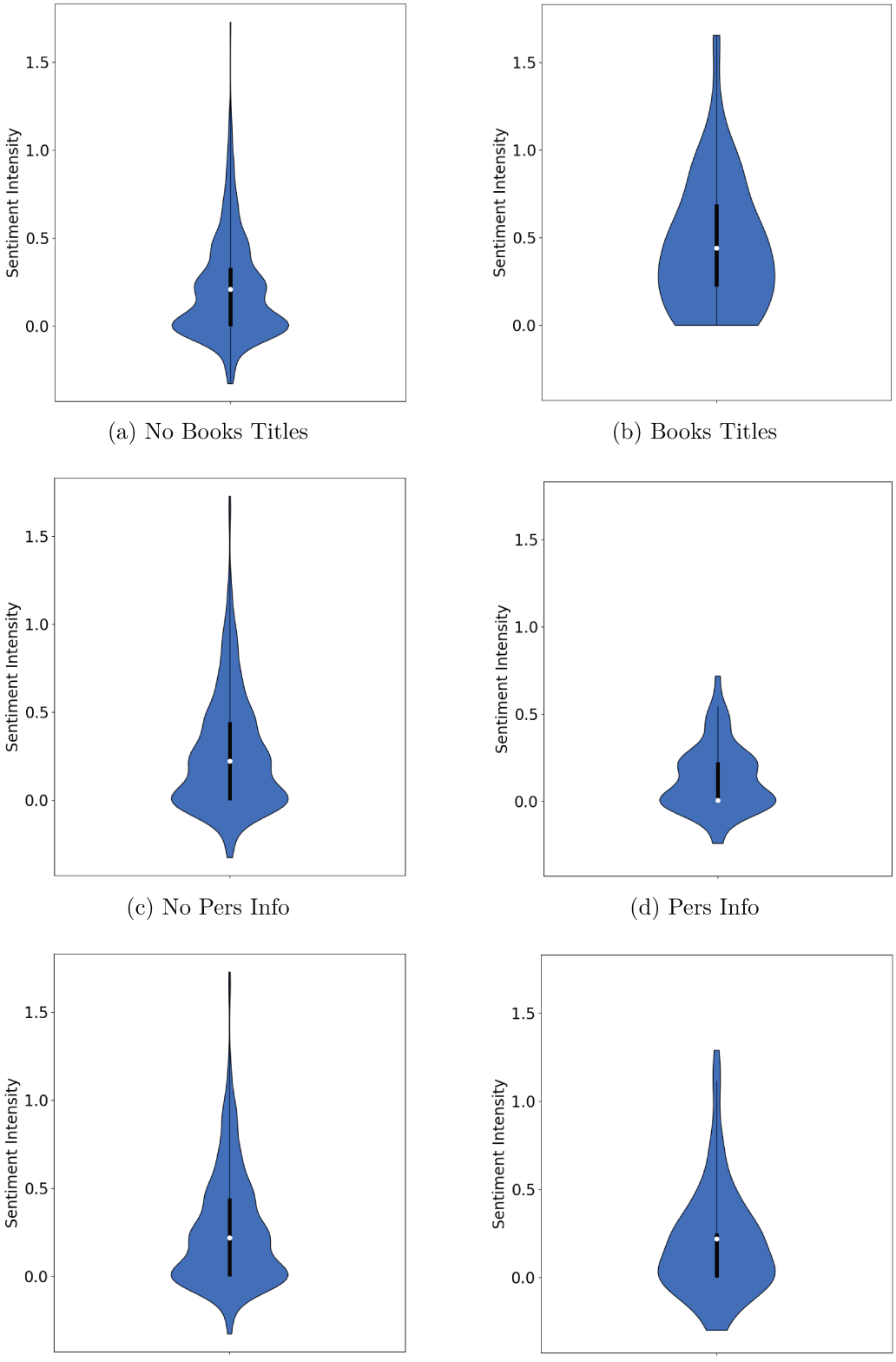
The graphs show that sentiment intensity can be an important feature to classify the sentences based on their helpfulness to the search. Since unhelpful sentences (or noise sentences) are more likely to be of sentiment polarity "neutral", than helpful sentences. Also, the graphs reveal that sentiment intensity can also be an important feature to classify the sentences based on their topic. It is clear in the graphs, that the sentences containing book titles are richer in sentiment and mostly positive compared to sentences not containing book titles. In addition, the graphs show that sentences with personal information tend to be neutral, in a higher probability than those with no personal information.

On the other hand, the graphs reveal that the similarity of sentences to reviews style can also be a feature to classify sentences by helpfulness and by their topic, but in a slightly lower level of importance than sentiment analysis. Similarity between sentences and book reviews style is higher for helpful sentences, for sentences with personal information and for sentences with narration of book content, but not for sentences containing book titles.

The previous analysis and conclusions gives a preview on the role that sentiment analysis and similarity to reviews can play in sentence classification of long book-search queries. The next task would be to test these conclusions by using sentiment analysis and similarity to reviews, as new features, in a supervised machine learning classification of sentences in long book-search queries.

6.5. Conclusion

In this chapter, we were able to experiment on two new methods to benefit from sentiment analysis in book retrieval. The first method is a pseudo relevance feedback, based on sentiment analysis, where experiments showed its ability to improve the ranking quality between 25% and 56%, on every retrieval method tested in this work. The second method is long-query sentences classification, with sentiment analysis as a feature. Such classification can open doors to many methods of retrieval improvement, like query reduction and sub-query creation. For this purpose, we analysed the correlation between the sentiment intensity within the sentence of user's query and the topic that the sentence holds. We were able to detect a high correlation regarding the usefulness of the sentence for the retrieval, and also regarding certain topics in sentences, like the content of book title, author name and personal information. These conclusions are a positive sign to the possibility of a sentiment analysis exploitation in the sentences classification of user's requests.



(a) No Books Titles

(b) Books Titles

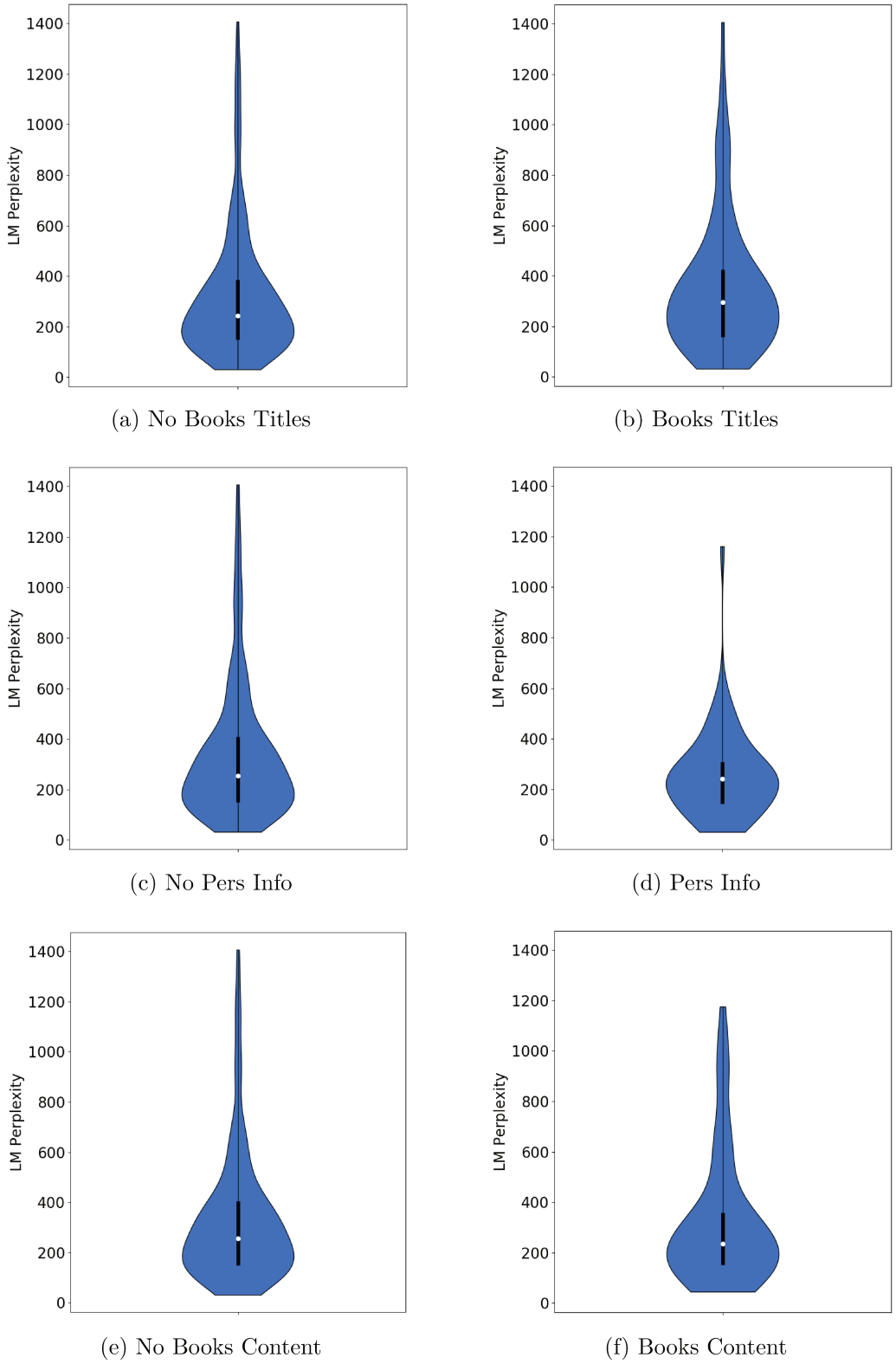
(c) No Pers Info

(d) Pers Info

(e) No Books Content

(f) Books Content

Figure 6.30.: The distribution of Sentiment between the topic of sentences: Books titles or authors names, Personal information and Narration of book content



(a) No Books Titles

(b) Books Titles

(c) No Pers Info

(d) Pers Info

(e) No Books Content

(f) Books Content

Figure 6.31.: The distribution of perplexity between the topic of sentences: Books titles or authors names, Personal information and Narration of book content.

General Conclusion

The presented thesis consisted on developing new approaches concerning the sentiment analysis field and the information retrieval and filtering fields. It is guided by the purpose of improving the multilingual OpenEdition platforms' quality, regarding documents' search and recommendation, using sentiment analysis in addition to other techniques. The main objectives in this thesis were, first, to find an approach for sentiment analysis prediction, easily applicable on different languages, therefore, low cost in time and annotated data, and second, to employ sentiment analysis in information retrieval using new approaches for book search quality improvement.

Summary of contributions

Throughout this manuscript, we presented our contributions, starting with a first part including proposed and evaluated methods related to sentiment analysis. The main contribution in that part is a semi-supervised method for sentiment intensity prediction, based on adapted to domain seed-words and word embedding models. As part of this method, two seed-words extraction methods are proposed and tested, a semi-automatic and an automatic method, in addition to the creation of word embedding models from large text corporas.The semi-supervised method proposed for sentiment analysis proved its efficiency in our manuscript while applied on sentiment intensity and sentiment polarity predictions, on microblogs (Twitter) and book reviews domains, and on several languages: English, French and Arabic. The tests showed that the seed-words and word embeddings method exceeded the results of the first proposed method of combining lexicon-based and search engines approaches, therefore, it was employed into the second part of the manuscript. Both sentiment analysis systems took part of Semantic Evaluation's (SemEval) workshops, achieving good results, as the only semi-supervised systems competing with supervised systems (mostly based on deep-learning approaches). In addition to the sentiment intensity prediction, the first part of this manuscript included a proposed method for an automatic creation of normalisation thesaurus, based on word embeddings. The purpose of these thesaurus is to return words to their normal and standard state, therefore, to substitute all misspellings with the correctly spelled words. Based on the tests, the employment of thesauruses proved its ability to improve the

sentiment intensity prediction of supervised machine learning systems.

In the second part of the manuscript, we presented our contribution in the information filtering and retrieval fields, put in an application for documents' recommendation and search. First, an unsupervised method for bibliographical zone detection is suggested and tested on a corpora of articles, achieving very good results. The method is the pre-step for a future work concerning the creation of a graph, based on an inter documents citation. Such graph would serve for documents' recommendation, in addition to the possibility of using it for retrieval re-ranking. In the second section of that manuscript part, we exhibited two new employment of sentiment analysis in book search. The first concerns the extraction of terms from highly emotional book reviews' sentences, for a pseudo relevance feedback process, where the tests proved its ability to improve the book search quality. The second employment of sentiment analysis in book search concerned the classification of long book search query sentences by topic. For that purpose, we studied the correlation between the sentiment and the information within the query's sentences, concerning the helpfulness of the sentences to the search, and the topic of these sentences. The study showed an interesting relation between the sentiment and the topic of the sentence, in addition to a relation between sentences' similarity to book reviews style and the sentences' topic.

Perspectives

This thesis covered a variety of research fields, and it opened access to several future work conceivable in the short, medium and long term.

Regarding the work done in the sentiment intensity prediction, in short term, we attempt to apply and test the automatic extraction of adapted seed-words method on several languages and domains, since it is able to transform our proposed sentiment intensity method to an unsupervised method, which can lead to a total automatic application of the method on any domain and any language. In medium term, we are planning on introducing the concept of $Mixed$ polarity in the sentiment polarity prediction by predicting the polarities of sub-sentences and then detecting opposite polarities in the same sentence. In long term, we are planning on experimenting the use of the created lists of seed-words with a deep learning method; Since words with opposite polarities might be mapped as similar words by word embeddings, we attempt to benefit from seed-words in a deep learning training that initialize the word embeddings weights, incorporating sentiment information into these weights.

Regarding the created normalisation thesaurus, they have been used in this thesis for sentiment analysis improvement purposes only, but they can be exploited, in short term, for many other purposes. For example, in the thesaurus of Arabic language, we were able to detect pairs of different dialect words with their standard-form word, what offers the possibility of creating inter-dialects thesaurus of Arabic language.

Concerning the work covering the sentiment analysis employment in information retrieval:

- First, for the sentiment oriented pseudo relevance feedback new approach, we consider, in short term, exploring a combination of several terms extraction methods, like term frequency and term weighting, with our sentiment analysis based method to extract terms from book reviews. And in medium term, we attempt to profit from our terms extraction method for an aspects extraction method, and then apply it in both search query and book reviews to test a new re-ranking method based on these aspects similarities.

- Second, for the sentences classification in long book search queries, we plan, in short term, to create a large annotated corpora, with a similar annotation as the Figure 3.5 in Chapter 6, and use it as a training corpora in a machine learning method, using sentiment intensity score and similarity to reviews score as features. Then, based on the ability of the created machine learning model to classify the sub-query by their topic, we attempt, in medium term, to develop a multi-approach book search system that process every sub-query differently according to its topic. For example, the detected unhelpful to search sentences can be removed as a query reduction procedure, and the detection of sentences with personal information can be used for a personalised search, etc.

Bibliography

[All+17] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, et al. "A brief survey of text mining: Classification, clustering and extraction techniques". In: *arXiv preprint arXiv:1707.02919* (2017) (cit. on p. 63).

[Alt92] Naomi S Altman. "An introduction to kernel and nearest-neighbor nonparametric regression". In: *The American Statistician* 46.3 (1992), pp. 175–185 (cit. on p. 39).

[Ama03] Giambattista Amati. "Probability models for information retrieval based on divergence from randomness". PhD thesis. University of Glasgow, 2003 (cit. on pp. 94, 128).

[Aw+06] AiTi Aw, Min Zhang, Juan Xiao, et al. "A phrase-based statistical model for SMS text normalization". In: *COLING*. ACL. 2006, pp. 33–40 (cit. on p. 79).

[Ay +18] Betül Ay Karakuş, Muhammed Talo, Ibrahim Riza Hallaç, et al. "Evaluating deep learning models for sentiment classification". In: *Concurrency and Computation: Practice and Experience* 30.21 (2018), e4783 (cit. on p. 66).

[BES10] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. "Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining." In: *LREC*. Vol. 10. 2010. 2010, pp. 2200–2204 (cit. on pp. 52, 64).

[BFC18] Ismail Badache, Sébastien Fournier, and Adrian-Gabriel Chifu. "Predicting Contradiction Intensity: Low, Strong or Very Strong?" In: *SIGIR*. 2018 (cit. on p. 122).

[BNP04] Jing Bai, Jian-Yun Nie, and François Paradis. "Using language models for text classification". In: *Proceedings of the Asia Information Retrieval Symposium*. 2004 (cit. on p. 135).

[Bai+05] Jing Bai, Dawei Song, Peter Bruza, et al. "Query expansion using term relationships in language models for information retrieval". In: *CIKM*. ACM. 2005, pp. 688–695 (cit. on p. 122).

[BS97] Marko Balabanović and Yoav Shoham. "Fab: content-based, collaborative recommendation". In: *Communications of the ACM* 40.3 (1997), pp. 66–72 (cit. on p. 100).

[BCB94] Brian T Bartell, Garrison W Cottrell, and Richard K Belew. "Automatic combination of multiple ranked retrieval systems". In: *SIGIR'94*. Springer. 1994, pp. 173–181 (cit. on p. 97).

[BMW07] Holger Bast, Debapriyo Majumdar, and Ingmar Weber. "Efficient interactive query expansion with complete search". In: *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM. 2007, pp. 857–860 (cit. on p. 96).

[Bei+05] Steven M Beitzel, Eric C Jensen, Ophir Frieder, et al. "Improving automatic query classification via semi-supervised learning". In: *IEEE*. IEEE. 2005 (cit. on p. 135).

[Bel+95] Nicholas J. Belkin, Paul Kantor, Edward A. Fox, et al. "Combining the evidence of multiple query representations for information retrieval". In: *Information Processing & Management* 31.3 (1995), pp. 431–448 (cit. on p. 97).

[BC92] Nicholas J Belkin and W Bruce Croft. "Information filtering and information retrieval: two sides of the same coin". In: *Communications of the ACM*. Citeseer. 1992 (cit. on p. 102).

[EA13] Samhaa R El-Beltagy and Ahmed Ali. "Open issues in the sentiment analysis of Arabic social media: A case study". In: *Innovations in information technology (iit), 2013 9th international conference on*. IEEE. 2013, pp. 215–220 (cit. on p. 48).

[BB13] Chahinez Benkoussas and Patrice Bellot. "Book Recommendation based on Social Information." In: *CLEF (working notes)*. 2013 (cit. on p. 121).

[BOB15] Chahinez Benkoussas, Anaïs Ollagnier, and Patrice Bellot. "Book Recommendation Using Information Retrieval Methods and Graph Analysis." In: *CLEF (Working Notes)*. 2015 (cit. on pp. 97, 99).

[BN17] Thales Felipe Costa Bertaglia and Maria das Graças Volpe Nunes. "Exploring word embeddings for unsupervised textual user-generated content normalization". In: *arXiv preprint arXiv:1704.02963* (2017) (cit. on p. 79).

[BNJ03] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022 (cit. on p. 39).

[BDP07] John Blitzer, Mark Dredze, and Fernando Pereira. "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification". In: *ACL*. 2007, pp. 440–447 (cit. on p. 62).

[Bon+12] Ludovic Bonnefoy, Romain Deveaud, Patrice Bellot, et al. "Do social information help book search?" In: *Workshop INEX*. 2012, p. 109 (cit. on p. 125).

[Bor95] Jean-Charles de Borda. “On elections by ballot”. In: *Classics of social choice, eds. I. McLean, AB Urken, and F. Hewitt* (1995), pp. 83–89 (cit. on p. 97).

[BM00] Eric Brill and Robert C Moore. “An improved error model for noisy channel spelling correction”. In: *ACL*. ACL. 2000, pp. 286–293 (cit. on p. 78).

[Cam+10] Erik Cambria, Robert Speer, Catherine Havasi, et al. “SenticNet: A Publicly Available Semantic Resource for Opinion Mining.” In: *AAAI fall symposium: commonsense knowledge*. Vol. 10. 0. 2010 (cit. on p. 48).

[Che+17] Tao Chen, Ruifeng Xu, Yulan He, et al. “Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN”. In: *Expert Systems with Applications* (2017), pp. 221–230. ISSN: 0957-4174 (cit. on pp. 134, 135).

[CH90] Kenneth Ward Church and Patrick Hanks. “Word association norms, mutual information, and lexicography”. In: *Computational linguistics* 16.1 (1990), pp. 22–29 (cit. on p. 38).

[Cla+18] Vincent Claveau, Anne-Lyse Minard, Peggy Cellier, et al. *Actes de la conférence Traitement Automatique de la Langue Naturelle, TALN 2018. Volume 2: Rencontres Jeunes Chercheurs, démonstrations, atelier DeFT. France. 2018.* 2018 (cit. on pp. 61, 71).

[Col97] Michael Collins. “Three generative, lexicalised models for statistical parsing”. In: *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*. ACL. 1997, pp. 16–23 (cit. on p. 137).

[CV95] Corinna Cortes and Vladimir Vapnik. “Support-vector networks”. In: *Machine learning* 20.3 (1995), pp. 273–297 (cit. on p. 39).

[CT91] TM Cover and JA Thomas. *Elements of Information Theory. John Willey, New York*. 1991 (cit. on p. 95).

[DG06] Jesse Davis and Mark Goadrich. “The Relationship Between Precision-Recall and ROC Curves”. In: *International Conference on Machine Learning (ICML)* (2006) (cit. on p. 109).

[DS10] Gianluca Demartini and Stefan Siersdorfer. “Dear search engine: what’s your opinion about...?: sentiment analysis for semantic enrichment of web search results”. In: *Proceedings of the 3rd International Semantic Search Workshop*. ACM. 2010, p. 4 (cit. on p. 47).

[DV09] Eustache Diemert and Gilles Vandelle. “Unsupervised query categorization using automatically-built concept graphs”. In: *WWW*. ACM. 2009, pp. 461–470 (cit. on p. 135).

[ET17] Gülşen Eryiğit and Dilara Torunoğlu-Selamet. "Social media text normalization for Turkish". In: *Natural Language Engineering* 23.6 (2017), pp. 835–875 (cit. on p. 79).

[ER15] Ramy Eskander and Owen Rambow. "SLSA: A Sentiment Lexicon for Standard Arabic." In: *EMNLP*. 2015, pp. 2545–2550 (cit. on p. 48).

[ES06] Andrea Esuli and Fabrizio Sebastiani. "Sentiwordnet: A publicly available lexical resource for opinion mining." In: *LREC*. Vol. 6. Citeseer. 2006, pp. 417–422 (cit. on p. 47).

[Gia+17] Maria Giatsoglou, Manolis G Vozalis, Konstantinos Diamantaras, et al. "Sentiment analysis leveraging emotions and word embeddings". In: *Expert Systems with Applications* 69 (2017), pp. 214–224 (cit. on p. 40).

[GBH09] Alec Go, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision". In: *CS224N Project Report, Stanford* 1.12 (2009) (cit. on pp. 52, 58, 65).

[God+15] Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, et al. "Multimedia Lab $ $ ACL WNUT NER Shared Task: Named Entity Recognition for Twitter Microposts using Distributed Word Representations". In: *Proceedings of the Workshop on Noisy User-generated Text*. 2015, pp. 146–153 (cit. on pp. 66, 70).

[Gre00] Ed Greengrass. "Information retrieval: A survey". In: *University of Maryland, Baltimore County* (2000) (cit. on p. 92).

[GZD10] Quanquan Gu, Jie Zhou, and Chris Ding. "Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs". In: *Proceedings of the 2010 SIAM international conference on data mining*. SIAM. 2010, pp. 199–210 (cit. on p. 100).

[HBB15] Hussam Hamdan, Patrice Bellot, and Frederic Bechet. "Sentiment Lexicon-Based Features for Sentiment Analysis in Short Text." In: *Research in Computing Science* 90 (2015), pp. 217–226 (cit. on p. 86).

[HB11] Bo Han and Timothy Baldwin. "Lexical normalisation of short text messages: Makn sens a# twitter". In: *ACL: Human Language Technologies-Volume 1*. ACL. 2011, pp. 368–378 (cit. on p. 79).

[HW79] John A Hartigan and Manchek A Wong. "Algorithm AS 136: A k-means clustering algorithm". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28.1 (1979), pp. 100–108 (cit. on p. 63).

[HM16] Ruining He and Julian McAuley. "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering". In: *WWW*. International World Wide Web Conferences Steering Committee. 2016, pp. 507–517 (cit. on pp. 67, 73, 124, 131, 137).

[Hta18] Amal Htait. “Adapted Sentiment Similarity Seed Words For French Tweets’ Polarity Classification”. In: *Actes de DEFT* (2018) (cit. on pp. 31, 71).

[HFB16a] Amal Htait, Sébastien Fournier, and Patrice Bellot. “Bilbo-Val: Automatic Identification of Bibliographical Zone in Papers”. In: *LREC*. 2016 (cit. on pp. 31, 104).

[HFB16b] Amal Htait, Sébastien Fournier, and Patrice Bellot. “LSIS at SemEval-2016 Task 7: Using web search engines for English and Arabic unsupervised sentiment intensity prediction”. In: *SemEval*. 2016, pp. 469–473 (cit. on pp. 31, 54).

[HFB16c] Amal Htait, Sébastien Fournier, and Patrice Bellot. “SBS 2016: Combining Query Expansion Result and Books Information Score for Book Recommendation.” In: *CLEF (Working Notes)*. 2016, pp. 1115–1122 (cit. on p. 125).

[HFB17] Amal Htait, Sébastien Fournier, and Patrice Bellot. “LSIS at SemEval-2017 Task 4: Using Adapted Sentiment Similarity Seed Words For English and Arabic Tweet Polarity Classification”. In: *SemEval*. 2017, pp. 718–722 (cit. on pp. 31, 70).

[HFB18] Amal Htait, Sébastien Fournier, and Patrice Bellot. “Unsupervised Creation of Normalization Dictionaries for Micro-Blogs in Arabic, French and English”. In: *Computación y Sistemas* 22.3 (2018) (cit. on pp. 31, 78).

[HFB19] Amal Htait, Sébastien Fournier, and Patrice Bellot. “Sentiment Analysis and Sentence Classification in Long Book-Search Queries”. In: *CICLing* (2019) (cit. on p. 31).

[HL04] Minqing Hu and Bing Liu. “Mining and summarizing customer reviews”. In: *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining KDD 04* 04 (2004), p. 168 (cit. on p. 51).

[IBB15] Melanie Imhof, Ismail Badache, and Mohand Boughanem. “Multimodal social book search”. In: *6th Conference on Multilingual and Multimodal Information Access Evaluation (CLEF 2015)*. 2015, pp–1 (cit. on p. 121).

[Jen12] Inge Hviid Jensen. “Information Retrieval and Sentiment Analysis for Scientific Literature”. In: *Thesis* (2012) (cit. on p. 121).

[JCS04] Rong Jin, Joyce Y Chai, and Luo Si. “An automatic weighting scheme for collaborative filtering”. In: *SIGIR*. ACM. 2004, pp. 337–344 (cit. on p. 100).

[JWR00] K Sparck Jones, Steve Walker, and Stephen E. Robertson. "A probabilistic model of information retrieval: development and comparative experiments: Part 2". In: *Information processing & management* 36.6 (2000), pp. 809–840 (cit. on p. 94).

[Ju+12] Shengfeng Ju, Shoushan Li, Yan Su, et al. "Dual word and document seed selection for semi-supervised sentiment classification". In: *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM. 2012, pp. 2295–2298 (cit. on p. 47).

[Kam+04] Jaap Kamps, Maarten Marx, Robert J Mokken, et al. "Using WordNet to measure semantic orientations of adjectives." In: *LREC*. Vol. 4. Citeseer. 2004, pp. 1115–1118 (cit. on p. 47).

[KK03] In-Ho Kang and GilChang Kim. "Query type classification for web document retrieval". In: *SIGIR*. ACM. 2003, pp. 64–71 (cit. on p. 135).

[Ken38] Maurice G Kendall. "A new measure of rank correlation". In: *Biometrika* 30.1/2 (1938), pp. 81–93 (cit. on p. 42).

[Kim+12a] Young-Min Kim, Patrice Bellot, Jade Tavernier, et al. "Evaluation of BILBO reference parsing in digital humanities via a comparison of different tools". In: *In ACM*. 2012, pp. 209–212 (cit. on p. 104).

[Kim+12b] Young-Min Kim, Patrice Bellot, Jade Tavernier, et al. "Evaluation of BILBO reference parsing in digital humanities via a comparison of different tools". In: *Proceedings of the 2012 ACM symposium on Document engineering - DocEng '12* (2012), pp. 209–212 (cit. on pp. 105, 106).

[KZM14] Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. "Sentiment analysis of short informal texts". In: *Journal of Artificial Intelligence Research* 50 (2014), pp. 723–762 (cit. on p. 47).

[KYD08] Catherine Kobus, François Yvon, and Géraldine Damnati. "Transcrire les SMS comme on reconnaît la parole". In: *Actes de la Conférence sur le Traitement Automatique des Langues (TALN'08)*. 2008, pp. 128–138 (cit. on p. 79).

[Koo+15] Marijn Koolen, Toine Bogers, Maria Gäde, et al. "Overview of the CLEF 2015 social book search lab". In: *CLEF*. Springer. 2015, pp. 545–564 (cit. on p. 118).

[KBK16] Marijn Koolen, Toine Bogers, and Jaap Kamps. "Overview of the SBS 2016 Suggestion Track". In: *CLEF (Working Notes)*. 2016, pp. 1039–1052 (cit. on p. 118).

[KS05] Moshe Koppel and Jonathan Schler. "Using neutral examples for learning polarity". In: *International Joint Conference on Artificial Intelligence*. Vol. 19. 2005, p. 1616 (cit. on p. 37).

[KB15] Yehuda Koren and Robert Bell. "Advances in collaborative filtering". In: *Recommender systems handbook*. Springer, 2015, pp. 77–118 (cit. on p. 100).

[KA06] Giridhar Kumaran and James Allan. "Simple questions to improve pseudo-relevance feedback results". In: *SIGIR*. ACM. 2006, pp. 661–662 (cit. on pp. 96, 123).

[LJ01] Adenike M Lam-Adesina and Gareth JF Jones. "Applying summarization techniques for term selection in relevance feedback". In: *SIGIR*. ACM. 2001, pp. 1–9 (cit. on p. 96).

[LF73] Frederick Wilfrid Lancaster and Emily Gallup Fayen. *Information retrieval: on-line*. Springer, 1973 (cit. on p. 92).

[Lee95] Joon Ho Lee. "Combining multiple evidence from different properties of weighting schemes". In: *SIGIR*. ACM. 1995, pp. 180–188 (cit. on p. 125).

[LSB18] Zhang Lei, Wang Shuai, and Liu Bing. "Deep learning for sentiment analysis: A survey". In: *Cornell Science Library* (2018) (cit. on p. 39).

[Li+12] Shipeng Li, Gang Zeng, Qifa Ke, et al. "Fast approximate k-means via cluster closures". In: *IEEE*. IEEE. 2012, pp. 3037–3044 (cit. on p. 64).

[Liu12] Bing Liu. "Sentiment analysis and opinion mining". In: *Synthesis lectures on human language technologies* 5.1 (2012), pp. 1–167 (cit. on p. 46).

[Liu15] Bing Liu. *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press, 2015 (cit. on p. 36).

[LZ12] Bing Liu and Lei Zhang. "A survey of opinion mining and sentiment analysis". In: *Mining text data*. Springer, 2012, pp. 415–463 (cit. on p. 36).

[LDS11] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. "Content-based recommender systems: State of the art and trends". In: *Recommender systems handbook*. Springer, 2011, pp. 73–105 (cit. on p. 100).

[Maa+11] Andrew L Maas, Raymond E Daly, Peter T Pham, et al. "Learning word vectors for sentiment analysis". In: *ACL: Human language technologies-volume 1*. ACL. 2011, pp. 142–150 (cit. on pp. 40, 47).

[Mac+05] Craig Macdonald, Ben He, Vassilis Plachouras, et al. "University of Glasgow at TREC 2005: Experiments in Terabyte and Enterprise Tracks with Terrier." In: *TREC*. 2005 (cit. on p. 94).

[MK60] Melvin Earl Maron and John L Kuhns. "On relevance, probabilistic indexing and information retrieval". In: *Journal of the ACM (JACM)* 7.3 (1960), pp. 216–244 (cit. on p. 94).

[MN+98] Andrew McCallum, Kamal Nigam, et al. “A comparison of event models for naive bayes text classification”. In: *AAAI-98 workshop on learning for text categorization*. Vol. 752. 1. Citeseer. 1998, pp. 41–48 (cit. on p. 39).

[MC05] Donald Metzler and W Bruce Croft. “A Markov random field model for term dependencies”. In: *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. 2005, pp. 472–479 (cit. on p. 95).

[Mik+13] Tomas Mikolov, Kai Chen, Greg Corrado, et al. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013) (cit. on pp. 40, 65–67).

[Mil95] George A Miller. “WordNet: a lexical database for English”. In: *Communications of the ACM* 38.11 (1995), pp. 39–41 (cit. on p. 96).

[MT13] Saif M Mohammad and Peter D Turney. “Crowdsourcing a word–emotion association lexicon”. In: *Computational Intelligence* 29.3 (2013), pp. 436–465 (cit. on pp. 52, 59, 74).

[NAA15] Mahmoud Nabil, Mohamed Aly, and Amir Atiya. “Astd: Arabic sentiment tweets dataset”. In: *EMNLP*. 2015, pp. 2515–2519 (cit. on pp. 46, 52).

[Nak+16] Preslav Nakov, Alan Ritter, Sara Rosenthal, et al. “SemEval-2016 task 4: Sentiment analysis in Twitter”. In: *Semeval*. 2016, pp. 1–18 (cit. on pp. 50, 51).

[Oll17] Anaïs Ollagnier. “Analyse de requetes en langue naturelle et extraction d’informations bibliographiques pour une recherche de livres orientée contenu efficace”. PhD thesis. Aix-Marseille, 2017 (cit. on p. 104).

[OFB15] Anaïs Ollagnier, Sébastien Fournier, and Patrice Bellot. “Analyse en dépendance et classification de requêtes en langue naturelle, application à la recommandation de livres.” In: *Traitement Automatique des Langues* (2015) (cit. on p. 135).

[OFB16] Anaïs Ollagnier, Sébastien Fournier, and Patrice Bellot. “A supervised Approach for detecting allusive bibliographical references in scholarly publications”. In: *Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics*. ACM. 2016, p. 36 (cit. on p. 104).

[OFB18] Anaïs Ollagnier, Sébastien Fournier, and Patrice Bellot. “BIBLME RecSys: Harnessing bibliometric measures for a scholarly paper recommender system”. In: *BIR 2018 workshop on bibliometric-enhanced information retrieval*. 2018 (cit. on p. 100).

[PP10] Alexander Pak and Patrick Paroubek. “Twitter as a corpus for sentiment analysis and opinion mining.” In: *LREC*. Vol. 10. 2010. 2010, pp. 1320–1326 (cit. on p. 46).

[PY+10] Sinno Jialin Pan, Qiang Yang, et al. “A survey on transfer learning”. In: *IEEE Transactions on knowledge and data engineering* 22.10 (2010), pp. 1345–1359 (cit. on p. 106).

[PL08] Bo Pang and Lillian Lee. “Opinion mining and sentiment analysis”. In: *Foundations and Trends in Information Retrieval* 2.1–2 (2008), pp. 1–135 (cit. on p. 36).

[PLV02] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. “Thumbs up?: sentiment classification using machine learning techniques”. In: *EMNLP*. ACL. 2002, pp. 79–86 (cit. on p. 38).

[PC98] Jay Michael Ponte and W Bruce Croft. “A language modeling approach to information retrieval”. PhD thesis. University of Massachusetts at Amherst, 1998 (cit. on p. 95).

[Pow11] David Martin Powers. “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation”. In: (2011) (cit. on p. 86).

[PT09] Rudy Prabowo and Mike Thelwall. “Sentiment analysis: A combined approach”. In: *Journal of Informetrics* 3.2 (2009), pp. 143–157 (cit. on p. 47).

[RR16] Eshrag Refaee and Verena Rieser. “iLab-Edinburgh at SemEval-2016 Task 7: A hybrid approach for determining sentiment intensity of Arabic Twitter phrases”. In: *SemEval*. 2016, pp. 474–480 (cit. on p. 55).

[Roc71] Joseph John Rocchio. “Relevance feedback in information retrieval”. In: *The SMART retrieval system: experiments in automatic document processing* (1971), pp. 313–323 (cit. on p. 96).

[RFN17] Sara Rosenthal, Noura Farra, and Preslav Nakov. “SemEval-2017 task 4: Sentiment analysis in Twitter”. In: *SemEval*. 2017, pp. 502–518 (cit. on p. 70).

[Ros+15] Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, et al. “Semeval-2015 task 10: Sentiment analysis in twitter”. In: *SemEval*. 2015, pp. 451–463 (cit. on p. 50).

[Ros+14] Sara Rosenthal, Preslav Nakov, Alan Ritter, et al. “Semeval-2014 task 9: Sentiment analysis in twitter”. In: *SemEval 2014* (2014) (cit. on p. 86).

[Rus80] James A Russell. “A circumplex model of affect.” In: *Journal of personality and social psychology* 39.6 (1980), p. 1161 (cit. on p. 37).

[SWY75] Gerard Salton, Anita Wong, and Chung-Shu Yang. "A vector space model for automatic indexing". In: *Communications of the ACM* 18.11 (1975), pp. 613–620 (cit. on p. 93).

[Sha48] Claude Elwood Shannon. "A mathematical theory of communication". In: *Bell system technical journal* 27.3 (1948), pp. 379–423 (cit. on p. 78).

[SSP+03] Patrice Y Simard, David Steinkraus, John C Platt, et al. "Best practices for convolutional neural networks applied to visual document analysis." In: *Icdar*. Vol. 3. 2003. 2003 (cit. on p. 39).

[Spe04] Charles Spearman. "The proof and measurement of association between two things". In: *American journal of Psychology* 15.1 (1904), pp. 72–101 (cit. on p. 42).

[Sri15] Vivek Kumar Rangarajan Sridhar. "Unsupervised text normalization using distributed representations of words and phrases". In: *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. 2015, pp. 8–16 (cit. on pp. 78, 79).

[Sto02] Andreas Stolcke. "SRILM-an extensible language modeling toolkit". In: *Seventh international conference on spoken language processing*. 2002 (cit. on p. 137).

[Tou+03a] Kristina Toutanova, Dan Klein, Christopher D Manning, et al. "Feature-rich part-of-speech tagging with a cyclic dependency network". In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. ACL. 2003, pp. 173–180 (cit. on p. 123).

[Tou+03b] Kristina Toutanova, Dan Klein, Christopher D Manning, et al. "Feature-rich part-of-speech tagging with a cyclic dependency network". In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. ACL. 2003, pp. 173–180 (cit. on p. 136).

[Tur02] Peter D Turney. "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews". In: *ACL*. ACL. 2002, pp. 417–424 (cit. on pp. 38, 47, 49, 68).

[TL03] Peter D Turney and Michael L Littman. "Measuring praise and criticism: Inference of semantic orientation from association". In: *ACM Transactions on Information Systems (TOIS)* 21.4 (2003), pp. 315–346 (cit. on pp. 38, 47, 49, 57, 58).

[VV95] VN Valdimir and N Vapnik. *The nature of statistical learning theory*. 1995 (cit. on p. 39).

[Voo94] Ellen M Voorhees. "Query expansion using lexical-semantic relations". In: *SIGIR*. Springer. 1994, pp. 61–69 (cit. on p. 96).

[Wan+06] Fei Wang, Sheng Ma, Liuzhong Yang, et al. “Recommendation on item graphs”. In: *ICDM*. IEEE. 2006, pp. 1119–1123 (cit. on p. 100).

[WZL16] Feixiang Wang, Zhihua Zhang, and Man Lan. “Ecnu at semeval-2016 task 7: An enhanced supervised learning method for lexicon sentiment intensity ranking”. In: *SemEval*. 2016, pp. 491–496 (cit. on p. 55).

[WWH05] 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*. ACL. 2005, pp. 347–354 (cit. on pp. 51, 52).

[XC17] Jinxi Xu and W Bruce Croft. “Quary expansion using local and global document analysis”. In: *SIGIR*. Vol. 51. 2. ACM. 2017, pp. 168–175 (cit. on p. 96).

[Yag88] Ronald R Yager. “On ordered weighted averaging aggregation operators in multicriteria decisionmaking”. In: *IEEE Transactions on systems, Man, and Cybernetics* 18.1 (1988), pp. 183–190 (cit. on pp. 97, 98).

[YLF17] Xiangbin Yan, Yumei Li, and Weiguo Fan. “Identifying domain relevant user generated content through noise reduction: a test in a Chinese stock discussion forum”. In: *Information Discovery and Delivery* 45.4 (2017), pp. 181–193 (cit. on p. 79).

[YC18] Heng-Li Yang and August FY Chao. “Sentiment annotations for reviews: an information quality perspective”. In: *Online Information Review* 42.5 (2018), pp. 579–594 (cit. on p. 122).

[YH14] Zheng Ye and Jimmy Xiangji Huang. “A simple term frequency transformation model for effective pseudo relevance feedback”. In: *SIGIR*. ACM. 2014, pp. 323–332 (cit. on pp. 96, 122).

[Zad83] Lotfi A Zadeh. “A computational approach to fuzzy quantifiers in natural languages”. In: *Computers & Mathematics with applications* 9.1 (1983), pp. 149–184 (cit. on p. 98).

[Zha+14] Yongfeng Zhang, Guokun Lai, Min Zhang, et al. “Explicit factor models for explainable recommendation based on phrase-level sentiment analysis”. In: *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM. 2014, pp. 83–92 (cit. on p. 121).