جامعة سيدي محمد بن عبد الله بفاس
UNIVERSITE SIDI MOHAMED BEN ABDALLAH DE FES
كلية العلوم ظهر المهراز فاس
FACULTE DES SCIENCES DHAR EL MEHRAZ FES



N° d'Ordre : **17/2021**

Doctoral Training Center: *STIC*

Discipline: *Computer Science, Data Science, Artificial Intelligence*

Major Subject: *Computer Science*

Laboratory: *LISAC*

DOCTORAL THESIS

Submitted by

BENLAHBIB Abdessamad

Mining Customers' Opinions for Online Reputation Generation and Visualization in e-Commerce Platforms

Defended on 30/03/2021 in front of the committee composed of:

Pr. Jaouad Boumhidi	Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez	Chair
Pr. Abdellah Madani	Faculty of Sciences, Chouaib Doukkali University, El Jadida	Reviewer
Pr. Hajar Mousannif	Faculty of Sciences Semlalia, Cadi Ayyad University, Marrakech	Reviewer
Pr. Jamal Riffi	Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez	Reviewer
Pr. Said Benhlima	Faculty of Sciences, Moulay Ismail University, Meknes	Examiner
Pr. Ahmed Zinedine	Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez	Examiner
Pr. Ali Yahyaouy	Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez	Examiner
Pr. Chakir Loqman	Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez	Examiner
Pr. El Habib Nfaoui	Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez	Supervisor

Academic year: 2020-2021

Contents

Acknowledgments **10**

Abstract **11**

Abstract français **13**

Introduction **15**

1 Background and Literature Review **19**

1.1 Introduction 19

1.2 NLP techniques for online decision making in E-commerce 19

1.2.1 Feature-based opinion summarization 19

1.2.2 Reputation generation 20

1.3 Document-level Sentiment Analysis 21

1.4 Fine-grained Sentiment Analysis on the 5-class Stanford Sentiment Treebank (SST-5) dataset 25

1.5 Bidirectional Encoder Representations from Transformers (BERT) 26

1.6 Conclusion 27

2 Sentiment Analysis and Opinion Fusion for Reputation Generation **28**

2.1 Introduction . . . 28
2.2 Problem definition . . . 30
2.3 Proposed approach . . . 31
2.3.1 System overview . . . 31
2.3.2 Polarity classification phase . . . 32
2.3.3 Fusion and grouping phase . . . 34
2.3.4 Reputation generation . . . 34
2.4 Experimental results and discussion . . . 36
2.4.1 Training phase . . . 36
2.4.2 Datasets collection and preprocess . . . 36
2.4.3 Sentiment classification . . . 38
2.4.4 Opinion fusion and grouping . . . 40
2.4.5 Reputation generation . . . 41
2.4.6 Opinion fusion threshold t_0 . . . 46
2.4.7 Classification phase impact on reputation generation . . . 48
2.5 Summary and Limitations . . . 49

3 Fine-grained Opinion Mining and Semantic Analysis for Reputation Generation 53

3.1 Introduction . . . 53
3.2 Problem definition . . . 54
3.3 Proposed system . . . 55
3.3.1 System overview . . . 55
3.3.2 Fine-grained sentiment analysis . . . 56
3.3.3 Opinion groups custom scores . . . 56
3.3.4 Reputation generation . . . 59
3.4 Experimental evaluation . . . 60

3.4.1 Dataset gathering . . . 60
3.4.2 Training phase and fine-grained opinion mining . . . 60
3.4.3 Reputation evaluation . . . 63
3.5 Conclusion, summary and future direction . . . 67

4 Aggregating Customer Review Attributes for Online Reputation Generation 69

4.1 Introduction . . . 69
4.2 Problem definition . . . 70
4.3 Proposed approach . . . 71
4.3.1 System overview . . . 71
4.3.2 Data collection and preprocessing . . . 72
4.3.3 Review helpfulness . . . 73
4.3.4 Review Time . . . 75
4.3.5 Review sentiment orientation . . . 77
4.3.6 Review score . . . 78
4.3.7 Reputation generation . . . 79
4.3.8 Reputation visualization . . . 81
4.4 Experiment results . . . 81
4.4.1 Data collection and preprocessing . . . 81
4.4.2 Sentiment analysis . . . 82
4.4.3 Reputation visualization . . . 84
4.4.4 System evaluation . . . 85
4.4.5 Further discussion . . . 88
4.5 Conclusion . . . 91

5 Reputation Generation and Visualization for Amazon's Products 93

5.1 Introduction . . . 93

5.2 Problem definition 94
5.3 Proposed approach 95
5.3.1 System overview 95
5.3.2 Review structure 96
5.3.3 Review helpfulness 96
5.3.4 Review Time 98
5.3.5 User credibility 98
5.3.6 Review score 99
5.3.7 Reputation generation 100
5.3.8 Reputation visualization 101
5.4 Experiment results 101
5.4.1 Data collection 101
5.4.2 Sentiment analysis 101
5.4.3 Reputation generation and visualization 102
5.5 Conclusion 103

Conclusions **105**

Bibliography **106**

List of Figures

1.1 Kangale et al. [1] feature-based summary 20

1.2 Typical class labels (or intensities) for fine-grained sentiment classification 25

1.3 Bidirectional Encoder Representations from Transformers (BERT) pre-training architecture [2] 27

1.4 Overall pre-training and fine-tuning procedures for BERT [2] 27

2.1 Pipeline of our work 32

2.2 Dataset Sample 37

2.3 Sentiment classification accuracy of Algorithm 1, Naïve Bayes and SVM 39

2.4 Distribution of sentiments over reviews 41

2.5 IMDb users weighted average vote for The Godfather I movie 42

2.6 Comparison between Yan et al. [3] method and our contribution for dataset 1 and dataset 2 43

2.7 Comparison between Yan et al. [3] method and our contribution for dataset 3 and dataset 4 43

2.8 Comparison between Yan et al. [3] method and our contribution for dataset 5 and dataset 6 44

2.9 Comparison between Yan et al. [3] method and our contribution for dataset 7 and dataset 8 44

2.10 Comparison between Yan et al. [3] method and our contribution for dataset 9 and dataset 10 44

2.11 Absolute Error (AE) of reputation value computed by our method 46

2.12 $MAER_{t_0}$ between IMDb weighted average votes and the reputation values computed by our method using all the reviews of 10 movies for t_0 varied from 0.05 to 0.95 with a step of 0.05 48

2.13 Reputation Mean Absolute Error for the three approaches 49

3.1 Reputation system pipeline 57

3.2 IMDb user reviews structure 60

3.3 Number of samples in the SST-5 training and test set 61

3.4 Confusion matrix of the Multinomial Naive Bayes (unigrams) for the SST-5 test set 63

3.5 Reputation visualization 64

3.6 IMDb Weighted Average Ratings 65

3.7 Squared error comparison results: dataset 1 to dataset 10 66

3.8 Squared error comparison results: dataset 11 to dataset 20 67

4.1 Proposed system pipeline 72

4.2 IMDb and Amazon reviews structure. 74

4.3 Accuracy of BERT-base model in predicting the sentiment orientation of the collected reviews. 83

4.4 Reputation visualization. 84

4.5 Users and experts average satisfaction score for each reputation system. 90

5.1 AmazonRep pipeline. 95

5.2 Amazon's review structure. 97

5.3 Plot of the sigmoid function. 99

5.4 Reputation visualization. 104

List of Tables

1.1 State-of-the-art results for Sentiment Analysis on SST-5 Fine-grained classification . 26

2.1 Latent Semantic Analysis model weakness in extracting opinions with a similar or same perspective 29

2.2 Symbol denotation 31

2.3 Statistical information of Datasets 37

2.4 Datasets details 38

2.5 Precision, Recall, F-score, and accuracy of Naïve Bayes, SVM and Algorithm 1 for all reviews 39

2.6 Opinion fusion and grouping example results. 40

2.7 Experiment results summarization 45

2.8 $MAER_{t_0}$ between IMDb weighted average votes and the reputation values computed by our method using all the reviews of the 10 movies for t_0 value varied from 0.05 to 0.95 with a step of 0.05 47

3.1 Symbol denotation 55

3.2 Statistical information of the collected datasets 60

3.3 Sentiment analysis classification results 62

3.4 Training time of Bidirectional Gated Recurrent Unit (Bi-GRU), Bidirectional Long Short-Term Memory (Bi-LSTM), Recurrent Neural Network (RNN) and Multinomial Naïve Bayes (MNB) on the SST-5 dataset 63
3.5 Comparison results: reputation visualization 64
3.6 Comparison results: exploited features 65

4.1 Symbol denotation 71
4.2 Example results of review score 79
4.3 Statistical information of dataset 81
4.4 BERT-Base model classification result on Large Movie Review Dataset v1.0 82
4.5 Comparison results on Large Movie Review Dataset v1.0 83
4.6 Comparison results: reputation visualization 85
4.7 Comparison results: review attributes exploited by recent reputation systems 86
4.8 User satisfaction comparison 87
4.9 Expert satisfaction comparison 88
4.10 Experts reviews 89

5.1 Symbol denotation 94
5.2 Example results of review score 100
5.3 Bi-GRU model classification result on Large Movie Review Dataset v1.0 102
5.4 Comparison results: reputation visualization 102
5.5 Comparison results: review attributes exploited by recent reputation systems 102

Acknowledgments

I would like to thank my esteemed supervisor Dr. El Habib Nfaoui for his invaluable supervision, support, and tutelage during the course of my PhD degree. I would also like to thank my friend Mohammed El Moutaouakkil for helping me during the writing and proofreading of this thesis. My appreciation also goes out to my family and friends for their encouragement and support all through my studies.

Abstract

Customer reviews represent a very rich data source from which we can extract very valuable information about different online shopping experiences. The amount of the collected data may be very large especially for trendy items (products, movies, TV shows, hotels, services ...), where the number of available customers' opinions could easily surpass thousands. In fact, while a good number of reviews could indeed give a hint about the quality of an item, a potential customer may not have time or effort to read all reviews for the purpose of making an informed decision (buying, renting, booking ...). Thus, the need for the right tools and technologies to help in such a task becomes a necessity for the buyer as for the seller. My research goal in this thesis is to develop reputation systems that can automatically provide E-commerce customers with valuable information to support them during their online decision-making process by mining online reviews expressed in natural language.

The first chapter describes and examines previous research work done in the area of natural language processing (NLP) techniques for decision making in E-commerce, document-level sentiment analysis, and fine-grained sentiment analysis. The chapter also covers the necessary background for understanding Bidirectional Encoder Representations from Transformers (BERT) model since we employed it to determine the sentiment orientation of customer and user reviews.

Chapter 2 presents a reputation system that incorporates sentiment analysis, semantic analysis, and opinion fusion to generate accurate reputation values toward online items.

The next chapter describes MTVRep, a movie and TV show reputation system that exploits fine-grained sentiment analysis and semantic analysis for the purpose of generating and visualizing reputation toward movies and TV shows.

Chapter 4 presents a reputation system that incorporates four review attributes: review helpfulness, review time, review sentiment polarity, and review rating in order to generate reputation toward various online items (products, movies, TV shows, hotels, restaurants, services).
The system also provides a comprehensive reputation visualization form to help potential customers make an informed decision by depicting the numerical reputation value, opinion group categories, and top-k positive reviews as top-k negative reviews.

Chapter 5 describes AmazonRep, a reputation system that extends the system proposed in chapter 4 by exploiting review rating, review helpfulness votes, review time, review sentiment

orientation, and user credibility to support Amazon's customer decision making process.

Key words:

Reputation Generation, Reputation Visualization, E-commerce Decision Making, Opinion Mining, Natural Language Processing (NLP), Bidirectional Encoder Representations from Transformers (BERT), Embeddings from Language Models (ELMo), Machine Learning, Deep Learning, Semantic Analysis.

Abstract français

Un avis client ou avis consommateur est un élément apporté par un client ou consommateur pour évaluer sa satisfaction vis-à-vis d'une expérience, d'un produit ou d'un service. Aujourd'hui 88% des consommateurs consultent les avis clients en ligne avant un achat et 90% déclarent que ces avis ont influencé leur décision d'achat. Les avis clients représentent donc un canal incontournable pour accéder à des retours d'expérience client, des suggestions et idées de consommateurs. Cela permet de collecter des insights précieux et très diversifiés sur les produits et services qui permettent au client de se forger une première impression. Or, le nombre d'avis clients pourrait facilement dépasser les milliers envers les produits tendances peu importe leur catégorie (électroménager, films, séries TV, hôtels, services ...). Ainsi, bien qu'un nombre élevé d'avis pourrait mieux informer sur la qualité d'un article, le client de son côté peut se trouver perdu entre une multitude d'opinions sur le produit désiré. D'où émerge un nouveau défi à relever: accompagner la clientèle au cours de son processus de prise de décision en ligne.

Pour remédier à cette problématique, nous proposons à travers ce travail inscrit dans le cadre de thèse de doctorat, de développer des systèmes en faveur des clients et qui, à partir des commentaires exprimés en langage naturel, peuvent générer des réputations capables d'épargner temps et effort, et donc assurer le support digital nécessaire durant la phase de prise de décision en ligne facilitant ainsi les expériences d'achat en ligne.

Le premier chapitre introduit une étude bibliographique (related work) démontrant les techniques préexistantes relatives au domaine du traitement automatique du langage naturel appliquées pour assister le consommateur durant le processus de décision d'achat dans le social E-Commerce, l'analyse des sentiments au niveau du document (document-level sentiment analysis) ainsi que l'analyse fine des sentiments (fine-grained sentiment analysis).

Le deuxième chapitre présente un système de réputation qui combine l'analyse des sentiments (sentiment analysis), l'analyse sémantique (semantic analysis) ainsi que la fusion des opinions (opinion fusion) afin de générer une valeur de réputation envers différentes entités (produits, films, hôtels, services).

Le troisième chapitre introduit MTVRep, un système de réputation dédié aux films et aux séries TV. Le système proposé combine l'analyse sémantique (semantic analysis) avec l'analyse fine des sentiments (fine-grained sentiment analysis) pour la génération et la visualisation de la réputation.

Le quatrième chapitre présente un système de réputation qui prend en considération l'utilité du commentaire (review helpfulness), la date de publication du commentaire (review time), le sentiment exprimé dans le commentaire (review sentiment polarity) ainsi que le nombre d'étoiles associées au commentaire (review rating) dans le but de générer la réputation envers différentes entités (produits, films, séries TV, hôtels, services).

Le dernier chapitre décrit AmazonRep, une extension du système de réputation présenté en chapitre 4 qui considère l'utilité du commentaire (review helpfulness), la date de publication du commentaire (review time), le sentiment exprimé dans le commentaire (review sentiment polarity), le nombre d'étoiles associées au commentaire (review rating) ainsi que la crédibilité du commentateur (user credibility) afin d'assister les clients d'Amazon durant le processus: **"prise de décision d'achat"**.

Mots Clés:
Génération de la réputation, Visualisation de la réputation, Processus de prise de décision dans le social E-Commerce, Fouille d'opinions, Traitement automatique des langues, BERT, ELMo, Apprentissage automatique, Apprentissage profond, Analyse sémantique.

Introduction

Overview

Nowadays, the amount of unstructured text data that online users produce on E-commerce websites has grown dramatically. The International Data Corporation[1] (IDC) estimates that unstructured data already accounts for a staggering ninety percent of all digital data. The shocking fact is that in most cases, 90% of that data is never analyzed! Therefore, the need to automatically process it and extract different types of knowledge from it becomes a necessity. My research goal in this thesis is to develop reputation systems that can automatically provide E-commerce customers with valuable information to support them during their online decision-making process (buying, booking, renting ...) by mining online reviews expressed in natural language.

Previous studies on reputation generation of online products have primarily focused on exploiting user ratings [4–9] to generate a score toward online products. Surprisingly, textual data (reviews), which constitute a significant part of customer information, have been somewhat ignored until 2012 when Abdel-Hafez et al. [10] designed a product reputation model that uses text reviews rather than users' ratings for the purpose of generating a more realistic reputation value for every feature of the product as for the product itself. This was achieved by incorporating opinion orientation and opinion strength (sentiment analysis), though, there was no evidence to support the efficiency of the proposed reputation system since the authors **assumed** that the product features and the opinion orientation and strength to the features in each product review have been determined by using existing opinion mining techniques without actually applying them. In 2017, Yan et al. [3]

[1] `https://www.idc.com/`

proposed a reputation system that exploits textual reviews and their ratings to generate numerical reputation values toward Amazon's products. The system fuses and groups reviews into several principal opinion sets based on review relevance (Latent semantic analysis and cosine similarity measure), citations, votes, and ratings, and then aggregates the fused and grouped opinions to generate a reputation value by considering the popularity and other statistics of principal opinion sets, though, previous studies on reputation generation and visualization have two shortcomings.

1. **Exploited features**: Besides numerical rating and textual content, online reviews contain other useful information that could be exploited for reputation generation and visualization, such as review helpfulness, which implies that reviews that receive higher votes from other users typically provide more valuable information, review time, which means that more recent reviews generally provide users with more up-to-date information, and user credibility, which implies that reviews written by trusted users have more impact on the popularity of the target item (product, movie, hotel, service ...). Nevertheless, previous studies on reputation generation have disregarded incorporating these features during the reputation generation phase.

2. **Reputation visualization**: It is very important to provide potential customers with valuable information toward the target entity (product, movie, hotel, service ...) in order to support them during the decision-making process in E-Commerce (buying, renting, booking ...). However, previous studies on reputation generation have provided customers with insufficient information during the reputation visualization phase, for instance, the numerical reputation value toward the target entity and the opinion categories, neglecting other helpful information such as the top-k positive reviews, the top-k negative reviews, the distribution of sentiment over reviews ...

The reputation systems presented in this thesis address these two shortcomings. They incorporate various features (review time, review helpfulness, numerical rating, user credibility, review sentiment orientation) and various opinion mining techniques (binary sentiment analysis, fine-grained sentiment analysis, semantic analysis) during the reputation generation phase. Furthermore, they provide customers with a comprehensive form of reputation visualization that contains valuable information toward the target entity (product, movie, hotel, service ...) including the numerical reputation value, the distribution of sentiment over reviews, the top-k positive reviews, and the

top-k negative reviews.

Contributions and Outline of This Thesis

This thesis is structured as follows:

Summary Chapter 1: Background and Literature Review

The first chapter describes and examines previous research work done in the area of natural language processing (NLP) techniques for decision making in E-commerce, document-level sentiment analysis, and fine-grained sentiment analysis. Besides, the chapter covers the necessary background for understanding BERT model [2] because we employed it to determine the polarity of customer and user reviews.

Summary Chapter 2: Sentiment Analysis and Opinion Fusion for Reputation Generation

In the second chapter, we have an interest in improving Yan et al. [3] approach by generating accurate reputation values toward online entities (movies, products, hotels, services . . .). The fact that the majority of reviews hold a positive or negative sentiment (point of view) toward a target entity led us to the idea of using a classification step in order to separate positive and negative reviews before grouping them into different sets based on semantic relations, then computing reputation toward the target entity using Weighted Arithmetic Mean.

Summary Chapter 3: Fine-grained Opinion Mining and Semantic Analysis for Reputation Generation

In the third chapter, we propose MTVRep, a movie and TV show reputation system that applies fine-grained opinion mining to separate reviews into five opinion groups: strongly negative, weakly negative, neutral, weakly positive, and strongly positive. Then, it computes a custom score for each group based on acquired statistics. These statistics include the number of reviews, the sum of their

ratings, and the sum of their semantic similarity (ELMo and cosine metric). Finally, a numerical reputation value is produced toward the target movie or TV show using the weighted arithmetic mean.

Summary Chapter 4: Aggregating Customer Review Attributes for Online Reputation Generation

In the fourth chapter, we propose a reputation system that generates reputation toward various items (products, movies, TV shows, hotels, restaurants, services) by mining customer and user reviews expressed in natural language. The system incorporates four review attributes: review helpfulness, review time, review sentiment polarity, and review rating. The system also provides a comprehensive reputation visualization form by depicting the numerical reputation value, opinion group categories, top-k positive reviews, and top-k negative reviews.

Summary Chapter 5: Reputation Generation and Visualization for Amazon's Products

In the fifth chapter, we propose AmazonRep, a reputation system that extends the system proposed in chapter 4 by exploiting review rating, review helpfulness votes, review time, review sentiment orientation, and user credibility for the purpose of supporting Amazon's customer decision making process.

Chapter 1

Background and Literature Review

1.1 Introduction

This chapter describes and examines previous research work done in the area of natural language processing (NLP) techniques for decision making in E-commerce, document-level sentiment analysis, and fine-grained sentiment analysis. Additionally, the chapter covers the necessary background for understanding BERT model [2] since we employed it to determine the sentiment orientation of customer and user reviews.

1.2 NLP techniques for online decision making in E-commerce

The BusinessDictionary[1] defines decision making as: *"The thought process of selecting a logical choice from the available options"*. During the last twenty years, few systems have been proposed to help potential customers in making decisions in E-commerce websites using mainly two approaches: feature-based summarization of customer reviews and reputation generation.

1.2.1 Feature-based opinion summarization

Feature-based opinion summarization is :*"one of the opinion summarization techniques which provide brief yet most important information containing summary about different aspects related to the*

[1] `http://www.businessdictionary.com/definition/decision-making.html`

target product. Since it focuses on different features instead of giving the general details about a product, it has become more significant and demanded form of summarization. This technique is also known as Aspect-based Opinion Summarization. It is actually a way of generating summaries for a set of aspects or features of a specific product" [11]. Hu & Liu (2004) [12] were the first to design and build a system that produces a feature-based summary from customer reviews. The proposed system performs three tasks: (1) association rule mining [13] is used to extract product features from customer reviews, (2) WordNet [14] is utilized to predict the semantic orientations of opinion words, (3) a structured feature-based summary is produced. Over the last two decades, few systems have been proposed to perform feature-based summarization. The summarizers are applied on various domains: product reviews [12, 15–18], movie reviews [19, 20], local services reviews [21] and hotel reviews [22, 23], etc. Figure 1.1 depicts a sample of feature-based summary proposed by Kangale et al. [1].

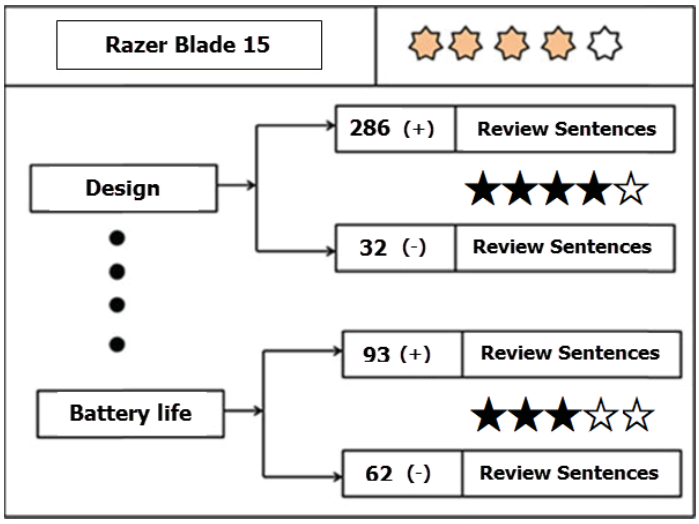


Figure 1.1: Kangale et al. [1] feature-based summary

1.2.2 Reputation generation

Backing to reputation generation. The Cambridge Dictionary [2] defines reputation as *"the opinion that people, in general, have about someone or something, or how much respect or admiration someone or something receives, based on past behavior or character"*. The pioneer work that tackles the task of reputation generation based on mining opinions expressed in natural languages was

[2] https://dictionary.cambridge.org/fr/dictionnaire/anglais/reputation

firstly proposed by Yan et al. (2017) [3] in which reviews are fused into several opinion sets based on their semantic relations, then, a single reputation value is generated by aggregating the fused and grouped opinions' statistics (the sum of similarities, the sum of ratings, the number of reviews). In [24], the authors applied K-means clustering algorithm to group similar reviews into the same cluster using Latent Semantic Analysis (LSA) before producing a reputation value using the statistics of each cluster. However, both approaches have relied on extracting semantic relations between reviews and have disregarded the fact that the majority of online customer and user reviews are opinionated. Benlahbib & Nfaoui (2019) [25] proposed a fourfold approach to improve the reputation system proposed in [3]. First, Naïve Bayes and Linear Support Vector Machines classifiers were applied to separate reviews into positives and negatives by predicting their sentiment polarity. Second, positive and negative reviews were fused into different sets based on their semantic similarity (Latent Semantic Analysis and cosine similarity). Third, a custom reputation value is computed separately for both positive opinion sets and negative opinion sets. Finally, a single reputation value is calculated using the weighted arithmetic mean. The same authors designed and built a new reputation system [26] that considers review time, review helpfulness votes, and review sentiment orientation for the purpose of generating reputation toward online entities (products, movies, hotels, and services). The system also provides a new form of reputation visualization by depicting the numerical reputation value, the distribution of sentiment over the reviews, the top-k positive reviews, and the top-k negative reviews. Recently, Benlahbib & Nfaoui [27] incorporated fine-grained opinion mining and semantic analysis to generate and visualize reputation toward movies and TV shows (MTVRep). The authors trained the Multinomial Naïve Bayes classifier on the SST-5 dataset [28] to group reviews into five emotion classes: strongly negative, weakly negative, neutral, weakly positive, and strongly positive, then, Embeddings from Language Models (ELMo) [29] and cosine similarity were applied to extract the semantic similarity between reviews and to compute a custom score for each emotion class. Finally, the weighted arithmetic mean is used to compute the movie or TV show reputation value.

1.3 Document-level Sentiment Analysis

Ahlgren (2016) [30] defines sentiment analysis as *"the process of identifying and detecting subjective information using natural language processing, text analysis, and computational linguistics"*. Gen-

erally, sentiment analysis can be divided into three levels: sentence-level opinion mining, document-level opinion mining, and fine-grained opinion mining.

According to [31], document-level opinion mining is *"a task of extracting the overall sentiment polarities of given documents, such as movie reviews, product reviews, tweets, and blogs"*.

Many approaches have been used to handle the task of document-level sentiment analysis:

- Supervised approaches: These approaches require annotated corpus to train machine learning models. The first work for supervised document-level opinion mining was proposed by Pang et al. (2002) [32]. Three machine learning classifiers (Support Vector Machines (SVMs) [33], Naïve Bayes classifier [34] and Maximum Entropy classifier [35]) were trained with movie reviews labeled by sentiment (positive/negative). The authors trained the three models on various kinds of features (unigrams, bigrams, parts of speech, and position) and found that the sentiment classification task performs well when adopting unigrams as features. Kennedy & Inkpen (2006) [36] trained Support Vector Machine classifiers on unigrams and bigrams by incorporating three types of context valence shifters: *"intensifiers"*, *"negations"* and *"diminishers"*. The trained model achieved an accuracy of 0.859 on movie review data[3] [37]. Koppel & Schler (2006) [38] defined the sentiment classification task as a three-category problem (positive, negative, and neutral) and used different learning algorithms: SVM, J48 Decision Tree [39]. Naïve Bayes, Linear Regression [40], and Frank & Hall (2001) [41] classification method. The results show significant improvement in the sentiment classification accuracy when using neutral examples over ignoring them. In [42], the authors combined Naïve Bayes and Support Vector Machine by training SVM with Naïve Bayes log-count ratios as features. The proposed model has achieved promising results across several datasets. Jing et al. (2015) [43] applied Naïve Bayes algorithm on 3046 customer reviews related to fifty-eight business-to-team (B2T) websites to study the survival conditions of B2T companies. Augustyniak et al. (2016) [44] presented a wide comparison and analysis of opinion mining task for several classifiers: Random Forests [45], Linear SVC, Bernoulli Naïve Bayes, Mulinomial Naïve Bayes, Extra Tree Classifier [46], Logistic Regression, and AdaBoost [47, 48]. They conducted experiments on Amazon review data [49] and found that the Logistic Regression classifier outperforms the other classifiers in predicting sentiment polarity of product reviews.

[3] `http://www.cs.cornell.edu/people/pabo/movie-review-data/`

- Unsupervised approaches: They attempt to determine the sentiment orientation of a text by applying a set of rules and heuristics obtained from language knowledge. Turney (2002) [50] was the first to propose an unsupervised sentiment analysis technique to classify reviews as *"recommended"* or *"not recommended"*. The semantic orientation of a phrase is computed as the pointwise mutual information (PMI) [51] between the given phrase and the word *"excellent"* minus the pointwise mutual information between the given phrase and the word *"poor"*. The proposed algorithm achieved an accuracy of 84% for automobile reviews, 80% for bank reviews, 71% for travel destination reviews, and 66% for movie reviews. In [52], the authors proposed a lexicon-based method to mine text by using a dictionary of sentiment words and their semantic orientations varied between -5 and +5. The authors also incorporated amplifiers, downtoners, and negation words to compute a sentiment score for each document. Vashishtha & Susan (2020) [53] proposed a fuzzy rule-based approach to perform opinion mining of tweet. The authors use a novel unsupervised nine fuzzy rule-based system to predict the sentiment orientation of the post (positive, negative, or neutral). In [54], Fernández-Gavilanes et al. (2016) proposed a sentiment analysis approach to predict the polarity in online textual messages such as tweets and reviews using an unsupervised dependency parsing-based text classification method.

- Deep learning approaches: Over the past few years, deep learning models have greatly improved the state-of-the-art of opinion mining. Moraes et al. (2013) [55] made a comparative study between Support Vector Machines (SVM) and Artificial Neural networks (ANN) for document-level opinion mining and found that ANN results are at least comparable or superior to SVMs. In [56], the authors proposed an unsupervised algorithm named paragraph vector (doc2vec), an extension to word2vec approach [57] in order to overcome the weakness of the bag-of-words (BoW) model. The proposed algorithm learns vector representations for variable-length texts such as sentences, paragraphs, and documents. Experimental results depict that doc2vec algorithm achieved new state-of-the-art results on several sentiment analysis tasks. Johnson & Zhang (2015) [58] trained a parallel Convolutional Neural Network (CNN) [59] without using pre-trained word vectors: word2vec, doc2vec, and GloVe[4] [60]. Instead, convolutions are directly applied to one-hot encoding vectors to leave the network solely with information about the word order. The proposed approach achieved an accuracy rate

[4] `https://nlp.stanford.edu/projects/glove/`

of 92.33% on Large Movie Review Dataset[5] outperforming both SVM [61] and NB-LM [62]. Baktha & Tripathy (2017) [63] investigated the performance of Long Short-Term Memory (LSTM) [64], vanilla RNNs, and Gated Recurrent Units (GRU) [65] on the Amazon health product reviews dataset and sentiment analysis benchmark datasets SST-1 and SST-2. The results depict that GRU achieved the highest sentiment classification accuracy. In [66], the authors combined unsupervised data augmentation (UDA) with Bidirectional Encoder Representations from Transformers (BERT) [2] and compared it with fully supervised $\boldsymbol{BERT_{LARGE}}$ on six text classification benchmark datasets. The authors reported that their proposed approach outperforms $\boldsymbol{BERT_{LARGE}}$ on five text classification benchmark datasets including Large Movie Review Dataset. Facebook AI and University of Washington researchers [67] improved BERT by proposing *"Robustly Optimized BERT approach"* (RoBERTa) that was trained with more data and more number of pretraining steps and has dropped the next sentence prediction (NSP) approach used in BERT. Recently, Google researchers [68] proposed XLNet, a *"generalized autoregressive pretraining method"* that outperforms Bidirectional Encoder Representations from Transformers (BERT) on twenty text classification tasks and achieves state-of-the-art results on eighteen text classification tasks including sentiment analysis. At the beginning of 2020, Lan et al. (2019) [69] proposed ALBERT: *"A Lite BERT for Self-supervised Learning of Language Representations"*. The paper describes parameter reduction techniques to lower memory reduction and increase the training speed and accuracy of BERT models. In [70], the authors introduced a novel *"Text-to-Text Transfer Transformer"* (T5) neural network model pre-trained on a large text corpus that can convert any language problem into a text-to-text format. The T5 model achieved state-of-the-art results on the SST-2 binary classification dataset with an accuracy of 97.4%. Recently, Clark et al. (2020) [71] presented *ELECTRA*, a method for self-supervised language representation learning that uses a new pre-training task called replaced token detection (RTD). The experiment results showed that RTD is more efficient than masked language modeling (MLM) pre-training models such as BERT.

[5] `https://ai.stanford.edu/~amaas/data/sentiment/`

1.4 Fine-grained Sentiment Analysis on the 5-class Stanford Sentiment Treebank (SST-5) dataset

Differently from binary sentiment analysis that aims to determine whether a given text holds a positive or negative polarity, fine-grained sentiment analysis is a significantly more challenging task in which the polarity of a given text is classified into five discrete classes: strongly negative, weakly negative, neutral, weakly positive, and strongly positive.

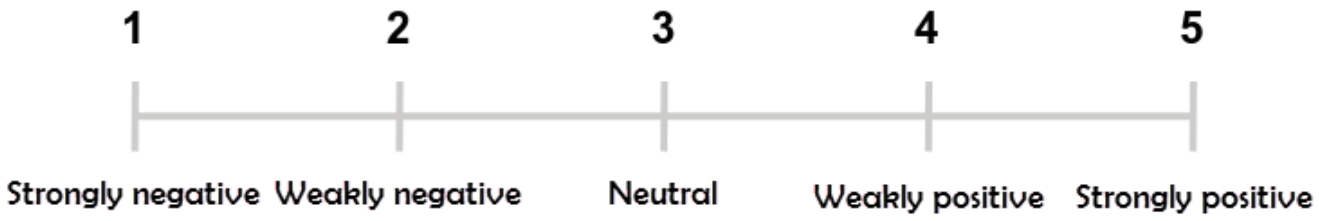


Figure 1.2: Typical class labels (or intensities) for fine-grained sentiment classification

Many approaches have been proposed to handle the task of fine-grained sentiment analysis for the 5-class Stanford Sentiment Treebank (SST-5) dataset. Xu et al. [72] proposed Emo2Vec which are word-level representations that encode emotional semantics into fixed-sized, real-valued vectors. Mu et al. [73] presented a simple post-processing operation that renders word representations even stronger by eliminating the top principal components of all words. Socher et al. [74] introduced Recursive Neural Tensor Networks and the Stanford Sentiment Treebank. Wang et al. [75] proposed RNN-Capsule, a capsule model based on Recurrent Neural Network (RNN) for sentiment analysis. Yang [76] presented RNFs, a new class of convolution filters based on recurrent neural networks. McCann et al. [77] introduced an approach for transferring knowledge from an encoder pre-trained on machine translation to a variety of downstream natural language processing (NLP) tasks. Munikar et al. [78] used the pre-trained BERT [2] model and fine-tuned it for the fine-grained sentiment classification task on the SST-5 dataset.

Table 1.1 summarizes the latest works on fine-grained opinion mining applied to the Stanford Sentiment Treebank dataset (SST-5).

Method	Authors & Year	Accuracy %
RoBERTa-large+Self-Explaining	Sun et al. (2020) [79]	59.1
BCN+Suffix BiLSTM-Tied+CoVe	Brahma (2018) [80]	56.2
BERT large	Munikar et al. (2019) [78]	55.5
BCN+ELMo	Peters et al. (2018) [29]	54.7
BCN+Char+CoVe	McCann et al. (2017) [77]	53.7
CNN-RNF-LSTM	Yang (2018) [76]	53.4
RNN-Capsule	Wang et al. (2018) [75]	49.3
SWEM-concat	Shen et al. (2018) [81]	46.1
RNTN	Socher et al. (2013) [74]	45.7
GRU-RNN-WORD2VEC	Mu et al. (2017) [73]	45.02
GloVe+Emo2Vec	Xu et al. (2018) [72]	43.6
Emo2Vec	Xu et al. (2018) [72]	41.6

Table 1.1: State-of-the-art results for Sentiment Analysis on SST-5 Fine-grained classification

1.5 Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT) [2] is a pre-trained model that aims to pre-train deep bidirectional representations from unlabeled text (Figure 1.3).

BERT was pre-trained on two unsupervised tasks (Figure 1.4):

- **Masked LM (MLM)**: In order to train a deep bidirectional representation, BERT masks some percentage of the input tokens at random, and then predicts those masked tokens.

- **Next Sentence Prediction (NSP))**: In order to train a model that understands sentence relationships, BERT predicts if a sentence B is the actual next sentence that follows a sentence A or not.

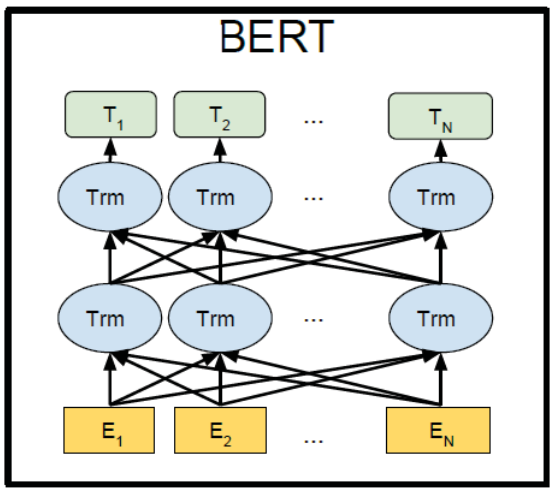


Figure 1.3: Bidirectional Encoder Representations from Transformers (BERT) pre-training architecture [2]

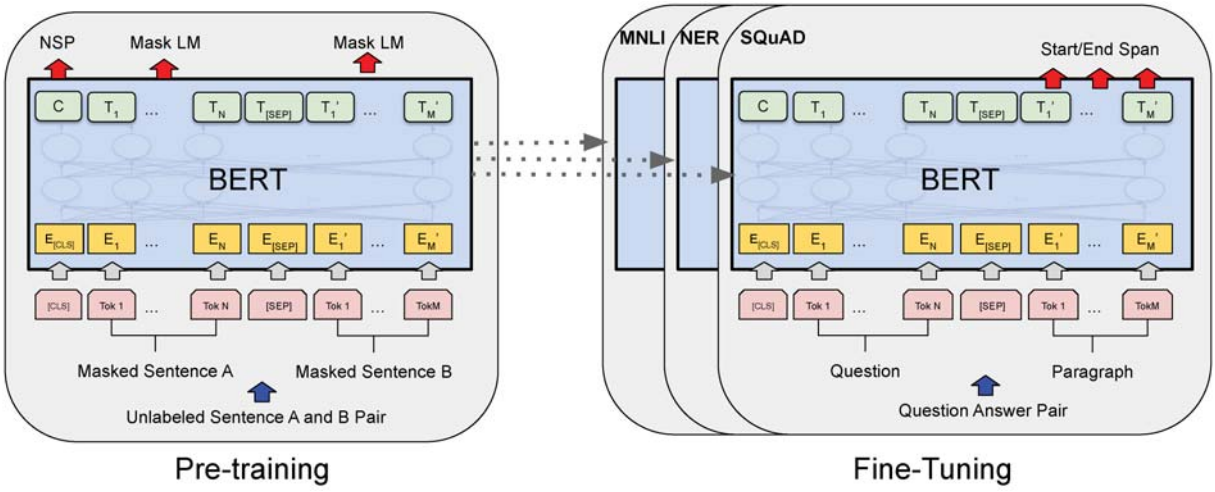


Figure 1.4: Overall pre-training and fine-tuning procedures for BERT [2]

1.6 Conclusion

In this chapter, we described and examined previous research work done in the area of natural language processing (NLP) techniques for decision making in E-commerce, document-level sentiment analysis, and fine-grained sentiment analysis. We also provided the necessary background for understanding BERT model as we employed it to determine the polarity of customer and user reviews.

The next chapter will present a reputation system that incorporates sentiment analysis, semantic analysis, and opinion fusion to generate accurate reputation values toward online items.

Chapter 2

Sentiment Analysis and Opinion Fusion for Reputation Generation

2.1 Introduction

Over the 21 century, the web has been expanding at an incredible rate. Thanks to social media and web communities, people express freely their opinions toward any entity they desire, with no chains or restrictions. Those opinions represent a valuable source of information that can contribute to track public perspectives and to generate reputation about a target entity for the reason that they carry the preferences and attitudes of humans. We can find many research papers and books for reputation systems [82], [83] and [84]. However, there is particularly a lack of studies that handle the task of generating reputation based on mining opinions expressed in natural languages. Hence the importance of Yan et al. [3] pioneering work that presents an approach to generate reputation toward Amazon products by fusing and mining opinions expressed in natural languages.

Yan et al. [3] applied Latent Semantic Analysis (LSA) model and cosine similarity metric to group reviews into several principal opinion sets. The authors claimed that the opinions in each set hold a similar or same perspective. However, we believe that relying only on Latent Semantic Analysis model is not enough for extracting opinions with a similar perspective. In order to justify our point of view, we study the three reviews in Table 2.1.

Table 2.1 represents the results after applying LSA model, then computing document similarity using LSA components and cosine similarity metric. We notice that the similarity value between

Table 2.1: Latent Semantic Analysis model weakness in extracting opinions with a similar or same perspective

	The movie was good	The movie was not good at all	Amazing acting
The movie was good	1.000000	0.996730	0.221941
The movie was not good at all	0.996730	1.000000	0.142422
Amazing acting	0.221941	0.142422	1.000000

the first and the second review is very high, on the contrary, the similarity between the first and the third review is low, therefore, based on the opinion fusion and grouping algorithm proposed in [3], and by choosing an opinion fusion threshold $t_0 \geq 0.23$, we will get two opinion sets, the first set contains the first and the second reviews and the second set contains the third review, yet, it is obvious that the first and the third reviews stand for a positive point of view and the second review stands for a negative point of view, hence, we believe that adding a polarity classification step is an effective idea in order to classify reviews into positives and negatives (sentiment analysis) before grouping them based on their semantic relations. As a matter of fact, applying a sentiment classification step before the grouping phase will overcome the weakness of LSA model in extracting reviews with the same perspective and will guarantee that positive reviews will be grouped together and so on.

In this chapter, we have an interest in improving Yan et al. [3] approach by generating accurate reputation values toward various online entities (movies, products, hotels, services ...). The fact that the majority of reviews hold a positive or negative sentiment (point of view) toward a target entity led us to the idea of using a sentiment classification step to separate positive and negative reviews before grouping them into different sets based on semantic relations, then computing reputation toward the target entity using Weighted Arithmetic Mean. Indeed, the main steps of this contribution can be summarized as follows:

1. We apply a polarity classification step based on the two classifiers Naïve Bayes and Linear Support Vector Machine (LSVM) in order to determine positive and negative reviews.

2. We group **separately** positive and negative reviews into different opinion sets based on semantic relations.

3. We calculate a custom reputation value **separately** for positive and negative groups by considering some statistics of principal opinion sets.

4. We compute the final reputation value toward the target entity using Weighted Arithmetic Mean.

2.2 Problem definition

This section covers the necessary background for understanding the remainder of this chapter, including the problem definition.

In this chapter, we face the problem of generating reputation for movies by combining opinion fusion and opinion mining. Given a set of reviews $R = \{r_1,\ r_2,\ \ldots,\ r_n\}$ expressed for an entity A, the set of their attached ratings $V = \{v_1,\ v_2,\ \ldots,\ v_n\}$ where $v_i \in [1,5]$ or $v_i \in [1,10]$ depending on the rating system, the set of their sentiment orientation predicted by Naïve Bayes model $nbp_{=}\{nbp(r_1),\ nbp(r_2),\ \ldots,\ nbp(r_n)\}$, the set of their sentiment orientation predicted by Linear Support Vector Machine model $svmp_{=}\{svmp(r_1),\ svmp(r_2),\ \ldots,\ svmp(r_p)\}$, where $nbp(r_i), svmp(r_i) \in [positive, negative]$. The goal is to separate reviews into positives and negatives before grouping them into principal opinion sets based on their semantic relations (LSA model and cosine similarity metric), then, to compute a custom reputation value $Rep(A^{polarity})$ **separately** for positive and negative groups by considering some statistics of principal opinion sets such as $V_k^{polarity}$: the sum of ratings in principal opinion set k for a specific polarity, $S_k^{polarity}$: the sum of semantic similarity in principal opinion set k for a specific polarity, $N_k^{polarity}$: the number of similar opinions in principal opinion set k for a specific polarity and $K^{polarity}$: the number of opinion sets for a specific polarity (positive or negative). Finally, we compute the final reputation value $finalRep(A)$ using the Weighted Arithmetic Mean. Table 2.2 presents the descriptions of notations used in the rest of this chapter.

Table 2.2: Symbol denotation

Symbol	Description
A	The target entity
R	The set of reviews expressed for the entity A
V	The set of ratings expressed for the entity A
$V_k^{polarity}$	The sum of ratings in principal opinion set k for a specific polarity
$S_k^{polarity}$	The sum of semantic similarity in principal opinion set k for a specific polarity
$N_k^{polarity}$	The number of similar opinions in principal opinion set k for a specific polarity
$K^{polarity}$	The number of opinion sets for a specific polarity (positive or negative)
$O^{polarity}$	The number of (positive/negative) opinions
$O^{positive}$	The number of positive opinions
$O^{negative}$	The number of negative opinions
$A^{polarity}$	Groups of (positive/negative) opinions expressed for the entity A
$A^{positive}$	Groups of positive opinions expressed for the entity A
$A^{negative}$	Groups of negative opinions expressed for the entity A
nbp	The set of sentiment orientation (positive or negative) predicted by Naïve Bayes model for reviews expressed toward the entity A
$svmp$	The set of sentiment orientation (positive or negative) predicted by Linear Support Vector Machine model for reviews expressed toward the entity A
$Rep(A^{polarity})$	The custom reputation value for groups of (positive/negative) opinions.
$Rep(A^{positive})$	The custom reputation value for groups of positive opinions.
$Rep(A^{negative})$	The custom reputation value for groups of negative opinions.
$finalRep(A)$	The final reputation value toward entity A

2.3 Proposed approach

2.3.1 System overview

Figure 2.1 describes the pipeline of our work.

1. **Polarity classification phase.** We separate reviews into positives and negatives based on their sentiment polarity and their attached ratings by combining two classifiers: Naïve

Bayes and Linear Support Vector Machine (LSVM). Both models are trained with 2000 movie reviews (1000 positive and 1000 negative processed reviews).

2. **Fusion and grouping phase.** We apply the opinion fusion and grouping algorithm proposed in [3] ***separately*** for positive and negative reviews in order to group them into principal opinion sets based on their semantic relations (LSA model and cosine similarity metric).

3. **Custom reputation generation phase.** We calculate a custom reputation value ***separately*** for positive and negative groups by considering some statistics of principal opinion sets.

4. **Final reputation generation phase.** We compute the final reputation value toward the target entity based on the custom reputation value for positive and negative groups using the Weighted Arithmetic Mean.

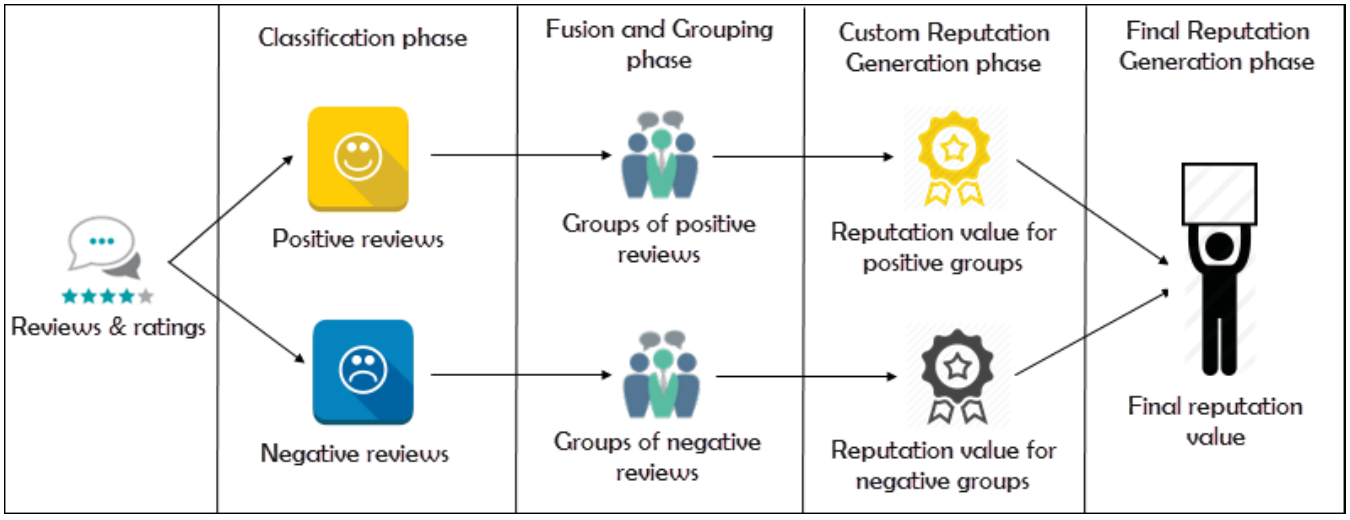


Figure 2.1: Pipeline of our work

2.3.2 Polarity classification phase

We propose a combination of Naïve Bayes and LSVM classifiers to determine the polarity of the collected reviews (positives or negatives). The classification step is organized as follows:

1. First, we apply Naïve Bayes classifier in order to determine the sentiment orientation of the collected reviews, then we consider both the predictions of Naïve Bayes (positive or negative) and their ratings (numeric scale from 1 to 5 or 1 to 10) in the following way:

   (a) If the review polarity is predicted as positive and its attached rating is greater than 5, then, the review will be added to the *"positiveReviews"* set.

   (b) If the review polarity is predicted as negative and its attached rating is less or equal to 5, then, the review will be added to the *"negativeReviews"* set.

   (c) If the review polarity is predicted as positive and its attached rating is less or equal to 5, then, the review will be added to the *"remainingReviews"* set.

   (d) If the review polarity is predicted as negative and its attached rating is greater than 5, then, the review will be added to the *"remainingReviews"* set.

2. Second, we apply SVM classifier with a linear kernel to the reviews contained in *"remainingReviews"* set and we compare SVM polarity prediction with Naïve Bayes polarity prediction:

   (a) If the review polarity is predicted as positive by both SVM and Naïve Bayes classifiers, then, the review will be added to the *"positiveReviews"* set.

   (b) If the review polarity is predicted as negative by both SVM and Naïve Bayes classifiers, then, the review will be added to the *"negativeReviews"* set.

   (c) If the review polarity is predicted as positive (respectively negative) by SVM and negative (respectively positive) by Naïve Bayes classifier, then, the review will be added to the *"remaining1Reviews"* set.

3. Third and finally, the polarity of reviews contained in *"remaining1Reviews"* set will be determined based on their attached ratings:

   (a) If the review attached rating is greater than 5, then, the review will be added to the *"positiveReviews"* set.

   (b) If the review attached rating is less or equal to 5, then, the review will be added to the *"negativeReviews"* set.

Shawe-Taylor and Sun [85] presented Naïve Bayes, Support Vector Machine (SVM) and maximum entropy as the most frequently used models for opinion mining. Furthermore, Pang, Lee, and

Vaithyanathan [32] were the first to use these 3 models to classify **movie reviews** as positive or negative. The authors reported that the use of unigrams as features improves the sentiment classification accuracy. Thus, we trained Naïve Bayes and SVM with unigrams (count) for the purpose of predicting reviews polarity.

We apply Naïve Bayes before SVM because, firstly, we found that Naïve Bayes achieves 82% accuracy in predicting sentiment polarity of the collected reviews compared to 77% accuracy for SVM (Table 2.5), and, secondly, Pang, Lee, and Vaithyanathan [32] applied Naïve Bayes and SVM to a movie review dataset (polarity dataset v2.0) [1] and found that Naïve Bayes achieves higher accuracy in predicting movie reviews polarity compared to SVM. Therefore, we adopt the accurate classifier: Naïve Bayes as the primary classifier, then, we apply SVM classifier when there is a mismatch between Naïve Bayes predicted polarity for a review and its attached rating. Algorithm 1 provides details about the sentiment classification step.

2.3.3 Fusion and grouping phase

In order to overcome the weakness of LSA model in extracting opinions with the same perspective, we apply the opinion fusion and grouping algorithm [3] ***separately*** for positive and negative reviews after classifying them based on their sentiment polarity, which guarantees that positive opinions will be grouped together and so on. Yan et al. [3] describe the opinion fusion and grouping algorithm as follows: *"By applying Algorithm 1, we fuse and group opinions into several principal opinion sets. The opinions in each set hold a similar or same perspective"* and *"Once the processing based on Algorithm 1 has been completed, the opinions are grouped into a number of K fused principal opinion sets. Meanwhile, we also get the statistics of the principal opinions, i.e., the number of similar opinions in each set, the sum of their ratings and the sum of their similarity"*.

2.3.4 Reputation generation

Based on the result of the grouping phase, we propose to compute a custom reputation value ***separately*** for positive and negative groups. We tailor the average similarity $S_k^{polarity}/N_k^{polarity}$ with the average rating value $V_k^{polarity}/N_k^{polarity}$ of principal opinion set k for a specific polarity.

[1] `www.cs.cornell.edu/people/pabo/movie-review-data/review_polarity.tar.gz`

We compute a custom reputation value separately for positive and negative groups by applying this formula:

$$Rep(A^{polarity}) = \frac{1}{K^{polarity}} \cdot \sum_{k=1}^{K^{polarity}} \frac{V_k^{polarity} . S_k^{polarity}}{N_k^{polarity} . N_k^{polarity}} \tag{2.1}$$

$K^{polarity}$: The number of opinion sets for a specific polarity (positive or negative opinions).

$N_k^{polarity}$: The number of similar opinions in principal opinion set k for a specific polarity.

$S_k^{polarity}$: The sum of semantic similarity in principal opinion set k for a specific polarity.

$V_k^{polarity}$: The sum of ratings in principal opinion set k for a specific polarity.

In general, the number of positive opinions and the number of negative opinions are not equal, which implies that $O^{positive}$ and $O^{negative}$ don't contribute equally to the final reputation value. Thus, we propose to use the Weighted Arithmetic Mean for computing the final reputation value instead of using the ordinary arithmetic mean:

$$finalRep(A) = \frac{\sum_{polarity}^{[positive,negative]} Rep(A^{polarity}) . O^{polarity}}{\sum_{polarity}^{[positive,negative]} O^{polarity}} \tag{2.2}$$

That leads to :

$$finalRep(A) = \frac{Rep(A^{positive}) . O^{positive} + Rep(A^{negative}) . O^{negative}}{O^{positive} + O^{negative}} \tag{2.3}$$

$O^{positive}$: The number of positive opinions.

$O^{negative}$: The number of negative opinions.

$A^{positive}$: Groups of positive opinions.

$A^{negative}$: Groups of negative opinions.

$Rep(A^{positive})$: Custom reputation value for groups of positive opinions.

$Rep(A^{negative})$: Custom reputation value for groups of negative opinions.

2.4 Experimental results and discussion

2.4.1 Training phase

As we mentioned earlier, we used Naïve Bayes and Linear Support Vector Machine classifiers, both are trained with a movie review dataset (polarity dataset v2.0) [2] that contains 1000 positive and 1000 negative reviews. The sentences in the corpus are processed and downcased.

The chosen features were unigrams (count) for Naïve Bayes classifier and unigrams TF-IDF (Term frequency - Inverse document frequency) weighted word frequency features for SVM.

2.4.2 Datasets collection and preprocess

In order to evaluate the proposed reputation system, we need a dataset that contains both reviews and ratings. However, there is no standard dataset suitable for our evaluation, therefore, we extracted reviews from IMDb[3] (Internet Movie Database) website which represents the world's most popular and authoritative source for movie, TV, and celebrity content. Figure 2.2 represents a sample of the dataset without review polarity

[2] `www.cs.cornell.edu/people/pabo/movie-review-data/review_polarity.tar.gz`

[3] `www.imdb.com`

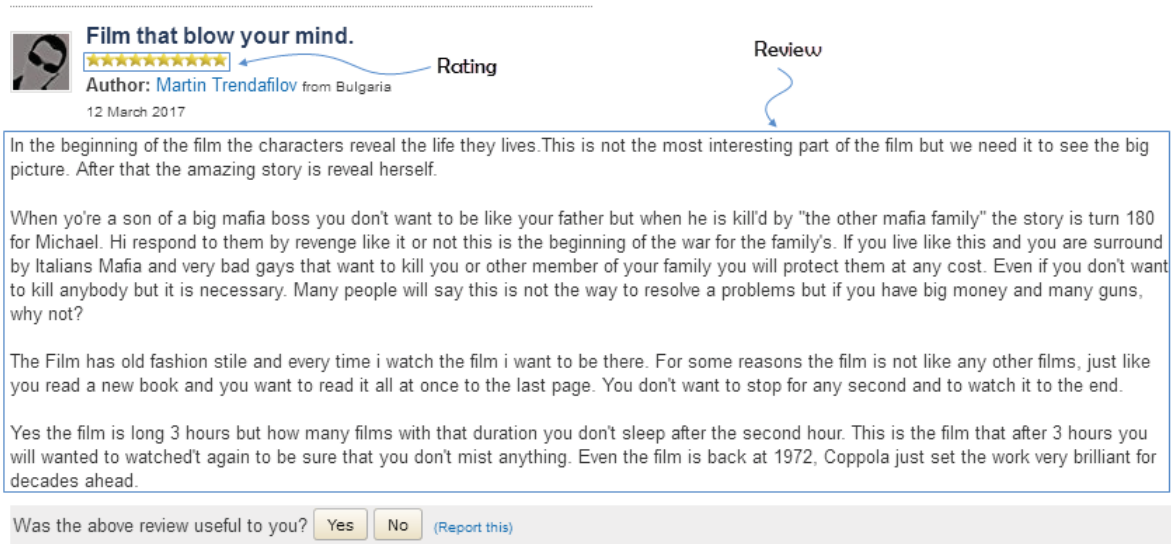


Figure 2.2: Dataset Sample

We manually created 10 datasets for 10 different movies, each one contains 100 reviews [comment + rating + sentiment polarity (manually annotated)] randomly extracted, and we verified that the datasets are representatives based on IMDb users weighted average vote. The statistical information of datasets is shown in Table 2.3.

Table 2.3: Statistical information of Datasets

The total number of movie descriptions	10
The total number of reviews and ratings (i.e., opinions)	1000
The number of reviews per movie	100

You may wonder why collecting 100 reviews for each entity instead of using all the reviews attached to it? Because not every single user who rates the entity (movie) shares its review toward it in the platform (IMDb, Amazon, etc...). Besides, the proposed approach exploits both reviews and ratings, so, if we consider all reviews attached to an entity (movie), we can encounter cases where a movie has a high IMDb weighted average vote, but only users that hold a negative point of view toward it, decide to share their opinions (review + rating), and vice versa. Therefore, those opinions are biased (not representative), which will lead to an unfair reputation value toward the target entity. Thus, we ensure that the ratings are representative during the manual gathering of reviews. Table 2.4 contains more details about datasets.

Table 2.4: Datasets details

Movie	IMDb weighted average vote	Dataset average rating
2012 (Dataset 1)	5.8 (301440 users)	5.8
A Beautiful Mind (Dataset 2)	8.2 (672962 users)	8.2
Amadeus (Dataset 3)	8.3 (295560 users)	8.3
Avatar (Dataset 4)	7.8 (946090 users)	7.8
Clash of the Titans (Dataset 5)	5.8 (240456 users)	5.8
Les Miserables (Dataset 6)	7.6 (260252 users)	7.6
Star Wars Episode I (Dataset 7)	6.5 (579916 users)	6.5
The Expendables (Dataset 8)	6.5 (286373 users)	6.5
The Godfather (Dataset 9)	9.2 (1257206 users)	9.2
The Matrix Revolutions (Dataset 10)	6.7 (384761 users)	6.7

According to Table 2.4, all datasets have an average rating equal to IMDb weighted average vote.

After collecting the reviews, we applied a preprocessing algorithm to remove word segmentation and stop words.

In their work, Yan et al. [3] developed a web spider to collect recent product descriptions, customer reviews, and review ratings from Amazon China (https://www.amazon.cn) and Amazon English (https://www.amazon.com). Each rating is associated with a review and subjectively given by customers. The number of reviews per product is between 80 and 100.

2.4.3 Sentiment classification

After training Naïve Bayes and Linear Support Vector Machine classifiers, we separate reviews into positive and negative based on their sentiment polarity by applying Algorithm 1.

We need to evaluate the effectiveness of Algorithm 1 in classifying reviews. Thus, we compared it to Naïve Bayes and SVM classifiers. Figure 2.3 shows the accuracy of the proposed classification phase compared to Naïve Bayes and SVM for each dataset.

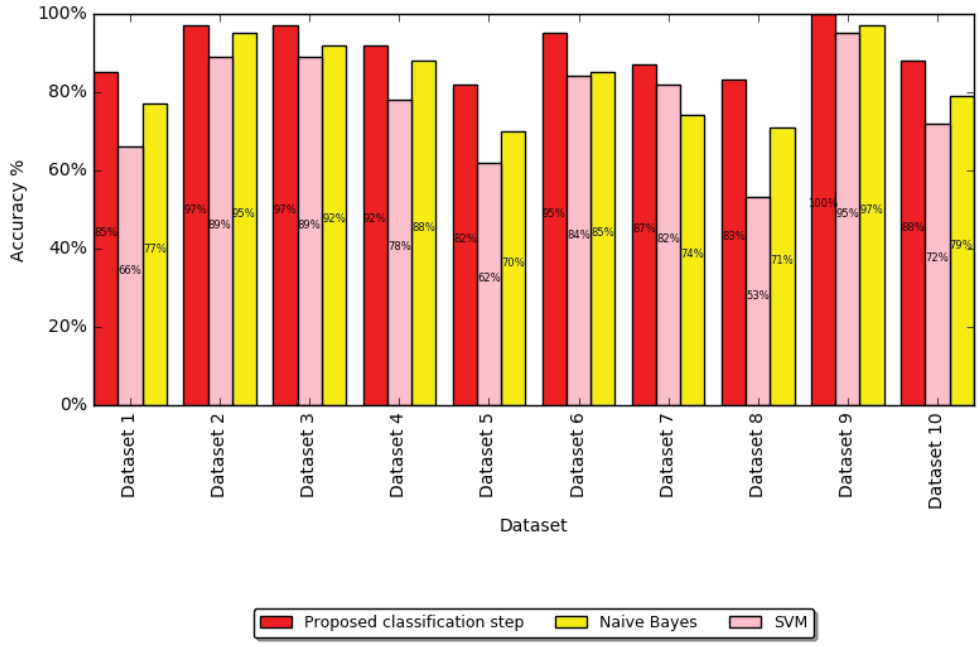


Figure 2.3: Sentiment classification accuracy of Algorithm 1, Naïve Bayes and SVM

We can see from Figure 2.3 that the combination of Naïve Bayes and SVM classifiers (Algorithm 1) gives the highest sentiment classification accuracy for all datasets compared to the separate use of each classifier. Table 2.5 depicts precision, recall, f-score, and accuracy of Naïve Bayes, SVM and Algorithm 1 applied to all 1000 reviews of datasets for sentiment analysis.

Table 2.5: Precision, Recall, F-score, and accuracy of Naïve Bayes, SVM and Algorithm 1 for all reviews

Approach	Polarity	Precision	Recall	F-score	Accuracy
Naive Bayes	Positive	0.88	0.91	0.89	0.82
	Negative	0.54	0.46	0.5	
SVM	Positive	0.94	0.76	0.84	0.77
	Negative	0.43	0.79	0.56	
Algorithm 1	Positive	**0.95**	**0.92**	**0.94**	**0.90**
	Negative	**0.71**	**0.81**	**0.76**	

We can see from Table 2.5 that our proposed classification method (Algorithm 1) outperforms

Naïve Bayes and Linear Support Vector Machine (LSVM) in terms of precision, recall, f-score, and accuracy for both positive and negative reviews.

2.4.4 Opinion fusion and grouping

During this phase, the reviews can be grouped into several opinion sets after classifying them into positive and negative based on their sentiment polarity. At the end of this process, we acquire some statistics such as the sum of semantic similarity $S_k^{polarity}$, the sum of ratings $V_k^{polarity}$ and the number of similar opinions $N_k^{polarity}$ for each set. Table 2.6 below provides example results of the grouping phase.

Table 2.6: Opinion fusion and grouping example results.

Positive Reviews				Negative Reviews			
Set	Sim	Rat	Num	Set	Sim	Rat	Num
R_1	66.86	569	67	R_1	3.99	16	4
R_2	3.99	33	4	R_2	1	3	1
R_3	13.98	113	14				
R_4	7.99	69	8				
R_5	1	8	1				
R_6	1	9	1				

We denote:

Set: the principal review set by fusing and grouping reviews.

Sim: the sum of semantic similarity in a principal opinion set.

Rat: the sum of ratings in a principal opinion set.

Num: the number of similar reviews in a principal opinion set.

From Table 2.6, we can see that reviews are grouped into several sets R_i after fusing and grouping them (each set contains one review at least).

It is very important to provide customers with sufficient information to help them make an informed decision toward the target entity. Therefore, in addition to the numerical reputation value, we provide them with a pie chart (Figure 2.4) that depicts the distribution of sentiments over reviews (based on ratings).

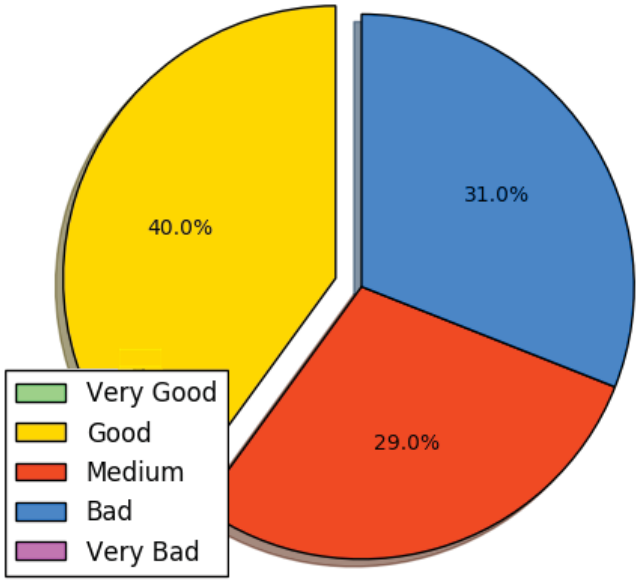


Figure 2.4: Distribution of sentiments over reviews

As illustrated in Figure 2.4, 40% of users are satisfied with the movie, 29% of them think that the movie is *"OK"* and the remaining 31% of them dissatisfy with the target movie.

2.4.5 Reputation generation

In their work, Yan et al. [3] have recruited ten volunteers to manually rate 1000 products after reading their descriptions and reviews. They averaged the volunteers' ratings (manually generated reputation values) toward each product, then, they computed the deviations between the manually generated reputation values and reputation values generated by their approach. However, we believe

that even if they have recruited 1000000 volunteers, there is no guarantee that their judgment is right toward the target products, since they may not have enough knowledge or experience to give proper judgment toward them. For that, we have chosen the IMDb weighted average vote as ground truth for the reason that IMDb is a platform where both regular movie fans and expert reviewers rate and share their opinions. We can see in Table 2.4 that more than 1257206 people gave a rating toward the Godfather 1 movie, which represent an enormous amount. Besides, the vast majority of those people share their ratings after watching the target movie, which imply that they have knowledge and opinion toward it, Thus, in order to evaluate the effectiveness of our method in generating reputation, we compare the final reputation value generated by our method and Yan et al. [3] with IMDb users weighted average vote applied by IMDb to represent a rating toward the target movie, which is a number ranging from 1 to 10 as shown in Figure 2.5.



Figure 2.5: IMDb users weighted average vote for The Godfather I movie

We compare our approach to Yan et al. work [3] by applying them to the datasets that we collected before. We varied the value of the opinion fusion threshold t_0 from 0.05 to 0.95 with a step of 0.05, and we used the evaluation measure *"Absolute Error"* where:

$$AER_{t_0}^{m_i} = |IMDbWAV^{m_i} - Rep_{t_0}^{m_i}| \tag{2.4}$$

We denote:

$AER_{t_0}^{m_i}$: Reputation Absolute Error for a specific t_0 value toward the target movie m_i.

$IMDbWAV^{m_i}$: IMDb weighted average vote toward the target movie m_i.

$Rep_{t_0}^{m_i}$: Reputation value computed by our method or Yan et al. [3] for a specific t_0 value toward the target movie m_i.

Figures 2.6, 2.7, 2.8, 2.9, and 2.10 represent comparison results between the two methods

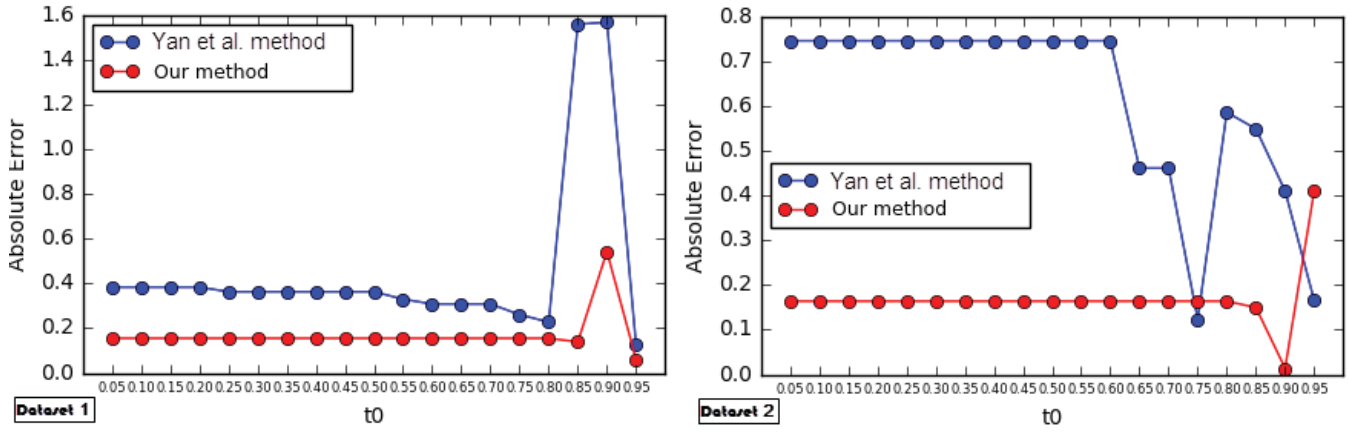


Figure 2.6: Comparison between Yan et al. [3] method and our contribution for dataset 1 and dataset 2

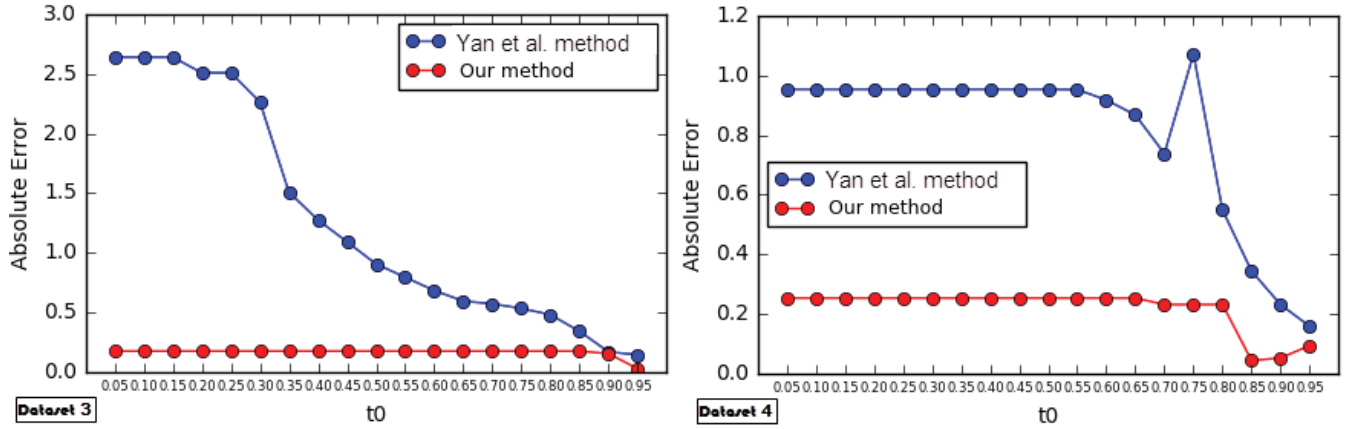


Figure 2.7: Comparison between Yan et al. [3] method and our contribution for dataset 3 and dataset 4

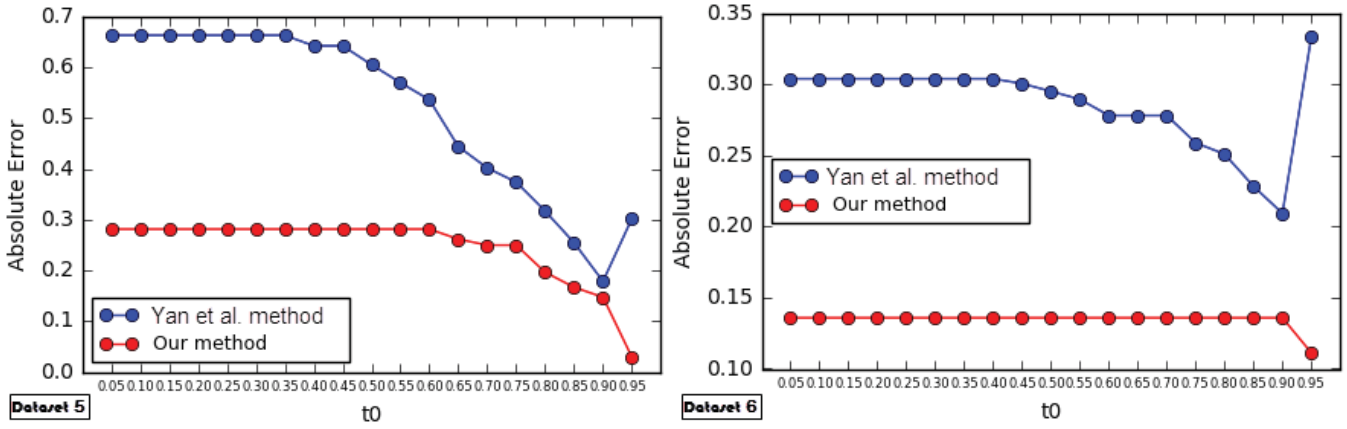


Figure 2.8: Comparison between Yan et al. [3] method and our contribution for dataset 5 and dataset 6

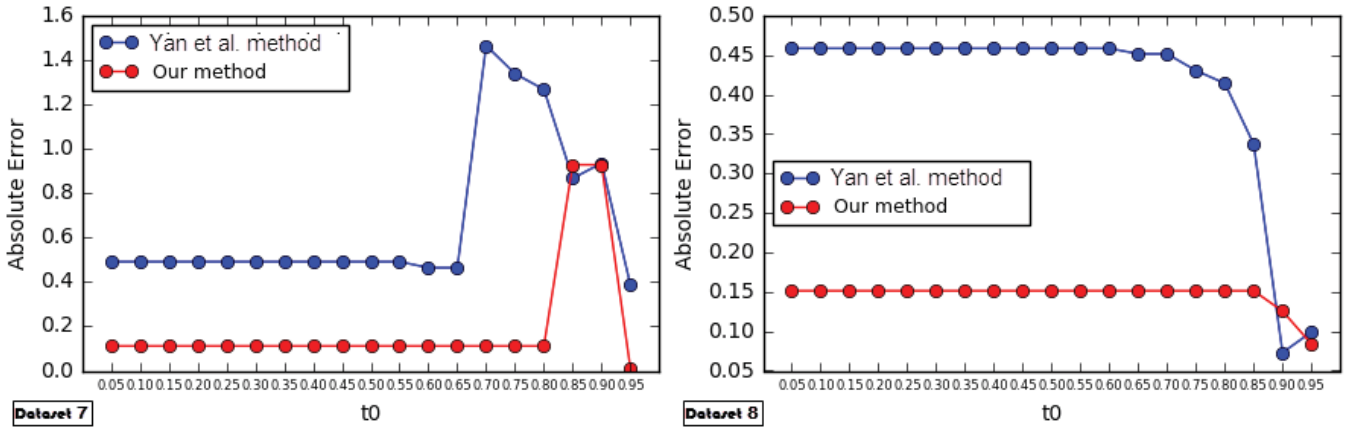


Figure 2.9: Comparison between Yan et al. [3] method and our contribution for dataset 7 and dataset 8

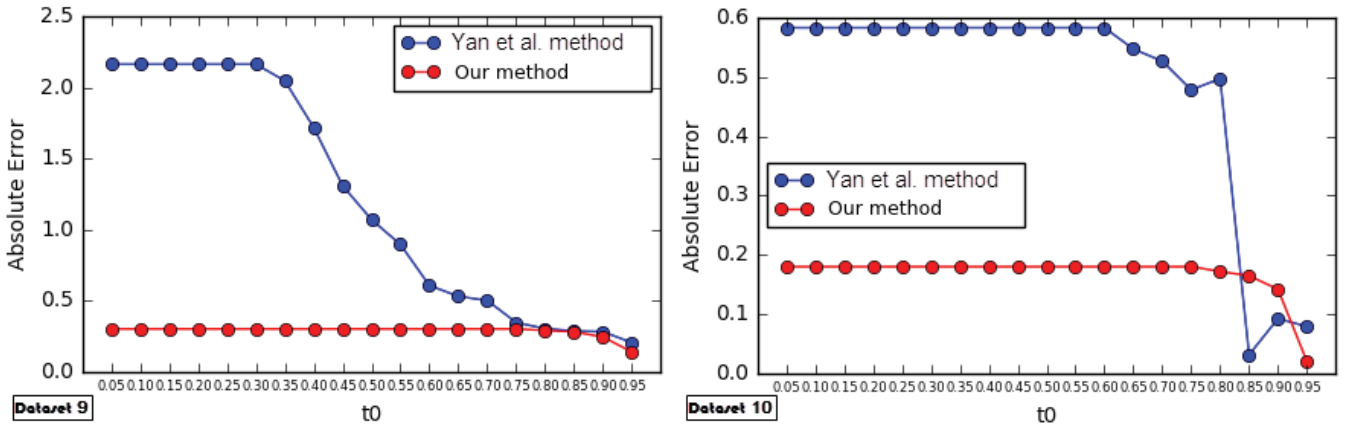


Figure 2.10: Comparison between Yan et al. [3] method and our contribution for dataset 9 and dataset 10

In Figure 2.6: dataset 1, Figure 2.7: dataset 3 and 4, Figure 2.8: dataset 5 and 6, and Figure 2.10: dataset 9, our method outperforms Yan et al. [3] method for all values of t_0 varied from 0.05 to 0.95 with a step of 0.05, additionally, we can see that for all datasets, our method works better than Yan et al. approach [3] when $0.05 \leq t_0 \leq 0.70$, even in the worst cases, Figure 2.6: dataset 1 when $t_0 = 0.90$, dataset 2 when $t_0 = 0.95$, and Figure 2.9: dataset 7 when $t_0 = \{0.85, 0.90\}$, the Absolute Error (2.4) of our method doesn't exceed 0.97. Table 2.7 summarizes the results given by the two methods.

Table 2.7: Experiment results summarization

	IMDb weighted average vote	Average reputation (Yan et al. [3] method)	$MAER^{m_i}$ (Yan et al. [3] method)	Average reputation (Our method)	$MAER^{m_i}$ (Our method)
Dataset 1	5.80	5.68401533142	0.458584523451	5.69411656318	**0.169344478854**
Dataset 2	8.20	7.58329159051	0.616708409489	8.033252139	**0.168193964067**
Dataset 3	8.30	7.0236673294	1.2763326706	8.14088245291	**0.161469513573**
Dataset 4	7.80	6.99168965196	0.808310348038	7.60117164592	**0.218177388535**
Dataset 5	5.80	5.27820594837	0.521794051633	5.55387542084	**0.246124579156**
Dataset 6	7.60	7.31385618805	0.286143811945	7.46540816442	**0.134591835581**
Dataset 7	6.50	6.26579054984	0.662862169432	6.30898667249	**0.191809871998**
Dataset 8	6.50	6.09915992375	0.408525260606	6.35358792992	**0.146412070077**
Dataset 9	9.20	7.98707747356	1.21292252644	8.91509031525	**0.284909684753**
Dataset 10	6.70	6.26880591164	0.486749395551	6.53429997883	**0.167701915098**

From Table 2.7, we can see that our method provides the nearest reputation value to the ground truth (IMDb weighted average vote) for all datasets, also, the $MAER^{m_i}$ (2.5) of our method doesn't exceed 0.29. However, the $MAER^{m_i}$ reaches 1.27 in the third dataset and 1.21 in the ninth dataset for Yan et al. [3] method. Moreover, the ninth dataset contains reviews of The Godfather I movie which is considered as one of the best movies of all time. Thus, this high error value could influence the popularity of the movie, which makes reputation generation a very delicate task.
We compute the $MAER^{m_i}$ by applying this formula:

$$MAER^{m_i} = \frac{1}{T}. \sum_{\substack{min \leq t_0 \leq max \\ t_0 \, multiple \, of \, step}} AER^{m_i}_{t_0} \tag{2.5}$$

We denote:

$MAER^{m_i}$: Reputation Mean Absolute Error toward the target movie m_i for all t_0 values.

$AER_{t_0}^{m_i}$: Reputation Absolute Error for a specific t_0 value toward the target movie m_i.

T : Total number of t_0 values, in our case, it varies from 0.05 to 0.95 with a step of 0.05, which means $t_0 = \{0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95\}$ and $T = 19$.

2.4.6 Opinion fusion threshold t_0

After comparing the two methods and showing the effectiveness of ours, we can see that different values of the opinion fusion threshold t_0 could lead to different grouping results, which cause different custom reputation values and different final reputation values. Therefore, choosing a suitable t_0 value is very important. Thus, we conducted experiments to determine the value of t_0 that provides the best results. We set the value of the opinion fusion threshold t_0 from 0.05 to 0.95 with a step of 0.05. Figure 2.11 shows the Absolute Error (AE) (2.4) of the final reputation value computed by our method.

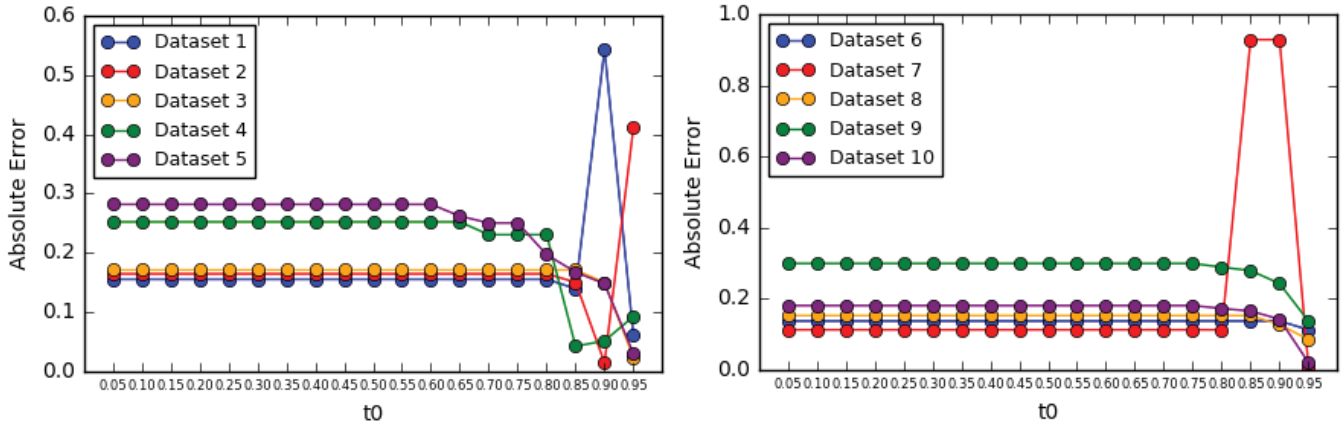


Figure 2.11: Absolute Error (AE) of reputation value computed by our method

From Figure 2.11, we observed that for all datasets, the Absolute Error (2.4) value is stable

when $0.05 \leq t_0 \leq 0.6$, and then, it begins to vary either by increasing or decreasing. But, it is still hard to determine the suitable t_0 value for generating an accurate reputation toward the target movie. That 's why we conducted more experiments. We computed the $MAER_{t_0}$ (2.6) between IMDb weighted average votes (ground truth) and the reputation values generated by our method using all movies of datasets for each t_0 value between 0.05 and 0.95 with a step of 0.05.

Table 2.8: $MAER_{t_0}$ between IMDb weighted average votes and the reputation values computed by our method using all the reviews of the 10 movies for t_0 value varied from 0.05 to 0.95 with a step of 0.05

Opinion fusion threshold t_0 value	$MAER_{t_0}$
$t_0 = 0.05$	0.18978855
$t_0 = 0.10$	0.18978855
$t_0 = 0.15$	0.18978855
$t_0 = 0.20$	0.18978855
$t_0 = 0.25$	0.18978855
$t_0 = 0.30$	0.18978855
$t_0 = 0.35$	0.18978855
$t_0 = 0.40$	0.18978855
$t_0 = 0.45$	0.18978855
$t_0 = 0.50$	0.18978855
$t_0 = 0.55$	0.18978855
$t_0 = 0.60$	0.18978855
$t_0 = 0.65$	0.18780241
$t_0 = 0.70$	0.18445943
$t_0 = 0.75$	0.18445943
$t_0 = 0.80$	0.17724106
$t_0 = 0.85$	0.23243659
$t_0 = 0.90$	0.24759925
$t_0 = 0.95$	**0.09713621**

Both Figure 2.12 and Table 2.8 show that the $MAER_{t_0}$ (2.6) is stable when $t_0 \leq 0.60$. We also found that our reputation generation method performs better when $t_0 = 0.95$, since the Mean Absolute Error between IMDb weighted average votes and the values computed by formula (2.3)

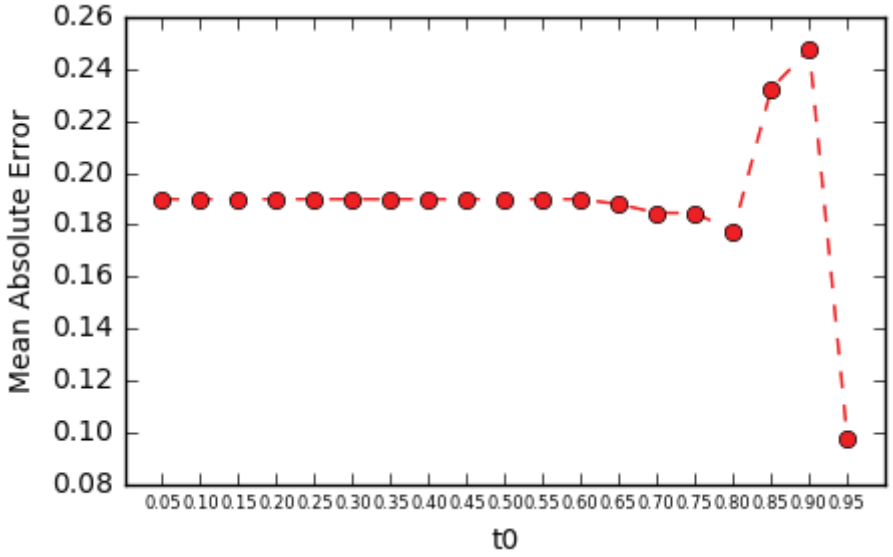


Figure 2.12: $MAER_{t_0}$ between IMDb weighted average votes and the reputation values computed by our method using all the reviews of 10 movies for t_0 varied from 0.05 to 0.95 with a step of 0.05

using all the reviews of the 10 movies reaches its minimum.
We compute the $MAER_{t_0}$ by applying this formula:

$$MAER_{t_0} = \frac{1}{E}. \sum_{1 \leq i \leq E} AER_{t_0}^{m_i} \tag{2.6}$$

We denote:

$MAER_{t_0}$: Mean Absolute Error between IMDb weighted average votes (ground truth) and the reputation values generated by our method for all movies of datasets for a specific t_0 value.

E : number of target entities (In this case $E = 10$).

2.4.7 Classification phase impact on reputation generation

As shown in Figure 2.3, Our proposed classification step gives the highest sentiment classification accuracy for all datasets compared to Naïve Bayes and SVM classifiers. It is important to analyze the impact of the classification step on reputation generation. For this purpose, we conducted more experiments in which we compared our method to two other approaches. The first approach

uses Naïve Bayes for the classification phase, and the second one uses SVM. For each approach, we computed the $MAER^{m_i}$ between IMDb weighted average vote and the final reputation value ($t_0 = 0.95$) computed by each approach for all datasets using formula (2.5).

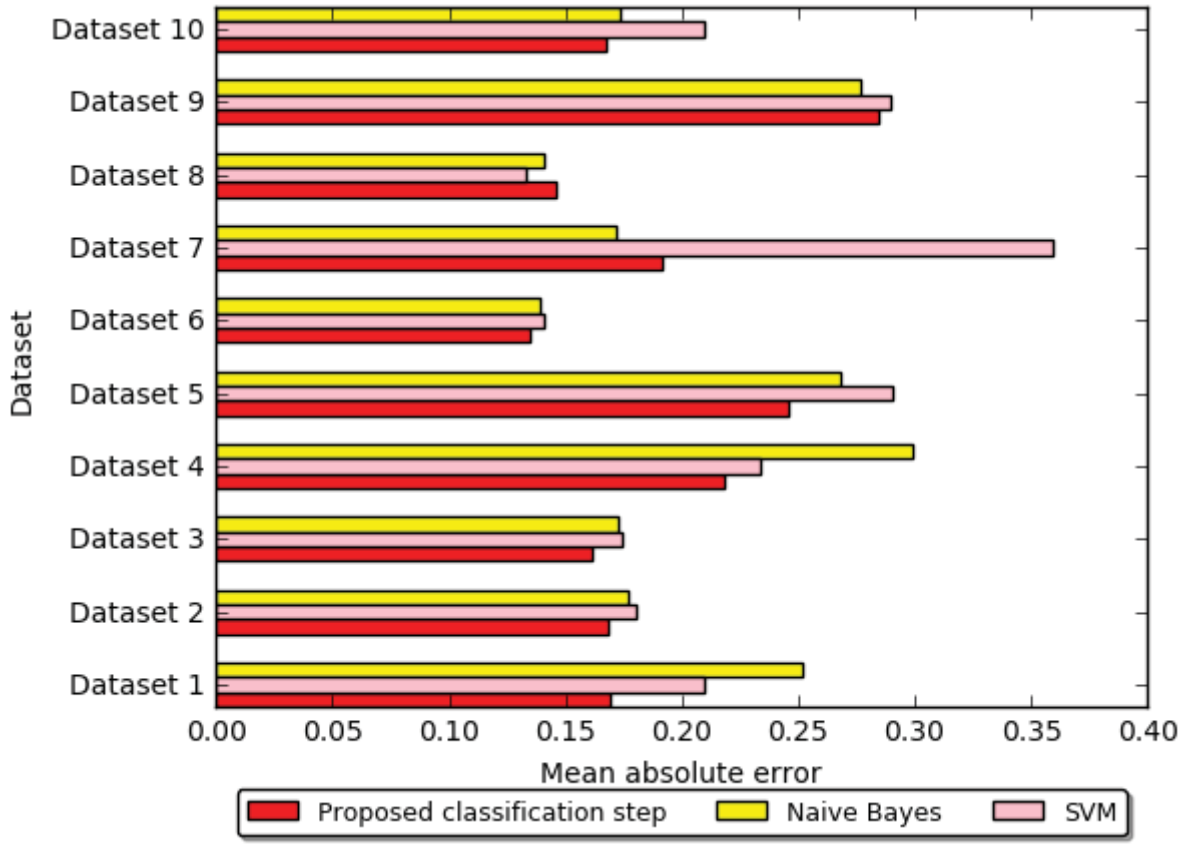


Figure 2.13: Reputation Mean Absolute Error for the three approaches

We can see from Figure 2.13 that our proposed classification step leads to the lowest reputation Mean Absolute Error value (2.5) for datasets 1, 2, 3, 4, 5, 6, and 10, more than that, there is a very small difference between the $MAER^{m_i}$ of our approach and the $MAER^{m_i}$ computed during the use of Naïve Bayes for the classification phase in datasets 7, 8 and 9. Thus, we conclude that the use of an accurate sentiment classifier leads to an accurate reputation value toward the target entity.

2.5 Summary and Limitations

In this chapter, we propose a reputation generation system that combines Naïve Bayes and Linear Support Vector Machine (LSVM) classifiers for the purpose of separating online reviews into positives or negatives (sentiment analysis) before grouping them into different opinion sets based on

their semantic relations, we also propose to compute the final reputation value using the Weighted Arithmetic Mean. As has been noted, our approach improves the work proposed in [3] by generating accurate reputation values toward different movies. However, the main issues of our work are:

- Polarity classification step: the classifier depends on the domain of use (politics, economic, business, sport...) and the language (English, French, Chinese, Arabic...). Consequently, it is not practical to use different datasets for each domain and each language. Fortunately, there are many methods to handle those issues such as Machine Translation, Transfer Learning, etc... [31].

- Useless reviews: a lot of reviews are irrelevant to the target entity (ads, spam). Therefore, adding a filtering phase [31, 86] will reduce the processing time and will increase the accuracy of the reputation system (the system will deal only with relevant and useful reviews).

The next chapter will describe MTVRep, a movie and TV show reputation system that exploits fine-grained sentiment analysis and semantic analysis for the purpose of generating and visualizing reputation toward movies and TV shows.

Algorithm 1 Polarity Classification phase

Define : $V = \{v_1, v_2, ..., v_{n\}}$: the set of ratings attached to each review;

$R = \{r_1, r_2, ..., r_{n\}}$: the set of reviews expressed in natural language;

$NBPrediction = \{nbp(r_1), nbp(r_2), ..., nbp(r_n)\}$: the set of polarity of reviews R predicted by Naïve Bayes classifier;

$SVMPrediction = \{svmp(r_1), svmp(r_2), ..., svmp(r_p)\}$: the set of polarity of *remainingReviews* predicted by SVM classifier;

remainingReviews: the set of indexes for remaining reviews after the first step;

remaining1Reviews: the set of indexes for remaining reviews after the second step;

Input : Reviews and their user ratings: R and V.

Output: The set of positive reviews and the set of negative reviews: *positiveReviews* and *negativeReviews*.

Apply Naïve Bayes classifier to predict the polarity of reviews R and store the results in *NBPrediction*.

for $i \leftarrow 0$ **to** $n-1$ **by** 1 **do**

  if $nbp(r_i)$ *is 'positive'* ***and*** $v_i > 5$ **then**
    add i to *positiveReviews*
  else if $nbp(r_i)$ *is 'negative'* ***and*** $v_i \leq 5$ **then**
    add i to *negativeReviews*
  else
    add i to *remainingReviews*
  end if

end for

Apply SVM classifier to predict the polarity of *remainingReviews* and store the results in *SVMPrediction*.

lengthRemainingReviews $\leftarrow$ *length of remainingReviews*

for $i \leftarrow 0$ **to** *lengthRemainingReviews* **by** 1 **do**

  if $nbp(r_{remainingReviews[i]})$ *is 'positive'* ***and*** $svmp(r_{remainingReviews[i]})$ *is 'positive'* **then**
    add *remainingReviews*$[i]$ to *positiveReviews*
  else if $nbp(r_{remainingReviews[i]})$ *is 'negative'* ***and*** $svmp(r_{remainingReviews[i]})$ *is 'negative'* **then**
    add *remainingReviews*$[i]$ to *negativeReviews*
  else
    add *remainingReviews*$[i]$ to *remaining1Reviews*
  end if

end for

$lengthRemaining1Reviews \leftarrow length\ of\ remaining1Reviews$

for $i \leftarrow 0$ **to** $lengthRemaining1Reviews$ **by** 1 **do**

  if $v_{remaining1Reviews[i]} > 5$ **then**

    add $remaining1Reviews[i]$ to $positiveReviews$

  else

    add $remaining1Reviews[i]$ to $negativeReviews$

  end if

end for

Chapter 3

Fine-grained Opinion Mining and Semantic Analysis for Reputation Generation

3.1 Introduction

The exponential growth of Web 2.0 has dramatically impacted the evolution of e-commerce platforms [22, 87–89]. On the one hand, some recent statistics show that 72% of customers will not take action until they read reviews, and only 6% of consumers don't trust customer reviews at all, on the other hand, the number of user-generated reviews attached to an online entity could easily exceed thousands [90,91]. Thus, a potential customer doesn't have time or effort to examine all the reviews manually in order to make an informed decision toward it [92, 93].

Little research has been conducted in mining customer and user reviews with regard to feature-based summarization and reputation generation for the purpose of supporting customers' decision making process in E-commerce (buying, renting, booking ...). Over the last two decades, few opinion summarizer systems have been proposed to produce an aspect-based summary for product reviews [94], movie reviews [88], hotel reviews [22] and local service reviews [21]. Backing to the reputation generation task, to the best of our knowledge, there are very few reputation systems that have been proposed to compute a single reputation value toward different entities based on fusing and mining user and customer reviews expressed in natural language [3, 24, 25, 95]. Yan et al. [3] applied opinion mining and fusion techniques on product reviews. Benlahbib & Nfaoui [24] used K-Means clustering algorithm on movie reviews. The same authors [25] incorporated semantic

and sentiment analysis to generate reputation by mining user and customer reviews expressed in natural language (English).

An important issue that was neglected in the past research on reputation generation is identifying the sentiment strength during the phase of sentiment classification and opinion fusion. In fact, existing works have only focused on classifying reviews into positive or negative before generating a single reputation value, disregarding the sentiment strength.

In this chapter, we propose MTVRep, a movie and TV show reputation system that applies fine-grained opinion mining to separate reviews into five opinion groups: strongly negative, weakly negative, neutral, weakly positive, and strongly positive. Then, it computes a custom score for each group based on the acquired statistics of each group, i.e., the number of reviews in each group, the sum of their ratings, and the sum of their semantic similarity (ELMo and cosine metric). Finally, a numerical reputation value is produced toward the target movie or TV show using the weighted arithmetic mean.

In this manner, this study addressed the following research question: with the combination of fine-grained opinion mining and semantic analysis, can the proposed reputation system offer better results in terms of reputation generation than the previous reputation systems (consider only semantic relations)?

3.2 Problem definition

This section covers the necessary background for understanding the remainder of this chapter, including the problem definition.

In this chapter, we face the problem of generating reputation for movies and TV shows by combining fine-grained opinion mining and semantic analysis. Given a set of reviews $R = \{r_1, r_2, \ldots, r_n\}$ expressed for an entity E, the set of their attached ratings $V = \{v_1, v_2, \ldots, v_n\}$ where $v_i \in [1,5]$ or $v_i \in [1,10]$ depending on the rating system. The goal is to separate reviews into five opinion groups: strongly negative, weakly negative, neutral, weakly positive, and strongly positive by applying the Multinomial Naïve Bayes model where $G^{polarity} = \{r_1^{polarity}, r_2^{polarity}, \ldots, r_m^{polarity}\}$ is the opinion group that contains reviews that hold the sentiment orientation *polarity* and $V^{polarity} = \{v_1^{polarity}, v_2^{polarity}, \ldots, v_m^{polarity}\}$ is the set of their numerical ratings, then, to compute a custom

score $CS(G^{polarity})$ for each opinion group $G^{polarity}$ by considering some statistics of each opinion group such as $SS^{polarity}$: the sum of semantic similarity for $G^{polarity}$, $SV^{polarity}$: the sum of ratings for $G^{polarity}$, $N^{polarity}$: the number of reviews in $G^{polarity}$, and $maxR$: the highest value of user ratings (5 or 10) depending on the rating scale (1 to 5 or 1 to 10). Finally, we compute the reputation value $Rep(E)$ toward the target entity E using the Weighted Arithmetic Mean. Table 3.1 presents the descriptions of notations used in the rest of this chapter.

Table 3.1: Symbol denotation

Symbol	Description
E	The target entity
R	The set of reviews expressed for the entity E
V	The set of ratings expressed for the entity E
$G^{polarity}$	The opinion group that contains reviews that hold the sentiment orientation *polarity*
$V^{polarity}$	The set of ratings attached to $G^{polarity}$ reviews
$SS^{polarity}$	The sum of semantic similarity for $G^{polarity}$
$SV^{polarity}$	The sum of ratings for $G^{polarity}$
$N^{polarity}$	The number of reviews in $G^{polarity}$
$maxR$	The highest value of user ratings (5 or 10) depending on the rating scale (1 to 5 or 1 to 10)
$CS(G^{polarity})$	The custom score for opinion group $G^{polarity}$
$Rep(E)$	The reputation value toward entity E

3.3 Proposed system

3.3.1 System overview

The proposed approach consists mainly of four steps:

1. We collect movie and TV show reviews from IMDb[1] website using the web scraping tool

[1] `https://www.imdb.com/`

ScrapeStorm[2], then, we preprocess them.

2. We train the Multinomial Naïve Bayes model on the 5-class Stanford Sentiment Treebank (SST-5) dataset in order to perform fine-grained sentiment analysis. The model classifies the collected reviews into five opinion groups: strongly negative, weakly negative, neutral, weakly positive, and strongly positive.

3. For each opinion group, we acquire the sum of user ratings and the sum of reviews' semantic similarity. The semantic similarity between two reviews is computed as the cosine between their deep contextualized word embeddings (ELMo). These acquired statistics are used to compute a custom score for each opinion group.

4. We compute the movie or TV show numerical reputation value based on the opinion groups' scores by applying the weighted arithmetic mean.

Figure 3.1 illustrates the work-flow of the reputation system (MTVRep).

3.3.2 Fine-grained sentiment analysis

We classify the collected reviews into five opinion groups based on their sentiment intensities by applying the Multinomial Naïve Bayes model trained on the 5-class Stanford Sentiment Treebank (SST-5) dataset. The reasons behind using the Multinomial Naïve Bayes model are discussed in subsection 3.4.2.

3.3.3 Opinion groups custom scores

After separating movie and TV show reviews into five opinion groups: strongly negative, weakly negative, neutral, weakly positive, and strongly positive, we compute a custom score for each opinion group based on the sum of their ratings and the sum of their reviews' semantic similarity. The statistics of opinion groups are acquired by applying algorithm 2.

By applying algorithm 2, we retrieve for each group, the sum of their ratings and the sum of their semantic similarity. We propose formula (3.1) to compute a custom score for each opinion

[2] `https://www.scrapestorm.com/`

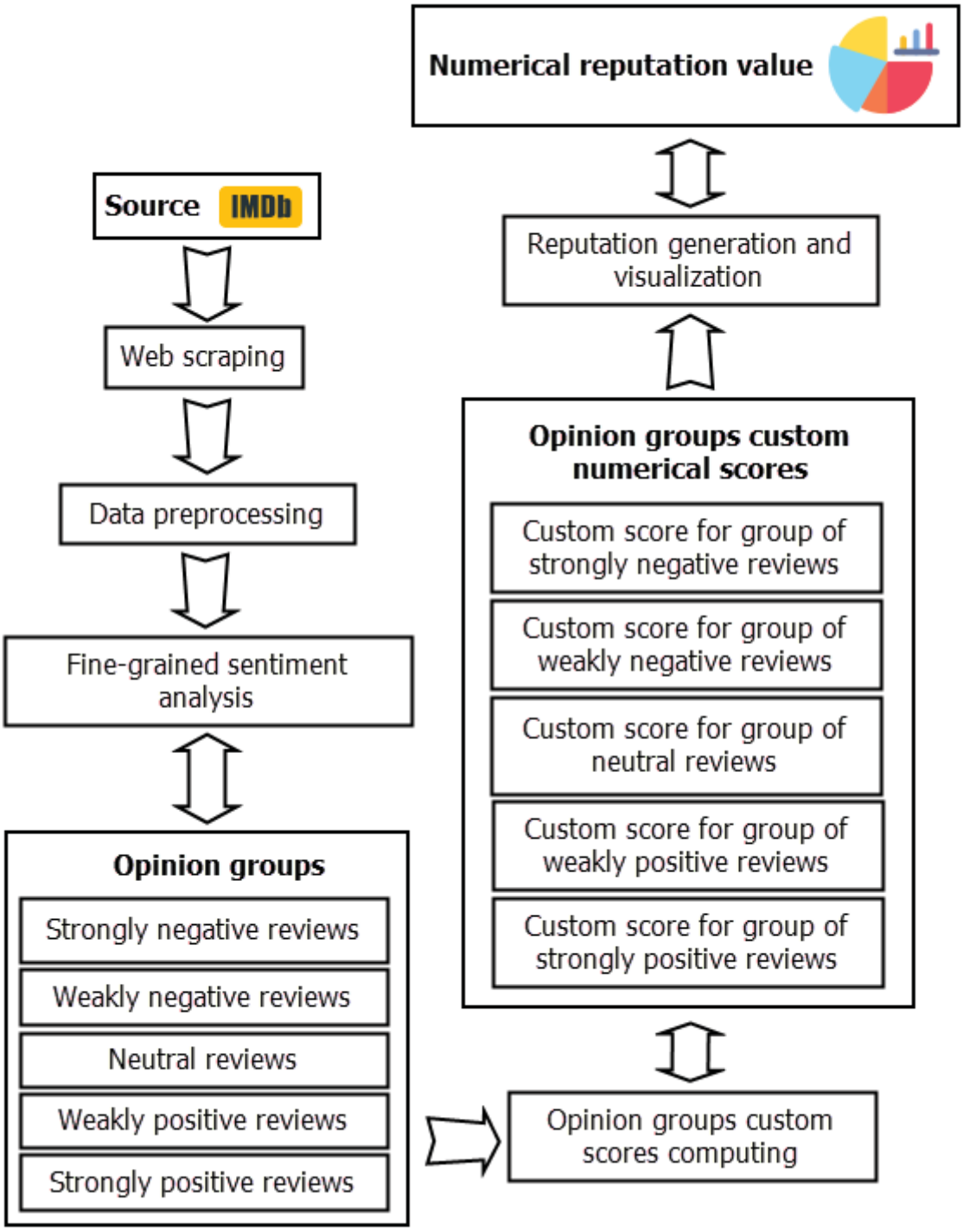


Figure 3.1: Reputation system pipeline

group.

$$CS(G^{polarity}) = \frac{\frac{maxR \cdot SS^{polarity}}{N^{polarity}} + \frac{SV^{polarity}}{N^{polarity}}}{2} \tag{3.1}$$

Algorithm 2 Opinion groups statistics acquisition

Define : $G^{polarity} = \{r_1^{polarity}, r_2^{polarity}, \ldots, r_m^{polarity}\}$: the opinion group that contains reviews that hold the sentiment orientation *polarity*.

$V^{polarity} = \{v_1^{polarity}, v_2^{polarity}, \ldots, v_m^{polarity}\}$: the set of ratings attached to $G^{polarity}$ reviews.

$SS^{polarity}$: the sum of semantic similarity for $G^{polarity}$ reviews.

$SV^{polarity}$: the sum of ratings for $G^{polarity}$ reviews.

$N^{polarity}$: the number of reviews in $G^{polarity}$.

$ELMo(r_i^{polarity})$: ELMo embeddings for review i from $G^{polarity}$.

$cos(ELMo(r_i^{polarity}), ELMo(r_j^{polarity}))$: the cosine similarity between ELMo embeddings for reviews i and j from $G^{polarity}$.

Input : Opinion groups, their lengths and their user ratings: $G^{polarity}$, $N^{polarity}$ and $V^{polarity}$.

Output: Opinion groups' statistics: $SS^{polarity}$ and $SV^{polarity}$

$polarity \leftarrow [strongly\ negative,\ weakly\ negative,\ neutral,\ weakly\ positive,\ strongly\ positive]$

```
/* After applying the trained model on the collected movie and TV show reviews, we
separate them into five opinion groups: strongly negative, weakly negative, neu-
tral, weakly positive, and strongly positive. For each opinion group, we acquire
the sum of their reviews' semantic similarity (cosine metric and ELMo embeddings)
and the sum of their ratings                                                    */
```

for *i in polarity* **do**
  $SS^i \leftarrow 0$
  $SV^i \leftarrow 0$
  for $j \leftarrow 1$ **to** N^i **do**
    $SS^i \leftarrow SS^i + cos(ELMo(r_1^i), ELMo(r_j^i))$
    $SV^i \leftarrow SV^i + v_j^i$
  end for
end for

Formula (3.1) could also be written as follows:

$$CS(G^{polarity}) = \frac{maxR \cdot SS^{polarity} + SV^{polarity}}{2 \cdot N^{polarity}} \tag{3.2}$$

We denote:

$maxR$: The highest value of user ratings (5 or 10) depending on the range of ratings (1 to 5 or 1 to 10).
$SS^{polarity}$: The sum of similarity for reviews contained in opinion group $G^{polarity}$.
$SV^{polarity}$: The sum of user ratings in opinion group $G^{polarity}$.
$N^{polarity}$: The number of reviews contained in opinion group $G^{polarity}$.

The custom score of each opinion group ranges between 1 and 5 or 1 and 10 depending on the range of user rating values.

Since the cosine metric returns values in the range of [0, 1], the average of the sum of semantic similarity for an opinion group is also between 0 and 1, therefore, we multiply the average of the sum of semantic similarity by 5 or 10 ($maxR$) to get a numerical value between 0 and 5 or 0 and 10, then, we add this value to the average of the sum of ratings and we divide them by 2.

3.3.4 Reputation generation

We propose formula (3.3) (weighted arithmetic mean) to compute the movie or TV show reputation value.

$$Rep(E) = \frac{\sum_{polarity} CS(G^{polarity}) \cdot N^{polarity}}{\sum_{polarity} N^{polarity}} \tag{3.3}$$

$CS(G^{polarity})$ is the custom score for opinion group $G^{polarity}$ computed by applying formula (3.1) or (3.2).

The movie or TV show reputation value has values in the range of [1, 5] or [1, 10] depending on the range of user ratings.

3.4 Experimental evaluation

3.4.1 Dataset gathering

We collect movie and TV show reviews and their numerical ratings from the IMDb web site using the web scraping tool ScrapeStorm. Figure 3.2 depicts the structure of IMDb user reviews.

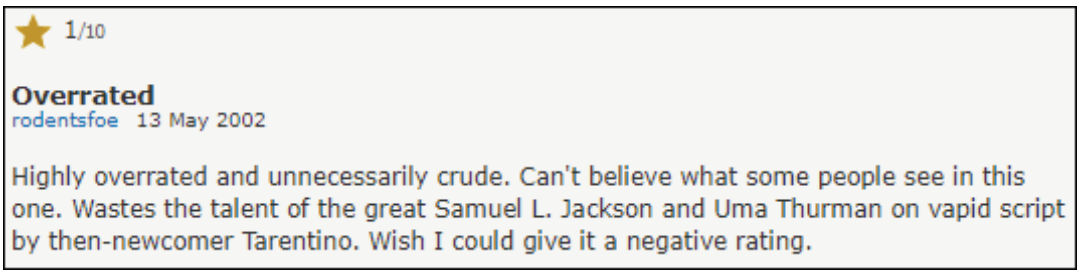


Figure 3.2: IMDb user reviews structure

The first ten datasets contain movie reviews and the remaining ten datasets contain TV show reviews. Table 3.2 shows the statistical information of the collected datasets.

	Movies	TV shows	Total
Number of reviews	1000	1000	2000
Number of entities	10	10	20

Table 3.2: Statistical information of the collected datasets

After collecting the reviews, we replace the missing rating values with the average of the ratings, then, we lowercase them and remove punctuation marks and numbers.

3.4.2 Training phase and fine-grained opinion mining

We train the Multinomial Naïve Bayes model with the SST-5 dataset. The training set contains 1092 strongly negative reviews, 2218 weakly negative reviews, 1624 neutral reviews, 2322 weakly positive reviews, and 1288 strongly positive reviews. The test set contains 279 strongly negative reviews, 633 weakly negative reviews, 389 neutral reviews, 510 weakly positive reviews, and 399 strongly positive reviews. Figure 3.3 depicts the distribution of training and test samples over the five classes.

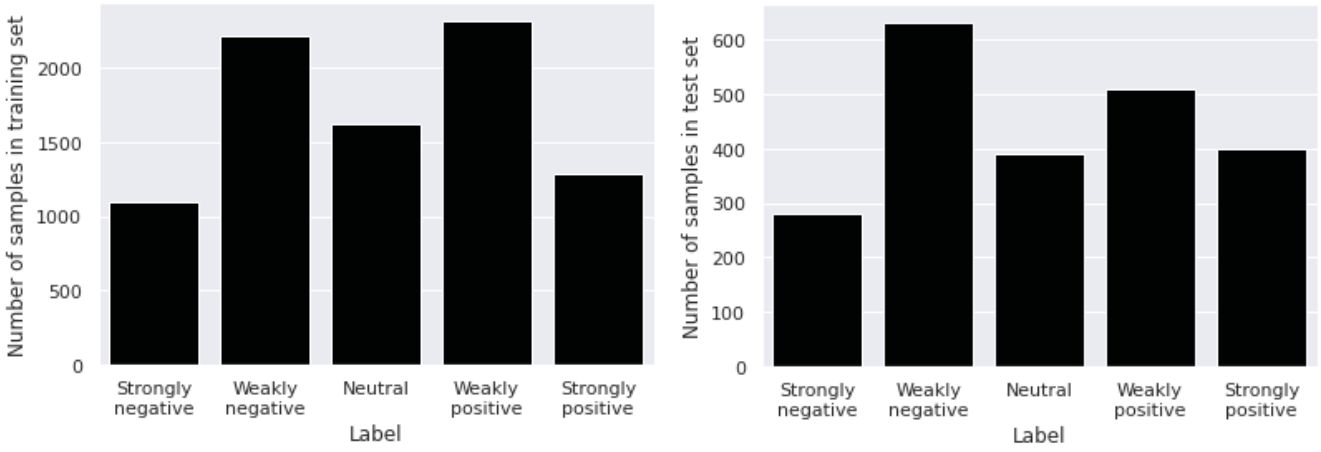


Figure 3.3: Number of samples in the SST-5 training and test set

Before feeding the data to the classifier for training, we preprocess them by removing punctuation marks, numbers, and whitespaces, then, we lowercase and lemmatize them.

After preprocessing the data, we must choose which classifier we will apply and which features we will use. Since deep learning models require substantial computing power (High-performance CPUs, GPUs, and RAM), we decided to work with one of the four models: Random Forest, Logistic Regression, Multinomial Naïve Bayes, and Linear Support Vector Machine (SVM). The last two classifiers (Naïve Bayes and SVM) have been recognized as the most popular supervised machine learning algorithms for polarity classification [85]. For feature selection, we have tried many combinations: unigrams (count), bigrams (count), trigrams (count), TF-IDF unigrams, TF-IDF bigrams, and TF-IDF trigrams. We discarded some popular models such as word2vec and doc2vec because Wang et al. [96] have conducted experiments on Naïve Bayes, Logistic Regression, and Linear Support Vector Classifier (SVC) for short text classification using TF-IDF weighting, word2vec, and paragraph2vec (doc2vec), and they have reported that TF-IDF/Counter features have the highest accuracy, while word2vec next, and doc2vec has the lowest accuracy. Table 3.3 summarizes the classification results of the four classifiers on the SST-5 dataset.

From table 3.3, we can see that the Multinomial Naïve Bayes classifier achieves the best classification results when it's trained with unigrams (count). Logistic Regression and linear SVM classifiers also gave good results when they are trained with unigrams (count) or TF-IDF unigrams. The worst results are provided by Random Forest since it achieves a 0.36 accuracy at its best. Figure 3.4 depicts the confusion matrix of the Multinomial Naive Bayes (unigrams) for the SST-5 test set.

	Macro average precision	Macro average recall	Macro average f1-score	Weighted average precision	Weighted average recall	Weighted average f1-score	Accuracy
Random Forest (unigrams)	0.40	0.31	0.30	0.40	0.36	0.33	0.36
Random Forest (bigrams)	0.34	0.29	0.28	0.34	0.32	0.31	0.32
Random Forest (trigrams)	0.29	0.23	0.20	0.31	0.23	0.22	0.23
Random Forest (TF-IDF unigrams)	0.40	0.30	0.28	0.39	0.35	0.31	0.35
Random Forest (TF-IDF bigrams)	0.34	0.29	0.28	0.34	0.32	0.31	0.32
Random Forest (TF-IDF trigrams)	0.28	0.22	0.20	0.29	0.23	0.21	0.23
Multinomial Naive Bayes (unigrams)	0.43	**0.38**	**0.38**	0.43	**0.43**	**0.41**	**0.43**
Multinomial Naive Bayes (bigrams)	0.36	0.30	0.29	0.36	0.35	0.32	0.35
Multinomial Naive Bayes (trigrams)	0.31	0.26	0.24	0.31	0.29	0.26	0.29
Multinomial Naive Bayes (TF-IDF unigrams)	**0.48**	0.34	0.29	**0.46**	0.41	0.34	0.41
Multinomial Naive Bayes (TF-IDF bigrams)	0.38	0.29	0.24	0.38	0.35	0.29	0.35
Multinomial Naive Bayes (TF-IDF trigrams)	0.29	0.24	0.19	0.30	0.29	0.23	0.29
Logistic Regression (unigrams)	0.42	0.37	0.37	0.42	0.41	0.39	0.41
Logistic Regression (bigrams)	0.38	0.28	0.23	0.37	0.34	0.27	0.34
Logistic Regression (trigrams)	0.36	0.23	0.18	0.35	0.28	0.22	0.28
Logistic Regression (TF-IDF unigrams)	0.42	0.35	0.34	0.41	0.40	0.37	0.40
Logistic Regression (TF-IDF bigrams)	0.43	0.28	0.23	0.41	0.35	0.27	0.35
Logistic Regression (TF-IDF trigrams)	0.30	0.23	0.17	0.32	0.29	0.21	0.29
Linear SVM (unigrams)	0.38	0.37	0.37	0.39	0.40	0.39	0.40
Linear SVM (bigrams)	0.33	0.31	0.31	0.34	0.34	0.33	0.34
Linear SVM (trigrams)	0.31	0.25	0.22	0.32	0.29	0.25	0.29
Linear SVM (TF-IDF unigrams)	0.38	0.38	0.38	0.39	0.41	0.39	0.41
Linear SVM (TF-IDF bigrams)	0.33	0.31	0.31	0.34	0.34	0.33	0.34
Linear SVM (TF-IDF trigrams)	0.31	0.27	0.25	0.31	0.30	0.27	0.30

Table 3.3: Sentiment analysis classification results

We mention that $\boldsymbol{BERT_{base}}$ achieves a 0.45 accuracy and 0.40 macro average f1-score, GRU-RNN-WORD2VEC achieves a 0.45 accuracy and Recursive Neural Tensor Network achieves a 0.46 accuracy. Besides, deep learning algorithms take a long time to train (Table 3.4) due to the large number of parameters. Based on that, we have chosen to apply the Multinomial Naïve Bayes classifier since it achieves an accuracy of 0.43 and it doesn't require substantial computing power to be trained. Table 3.4 depicts the training time of Bidirectional Gated Recurrent Unit (Bi-GRU), Bidirectional Long Short-Term Memory (Bi-LSTM), Recurrent Neural Network (RNN), and Multinomial Naïve Bayes (MNB) on the SST-5 dataset.

One of the benefits of fine-grained opinion mining is that it provides a better understanding of the distribution of reviews over the five emotion classes, therefore, visualizing these five classes will help users and customers make up their minds about the target item (buying, renting).

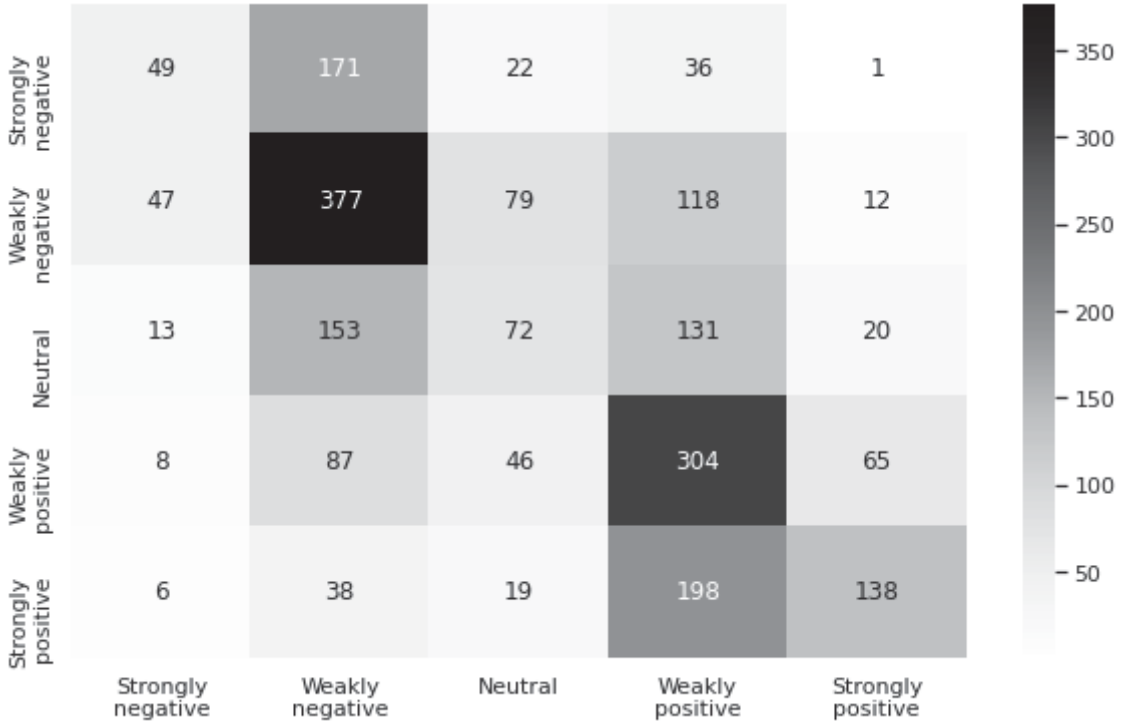


Figure 3.4: Confusion matrix of the Multinomial Naive Bayes (unigrams) for the SST-5 test set

Table 3.4: Training time of Bidirectional Gated Recurrent Unit (Bi-GRU), Bidirectional Long Short-Term Memory (Bi-LSTM), Recurrent Neural Network (RNN) and Multinomial Naïve Bayes (MNB) on the SST-5 dataset

Model	Epochs	Batch size	Training time (seconds)
Bi-GRU	50	64	210.10
Bi-LSTM	50	64	180.25
RNN	50	64	85.26
MNB	–	–	**3.77**

3.4.3 Reputation evaluation

MTVRep offers a comprehensive reputation visualization form (Figure 3.5) by depicting the numerical reputation value and the distribution of reviews over the five emotion classes, Table 3.5 shows comparison results between MTVRep and previous studies in terms of reputation visualization.

An important issue that was neglected in the past research on reputation generation is identifying the sentiment strength during the phase of opinion mining. Actually, existing studies have only

Table 3.5: Comparison results: reputation visualization

Work	Distribution of reviews polarity	Numerical reputation value
Yan et al. (2017) [3]	✗	✓
Benlahbib & Nfaoui (2019) [24]	✗	✓
Benlahbib et al. (2019) [95]	✗	✓
Benlahbib & Nfaoui (2020) [25]	✓	✓
MTVRep	✓	✓

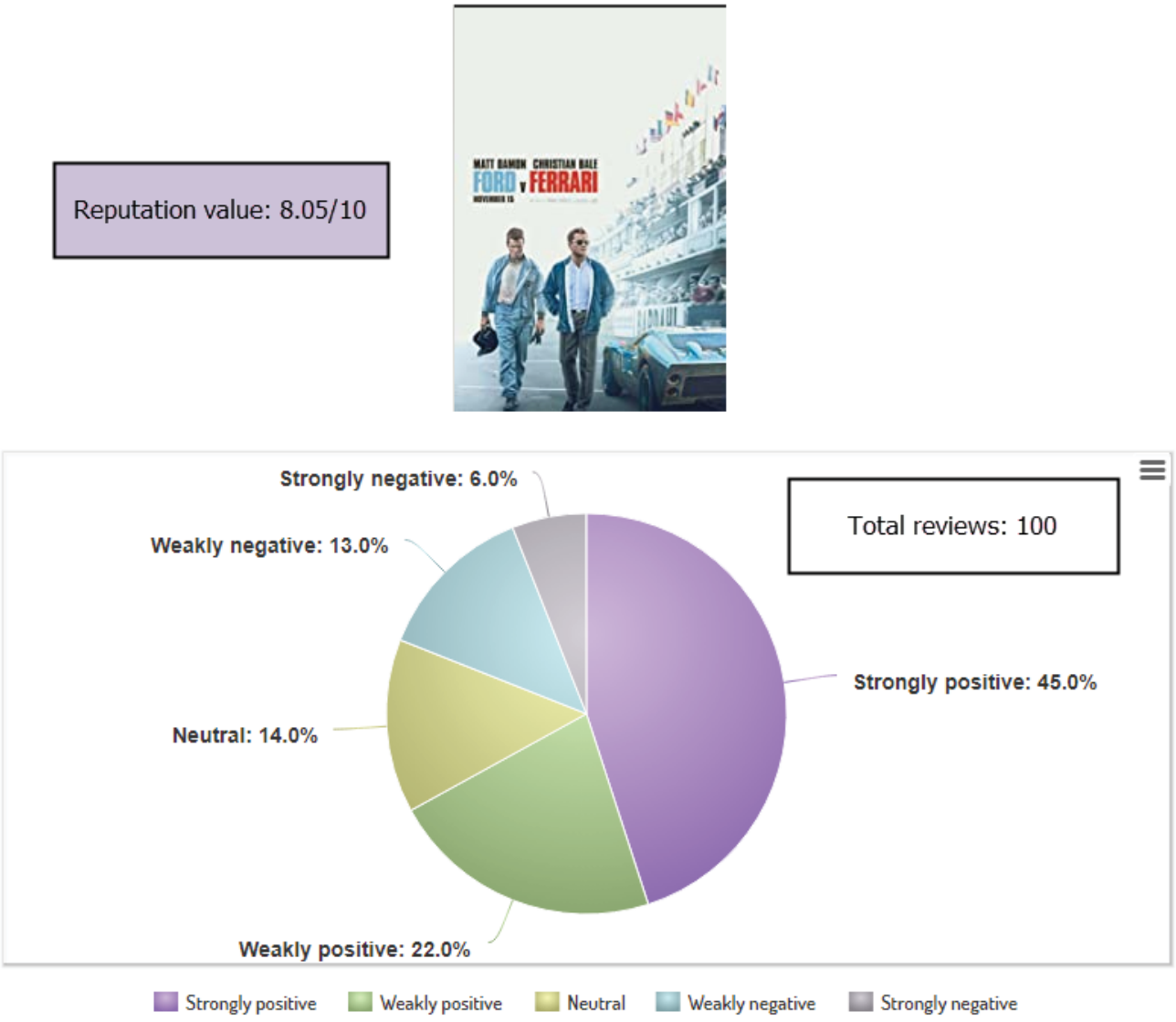


Figure 3.5: Reputation visualization

focused on classifying reviews as positive or negative, disregarding sentiment intensity. Therefore, we propose MTVRep, a movie and TV show reputation system that combines fine-grained sentiment analysis and semantic analysis for the purpose of generating and visualizing reputation toward movies and TV shows. Table 3.6 depicts the features exploited by previous studies and MTVRep during reputation generation and visualization.

Table 3.6: Comparison results: exploited features

Work	Semantic	Sentiment (binary)	Sentiment (fine-grained)
Yan et al. (2017) [3]	✓	✗	✗
Benlahbib & Nfaoui (2019) [24]	✓	✗	✗
Benlahbib et al. (2019) [95]	✓	✓	✗
Benlahbib & Nfaoui (2020) [25]	✓	✓	✗
MTVRep	✓	✗	✓

In order to evaluate the performance of MTVRep in generating accurate reputation values toward various movies and TV shows, we compared it to Yan et al. [3] reputation system. We set the opinion fusion threshold t_0 to 0.15 since the authors mentioned that their reputation system performs in its best when $t_0 = 0.15$. We applied the two reputation systems on the twenty collected datasets. The chosen evaluation measure is the squared error between the movie or TV show IMDb Weighted Average Ratings and the numerical reputation value computed by one of the two reputation systems.

The formula of the squared error is: $SE = (x_i - y_i)^2$ where x_i is the reputation value returned by one of the two systems and y_i is the IMDb Weighted Average Ratings toward the target movie or TV show. Figure 3.6 depicts the IMDb Weighted Average Ratings for the **Forrest Gump** movie.



Figure 3.6: IMDb Weighted Average Ratings

According to IMDb[3]: *"IMDb publishes weighted vote averages rather than raw data averages. Various filters are applied to the raw data in order to eliminate and reduce attempts at vote stuffing by people more interested in changing the current rating of a movie than giving their true opinion of it. The exact methods we use will not be disclosed. This should ensure that the policy remains effective. The result is a more accurate vote average".* The motivation behind choosing the squared error instead of absolute error resides in the fact that reputation systems don't tolerate high error values. Consequently, the squared error will penalize large errors more. Figures 3.7 and 3.8 show the comparison results between the two reputation systems over the twenty datasets.

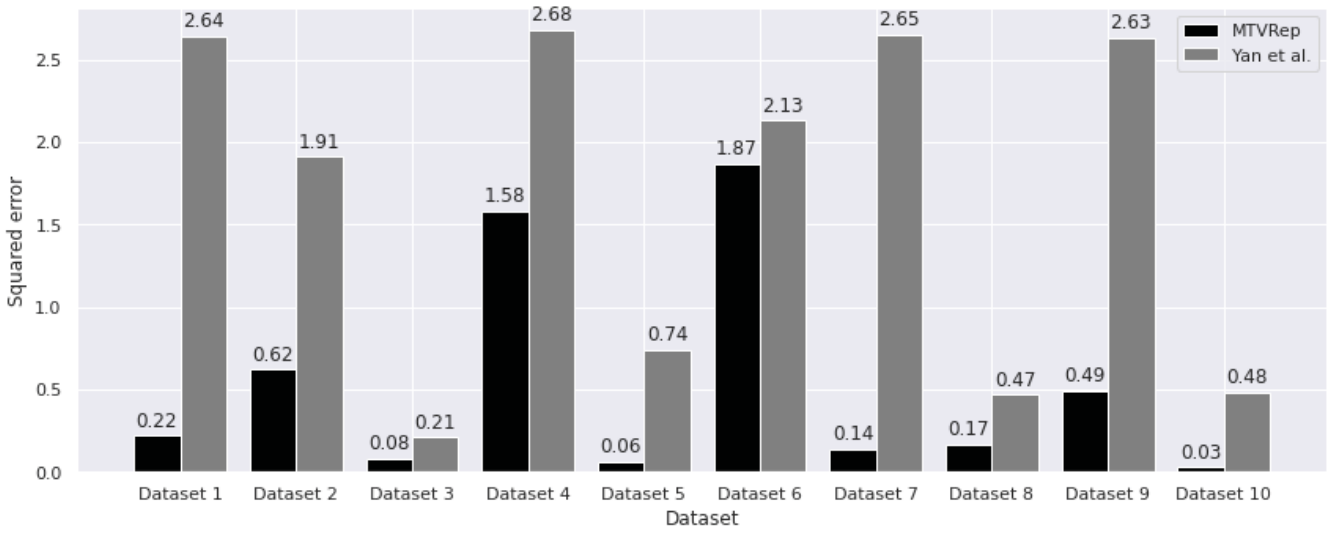


Figure 3.7: Squared error comparison results: dataset 1 to dataset 10

As illustrated in Figure 3.7, MTVRep produces the nearest reputation value to IMDb Weighted Average Ratings for the first ten datasets that contain movie reviews compared to Yan et al. reputation system [3]. We observe that the squared error of reputation system [3] exceeds 2.5 in dataset 1, dataset 4, dataset 7, and dataset 9. We also observe that the squared error of MTVRep doesn't surpass 0.1 in dataset 3, dataset 5, and dataset 10, which implies that the system generates accurate reputation values toward movies since the highest squared error achieved by MTVRep is 1.87 (dataset 6).

Figure 3.8 shows that except for dataset 15, MTVRep outperforms Yan et al. [3] reputation system on all the remaining nine datasets that contain TV show reviews. We also observe that

[3]`https://help.imdb.com/article/imdb/track-movies-tv/weighted-average-ratings/GWT2DSBYVT2F25SK?ref_=helpsect_pro_2_8#`

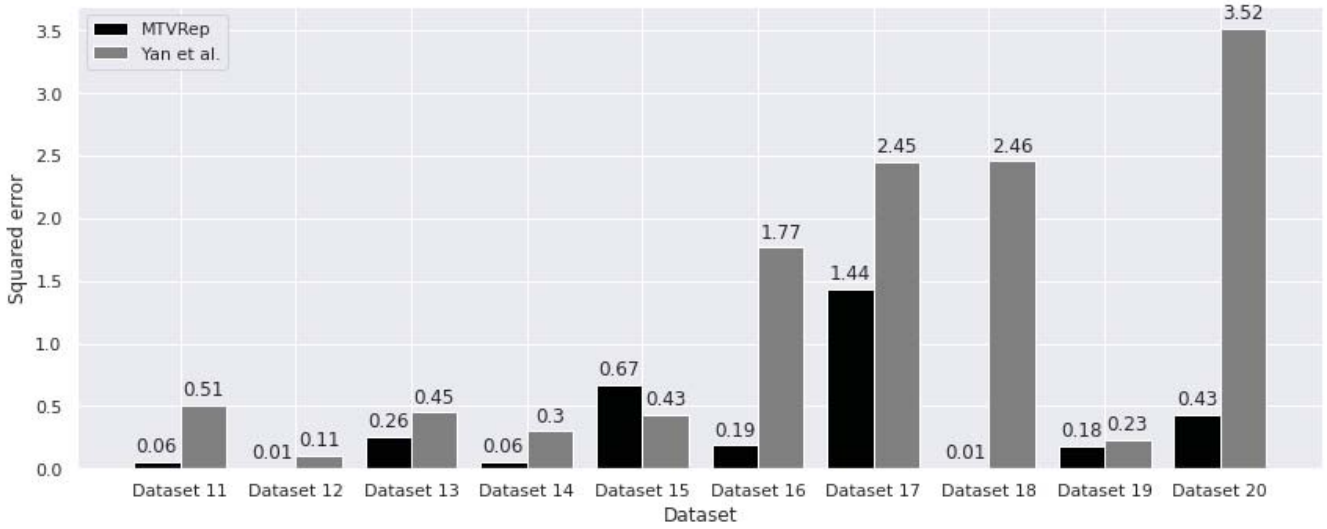


Figure 3.8: Squared error comparison results: dataset 11 to dataset 20

the squared error of reputation system [3] exceeds 3.5 in dataset 20, on the other hand, MTVRep doesn't exceed 1.44 in its worst.

We conclude that the proposed reputation system MTVRep performs well in generating and visualizing reputation for movies and TV shows since it produces the nearest reputation value to IMDb Weighted Average Ratings for both movies and TV shows compared to Yan et al. [3] reputation system.

3.5 Conclusion, summary and future direction

In this chapter, we have proposed MTVRep, a system that combines fine-grained opinion mining and semantic analysis for the purpose of generating and visualizing reputation toward movies and TV shows. The web scraping tool ScrapeStorm was used to collect 2000 movie and TV show reviews and their numerical ratings from IMDb, and the Multinomial Naïve Bayes classifier was trained on the SST-5 dataset to perform fine-grained opinion mining task. Experimental studies showed that MTVRep outperforms Yan et al. reputation system [3] since it produces the nearest reputation values to the ground truth (IMDb Weighted Average Ratings) for both movies and TV shows. We believe that MTVRep could be integrated into any platform where users share their reviews and ratings freely toward movies and TV shows.

Future works will focus on (1) using more sophisticated models for opinion mining such as

BERT and XLNet, (2) exploiting further features some of which are user credibility, review time, and review helpfulness, and (3) incorporating aspect-based opinion mining to enhance the reputation visualization form by showing more useful information toward the target movie or TV show (aspects).

The next chapter will present a reputation system that incorporates four review attributes: review helpfulness, review time, review sentiment polarity, and review rating in order to generate reputation toward various online items (products, movies, TV shows, hotels, restaurants, services). The system also provides a comprehensive reputation visualization form to help potential customers make an informed decision by depicting the numerical reputation value, opinion group categories, top-k positive reviews, and top-k negative reviews.

Chapter 4

Aggregating Customer Review Attributes for Online Reputation Generation

4.1 Introduction

The exponential growth of Web 2.0 has dramatically impacted the evolution of e-commerce platforms [89] [22] [87] [19]. Recent online shopping statistics showed that the number of users of some famous e-commerce websites such as Jingdong[1], Alibaba[2] and Amazon[3] has exceeded 1 billion [3]. Thereby, customer reviews attached to a product can easily surpass thousands [1, 12, 22]. In fact, while, a good number of reviews could indeed give a hint about the quality of an item, a potential customer may not have time or effort to read all reviews for the purpose of making a decision [92]. Thus, the need for the right tools and technologies to help in such a task becomes a necessity for the buyer as for the seller.

Currently, little work has been performed to support customer decision making in E-commerce using natural language processing techniques. We identify principally two techniques. The first one is feature-based summarization that aims to identify the target entity (product, movie, hotel, restaurant, service) features and its corresponding opinions polarity (positive/negative), then, a feature-based summary of the reviews is generated [12,19,87]. While the second technique is called reputation generation, whose main focus is to produce an estimation value in which an entity is

[1] https://www.jd.com

[2] https://www.alibaba.com/

[3] https://www.amazon.com

held based on mining customer reviews expressed in natural languages [3, 24, 25, 95].

Previous studies on reputation generation have primarily focused on using semantic and sentiment analysis [3, 24, 25, 95], disregarding other useful information that could be extracted from user reviews, such as review helpfulness, which implies that reviews that receive higher votes from other users typically provide more valuable information, and review time, which implies that more recent reviews generally provide users with more up-to-date information.

An accurate and reliable reputation system should consider exploiting more online reviews' features such as review attached rating, review helpfulness, review time, and review sentiment orientation. For that reason, we propose a reputation system that incorporates all these attributes during the process of generating and visualizing reputation for various entities (movies, products, hotels, restaurants, and services). In this manner, this study addressed the following research question: with the consideration of review helpfulness, review time, review sentiment orientation probability and review attached rating, can the proposed reputation system offer better results in terms of reputation generation and visualization than the previous reputation systems (consider only semantic and sentiment relations)?

The contributions of this chapter are summarized as follows:

- Firstly, we propose a novel system that incorporates review time, review helpfulness, review sentiment orientation and review attached rating for the purpose of generating a numerical reputation value toward various entities (movies, products, hotels, restaurants, services, etc).

- Secondly, we propose a new comprehensive form to visualize reputation by showing numerical reputation value, opinion categories, top positive review, and top negative review in order to support customer decision making process in E-commerce (buying, renting, booking).

4.2 Problem definition

This section covers the necessary background for understanding the remainder of this chapter, including the problem definition.

In this chapter, we face the problem of generating reputation for movies, products, hotels, restaurants and services by aggregating review time, review helpfulness votes, review sentiment

orientation and review attached rating. Given a set of reviews $R_j = \{r_{1j},\ r_{2j},\ \ldots,\ r_{nj}\}$ expressed for an entity E_j, the set of their attached ratings $V_j = \{v_{1j},\ v_{2j},\ \ldots,\ v_{nj}\}$ where $v_{ij} \in [1,5]$ or $v_{ij} \in [1,10]$ depending on the rating system, the set of their attached helpfulness votes $RH_j = \{rh_{1j},\ rh_{2j},\ \ldots,\ rh_{nj}\}$ where $rh_{ij} \in \mathbb{N}^*$, the set of their posting time $RT_j = \{rt_{1j},\ rt_{2j},\ \ldots,\ rt_{nj}\}$ and the set of their sentiment orientation probabilities predicted by fine-tuned $BERT_{base}$ model $BERT_j = \{bert(r_{1j}),\ bert(r_{2j}),\ \ldots,\ bert(r_{nj})\}$ where $bert(r_{ij}) \in [0,1]^2$. The goal is to compute a review score for each review $RS_j = \{rs_{1j},\ rs_{2j},\ \ldots,\ rs_{nj}\}$ based on its helpfulness votes, its posting time and its sentiment orientation, and finally, to compute a reputation value Rep for an entity j by averaging the product of reviews score and reviews attached rating. Table 4.1 presents the descriptions of notations used in the rest of this chapter.

Table 4.1: Symbol denotation

Symbol	Description
R_j	The set of reviews expressed for the entity j
V_j	The set of ratings expressed for the entity j
RH_j	The set of reviews helpfulness votes expressed for the entity j
RT_j	The set of reviews posting time expressed for the entity j
$BERT_j$	The set of sentiment orientation probabilities predicted by fine_tuned $BERT_{Base}$ for reviews expressed for the entity j
RS_j	The set of reviews score expressed for the entity j
E_j	The target entity j
O_j	The total number of reviews expressed for the target entity j
Rep	The reputation value

4.3 Proposed approach

4.3.1 System overview

Our approach consists mainly on four steps:

- Firstly, we collect real data from websites that specialize in gathering customer reviews such

as IMDb[4], TripAdvisor[5], and Amazon[6] using web scraping tools, then, we preprocess them.

- Secondly, we assign three numerical scores to each review: helpfulness score, time score, and sentiment orientation score.
- Thirdly, we compute a review score based on the pre-computed scores (helpfulness score, time score, and sentiment orientation score).
- Finally, we generate a numerical reputation value toward the target entity (product, movie, hotel, restaurant, service, etc). Then, we propose a new form to visualize reputation by depicting numerical reputation value, opinion categories, positive review with the highest score, and negative review with the highest score.

Figure 4.1 describes the pipeline of our work.

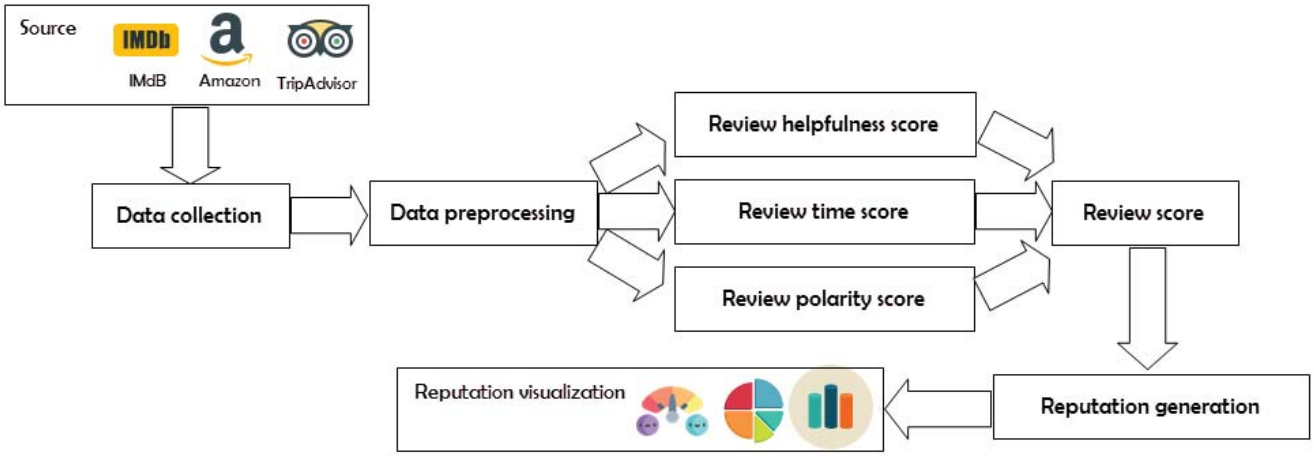


Figure 4.1: Proposed system pipeline

4.3.2 Data collection and preprocessing

Differently from previous studies on reputation generation, which mainly focus on extracting semantic and sentiment relations of reviews, our work incorporates other factors such as review helpfulness and review time. Hopefully, the majority of popular E-commerce websites such as TripAdvisor[7] and

[4] https://www.imdb.com/
[5] https://www.tripadvisor.com/
[6] https://www.amazon.com
[7] https://www.tripadvisor.com/

Amazon[8] gather online reviews with respect to the following structure: **textual review, review helpfulness votes, and review posting time**. Figure 4.2 describes the structure of the online reviews.

With the use of a web scraping tool, we have been able to collect raw data from some real data suppliers like Amazon, TripAdvisor, and IMDb.

After collecting all reviews, we applied some preprocessing techniques (lowercasing, tokenization, ...). Technical details of the data collection and preprocessing phase are described in section 4.4 Experiment results. subsection 4.4.1. Data collection and preprocessing.

4.3.3 Review helpfulness

The number of helpfulness votes attached to a review indicates how informative it is, which implies that reviews that receive higher votes from other users typically provide more valuable information. Thus, we design formula (4.1) to compute review helpfulness score.

$$H(r_{ij}) = \begin{cases} 0.75 & \text{if } rh_{ij} = 0 \text{ or } \frac{\log_{10}(rh_{ij})}{\log_{10}(N_j)} \leqslant 0.75 \\ \log_{N_j}(rh_{ij}) & \text{Otherwise} \end{cases} \tag{4.1}$$

We denote:

r_{ij} : Review number i expressed for the entity j.
$H(r_{ij})$: Helpfulness score of review r_{ij}.
rh_{ij} : The number of helpfulness votes attached to review r_{ij}.
N_j : The number of helpfulness votes attached to the most voted review toward the entity j.

The helpfulness score for a review ranges between 0.75 and 1 because we don't want to assign a low score to reviews with a small number of helpfulness votes.

We mention that $\frac{\log_{10}(rh_{ij})}{\log_{10}(N_j)} \leqslant 0.75$ means that $\log_{N_j}(rh_{ij}) \leqslant 0.75$ due to the fact that:
$\log_N(n) = \frac{\log_{base}(n)}{\log_{base}(N)} = \frac{\log_{10}(n)}{\log_{10}(N)}$

By applying equation (4.1), the most voted review will receive a review helpfulness score of 1

[8] `https://www.amazon.com`

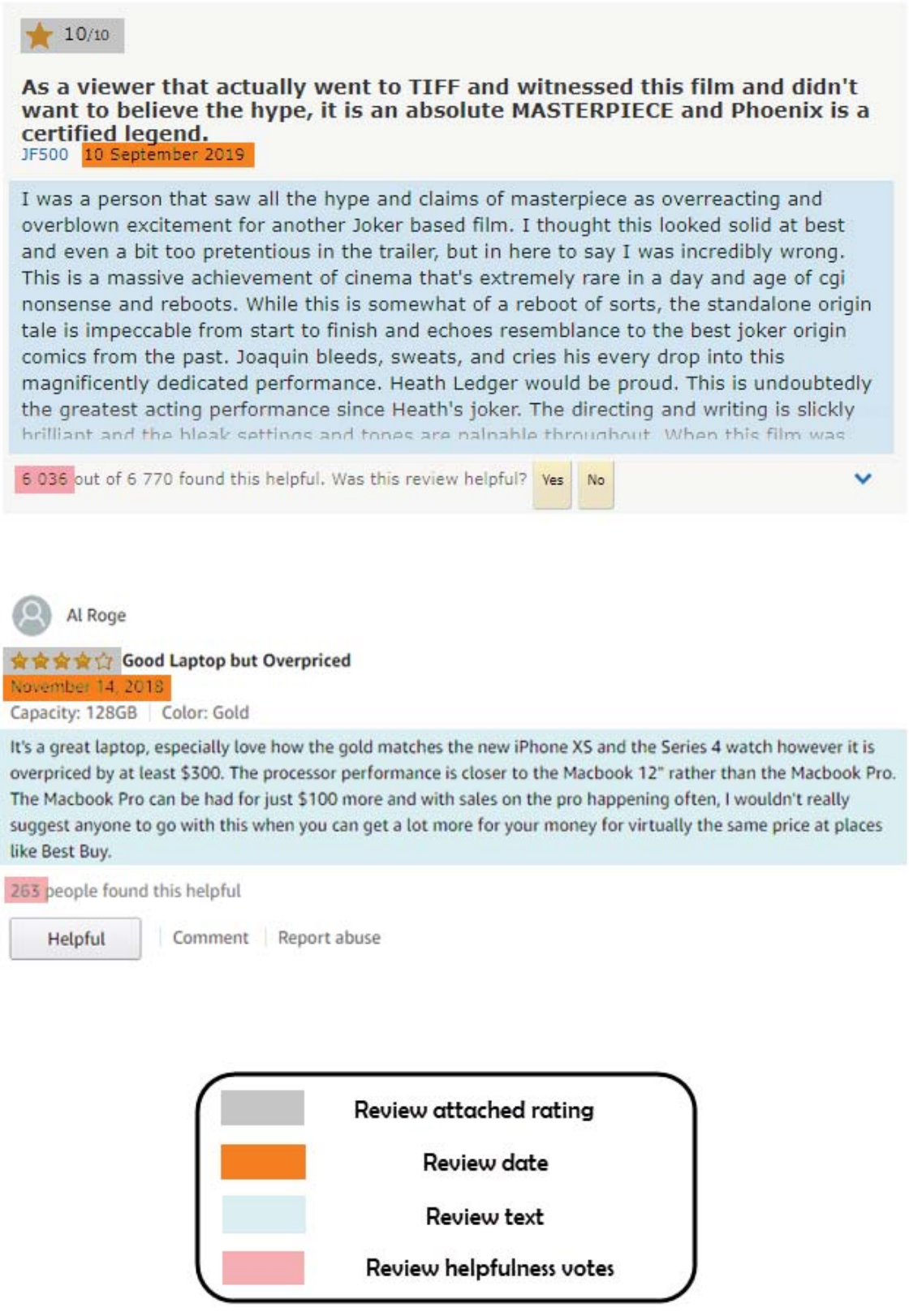


Figure 4.2: IMDb and Amazon reviews structure.

since $\log_{N_j}(N_j) = 1$. Also, reviews with high helpfulness votes will receive a high review helpfulness score and reviews with low helpfulness votes will receive a low review helpfulness score since for $x \in [1,N]$ and $y \in [1,N]$: $x \leq y$ implies that $log_N(x) \leq log_N(y)$. Algorithm 3 computes the helpfulness score for review r_{ij}.

Algorithm 3 Review helpfulness score

Define: $RH_j = \{rh_{1j},\ rh_{2j},\ \ldots,\ rh_{nj}\}$: The set of reviews' helpfulness votes expressed for the entity j.

Input: RH_j

Function `H`(rh_{ij})**:**

```
if rh_ij = 0 or log10(rh_ij) / log10(max(RH_j)) ⩽ 0.75 then
    H ⟵ 0.75
else
    H ⟵ log_{max(RH_j)}(rh_ij)
end if
return H
```

End Function

4.3.4 Review Time

Could you tell what would happen if we take a very well-reviewed gaming laptop from 10 years ago and put it on an online store? To answer this question, let us travel back in time to 20 years ago, where the gaming industry witnessed a great competition between gaming consoles, and where the enjoyment of a hardcore gaming experience was limited to that kind of tech. In that era, a gamer had to have a fat and heavy TV, with cables attached to a relatively big dedicated gaming console and its controllers in order to play a video game. All this bunch of materials and cables remain in one place in the house. Next, the industry shifted to computers, and then to mobile computers also known as laptops, which brought enough satisfaction to all the gaming consumers over the world. Although, laptops have been made much heavier than they should be, yet, it was very exciting to have the ability to enjoy your favorite games wherever you want just by packing your laptop on a backpack rather than having to be stuck in a room to play. Spatial freedom was a gift for the republic of players and so going mobile was their prior preference at that time.

Time goes forward, and so, the consumer preferences and choices. By today's standards, just going mobile is not good enough, gamers want lighter laptops, more performance,

high-end graphic cards, high resolution/fps screens, mechanical keyboard, and the list is long ... Today, gamers all over the planet become more demanding, their preferences changed drastically and so the industry does while trying to keep up with the human desires.

Back to our question, it is obvious that nobody will care about a 20-year-old gaming laptop, even if it was a best seller with 1 million 5-star reviews at that time. Why is that? simply because it becomes **obsolete** by the **modern user criteria**. Its 1 million review doesn't matter anymore. And so, **as all the things and beings**, **reviews** also have **an expiration date**. where they become irrelevant to the buyer.

Although, a product, laptop, movie or hotel may had very good reviews once, but time took off their power, their importance, and their effect over the judgment and decision of the consumer. At the end, **"you cannot beat time"**.

To conclude, we believe that more recent reviews generally provide users with more up-to-date information. Therefore, we design formula (4.2) to assign a time score to each review.

$$T(r_{ij}) = \begin{cases} 0.8 & \text{if } y - rt_{ij} \geqslant 100 \\ 1 - (y - rt_{ij}) \times 0.002 & \text{Otherwise} \end{cases} \tag{4.2}$$

We denote:

$T(r_{ij})$: Time score of review r_{ij}.
rt_{ij} : Publication year of review r_{ij}.
y : Current year.

The time score for a review ranges between 0.8 and 1, which implies that a higher time score is assigned to the most recent reviews.
Algorithm 4 computes the time score for review r_{ij}.

With the help of a film critic, we have been able to determine suitable minimum values for each of the review helpfulness and review time scores. Indeed, we have performed multiple experiments on a various range of movies, by trying different minimum values for both scores as parameters, where in each one among these, we compare the generated reputation value to the film critic's own rating regarding a given movie. Which leads to 0.75 and 0.8 to be chosen successively as

Algorithm 4 Review time score

Define: $RT_j = \{rt_{1j},\ rt_{2j},\ \ldots,\ rt_{nj}\}$: The set of reviews' posting time expressed for the entity j.

Input: RT_j

Function $\mathtt{T}(rt_{ij})$**:**

```
if currentYear − rt_ij ⩾ 100 then
    T ⟵ 0.8
else
    T ⟵ 1 − (currentYear − rt_ij) × 0.002
end if
return T
```

End Function

the fittest minimum values for review helpfulness and review time scores. Next, given the high accuracy achieved through our reputation system, the same last experiments have been done on other domains such as products, restaurants, and services, where we noticed very good results, particularly when using 0.75 and 0.8 as the minimum values for each of the scores.

4.3.5 Review sentiment orientation

We fine-tuned *BERT* model to determine the sentiment orientation probability of a target review due to the fact that it has achieved state-of-the-art results in a wide variety of natural language processing tasks by learning contextual relations between words or sub-words in a text. In this paper, we have an interest in assigning a sentiment orientation score to each review. Since fine-tuned BERT returns an array of 2 values: the probability of being negative and the probability of being positive (Softmax activation function), we apply the *max* function to the fine-tuned BERT output vector. The highest probability is kept as the sentiment orientation score of the target review.

$$S(r_{ij}) = max([P_{r_{ij}}^{negative}, P_{r_{ij}}^{positive}]) \tag{4.3}$$

We denote:

$S(r_{ij})$: Sentiment orientation score for review r_{ij}.

$P_{r_{ij}}^{negative}$: BERT model output prediction for review r_{ij} being negative.

$P_{r_{ij}}^{positive}$: BERT model output prediction for review r_{ij} being positive.

The sentiment polarity of a target review r_{ij} is predicted as negative if $P_{r_{ij}}^{negative} > P_{r_{ij}}^{positive}$ and predicted as positive if $P_{r_{ij}}^{negative} < P_{r_{ij}}^{positive}$.

Algorithm 5 computes the sentiment orientation score for review r_{ij}.

Algorithm 5 Review sentiment orientation score

Define: $R_j = \{r_{1j},\ r_{2j},\ \ldots,\ r_{nj}\}$: The set of reviews expressed for the entity j.

$BERT_j = \{bert(r_{1j}),\ bert(r_{2j}),\ \ldots,\ bert(r_{nj})\}$: The set of output vectors of fine-tuned $BERT_{Base}$ (the sentiment orientation probability of reviews expressed for the entity j.

Input: R_j

Function $\mathtt{S}(r_{ij})$**:**

$S \longleftarrow max(bert(r_{ij}))$

return S

End Function

4.3.6 Review score

Based on the above scores, we design formula (4.4) to compute a numerical score for each review:

$$RS(r_{ij}) = \frac{H(r_{ij}) + T(r_{ij}) + S(r_{ij})}{3} \tag{4.4}$$

We denote:

$RS(r_{ij})$: Review score for review r_{ij}.

$H(n_{ij})$: Helpfulness score of review r_{ij}.

$T(r_{ij})$: Time score of review r_{ij}.

$S(r_{ij})$: Sentiment orientation score for review r_{ij}.

Since review helpfulness score, review time score, and review sentiment orientation score range between 0 and 1, the generated review score is also between 0 and 1. Algorithm 6 computes the review score for all reviews.

Algorithm 6 Review score

Define : $R_j = \{r_{1j},\ r_{2j},\ \ldots,\ r_{nj}\}$: The set of reviews expressed for the entity j.

$RH_j = \{rh_{1j},\ rh_{2j},\ \ldots,\ rh_{nj}\}$: The set of reviews' helpfulness votes expressed for the entity j.

$RT_j = \{rt_{1j},\ rt_{2j},\ \ldots,\ rt_{nj}\}$: The set of reviews' posting time expressed for the entity j.

$RS_j = \{rs_{1j},\ rs_{2j},\ \ldots,\ rs_{nj}\}$: The set of reviews' score expressed for the entity j.

Input : R_j, RH_j, RT_j

Output: RS_j

for *i in range*(n) **do**

$rs_{ij} \longleftarrow (H(rh_{ij}) + T(rt_{ij}) + S(r_{ij}))/3$

end for

Table 4.2 represents example results of review score.

Table 4.2: Example results of review score

Review	Review helpfulness score	Review time score	Review sentiment orientation	Review sentiment score	Review score
Review 1	1	0.968	Positive	0.99732805	0.98844268
Review 2	0.75	0.982	Positive	0.99679191	0.9095973
Review 3	0.87468842	0.974	Negative	0.99608659	0.94825834
Review 4	0.91100877	0.964	Positive	0.9970323	0.95734702
Review 5	0.77448754	0.96	Negative	0.99694509	0.91047754

4.3.7 Reputation generation

We propose formula (4.5) to compute a single reputation value toward the target entity using review score $RS(r_{ij})$ and review attached rating v_{ij}:

$$Rep(E_j) = \frac{\sum_{i=1}^{O_j} RS(r_{ij}).v_{ij}}{O_j} \tag{4.5}$$

We denote:

E_j : Target entity j.

$Rep(E_j)$: Reputation value toward the target entity j.

$RS(r_{ij})$: Review score for review r_{ij}.

v_{ij} : Attached numerical rating to review r_{ij}.

O_j : Total number of reviews expressed for the target entity j.

The reputation value varies from 1 to 5 or 1 to 10 depending on the range of rating values. Algorithm 7 computes the reputation value toward a target item.

Algorithm 7 Reputation generation

Define : $V_j = \{v_{1j},\ v_{2j},\ \ldots,\ v_{nj}\}$: The set of ratings expressed for the entity j.

$RS_j = \{rs_{1j},\ rs_{2j},\ \ldots,\ rs_{nj}\}$: The set of reviews' score expressed for the entity j.

Rep: reputation value toward the target entity j.

Input : V_j, RS_j

Output: Rep

$temp \longleftarrow 0$

for i *in* $range(n)$ **do**

  $temp \longleftarrow temp + rs_{ij} \times v_{ij}$

end for

$Rep \longleftarrow temp/n$

Assuming that an entity E_j contains three reviews where $RH_j = \{100, 50, 1\}$, $RT_j = \{2020, 2010, 2000\}$, $BERT_j = \{0.998, 0.997, 0.996\}$ and $V_j = \{10, 10, 10\}$. By applying formula (4.1) and (4.2), we get the helpfulness and time scores: $H(r_{1j}) = 1$, $H(r_{2j}) = 0.849$, $H(r_{3j}) = 0.75$, $T(r_{1j}) = 1$, $T(r_{2j}) = 0.98$, and $T(r_{3j}) = 0.96$. After applying formula (4), we get the reviews scores: $RS_j = \{0.999, 0.942, 0.902\}$. In order to compute the reputation value toward E_j, we need to compute the product of rs_{ij} and v_{ij}. We get $rs_{1j}.v_{1j} = 9.99$, $rs_{2j}.v_{2j} = 9.42$, and $rs_{3j}.v_{3j} = 9.02$. Since $Rep(E_j) = \frac{\sum_{i=1}^{O_j} rs_{ij}.v_{ij}}{O_j}$, we can conclude that the first review r_{1j} has the highest impact (the highest product 9.99) on the reputation value of the entity E_j since it is very helpful and recent. In the contrary, the third review r_{3j} has the lowest impact (the lowest product 9.02). In fact, while it has the same attached rating as the first review, but, being both unhelpful and old made it by far less influential.

To conclude, recent and helpful reviews have more impact on the reputation value than old and unhelpful ones.

4.3.8 Reputation visualization

It is important to provide a potential customer or user with sufficient information for the purpose of assisting his decision. Thus, we propose a new way to visualize reputation by depicting the produced numerical reputation value toward the target entity, opinion categories, positive review with the highest review score (formula 4.4), and negative review with the highest review score (Figure 4.4).

4.4 Experiment results

4.4.1 Data collection and preprocessing

Five miscellaneous domains were addressed in our experiments, movie, TV show, product, hotel, and restaurant. We collected user reviews from IMDb[9], TripAdvisor[10], and Amazon[11] using a web scraping tool called ScrapeStorm[12]. We extracted 400 reviews for 4 movies, 400 reviews for 4 TV shows, 200 reviews for 2 products, 100 reviews for 1 hotel, and 100 reviews for 1 restaurant. Each extracted review contains: raw text, review time, review helpfulness votes, and review attached rating (Figure 4.2). The statistical information of the dataset is shown in Table 4.3.

Table 4.3: Statistical information of dataset

Domain	Number of entities	Number of reviews	Number of reviews per entity
Movie	4	400	100
TV show	4	400	100
Product	2	200	100
Hotel	1	100	100
Restaurant	1	100	100

After collecting the reviews, we:

[9] https://www.imdb.com/

[10] https://www.tripadvisor.com/

[11] https://www.amazon.com

[12] https://www.scrapestorm.com/

1. lowercase our text since we are using a BERT lowercase model

2. tokenize it: break words into WordPieces

3. add special "CLS" and "SEP" tokens

4. map our words to indexes using a vocab file that BERT provides

5. append "index" and "segment" tokens to each input

4.4.2 Sentiment analysis

We fine-tune $BERT_{Base}$ model to predict the sentiment orientation of the collected reviews. We build the model by creating a single new layer that will be trained with Large Movie Review Dataset v1.0 [97][13] which contains 25,000 positive and 25,000 negative processed movie reviews. We set the sequence length to 128, the batch size to 32, the learning rate to 0.00002, and the number of epochs to 3. Table 4.4 depicts the performance of fine-tuned BERT-base model on Large Movie Review Dataset v1.0.

Table 4.4: BERT-Base model classification result on Large Movie Review Dataset v1.0

	Precision	Recall	F1 score	Accuracy
BERT-Base	0.88048	0.89816	0.88923204	0.88812

We compared BERT-Base model to Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Bidirectional Long Short-Term Memory (BiLSTM). GloVe embeddings were used to train LSTM, BiLSTM, and CNN. Table 4.5 depicts the performance of fine-tuned BERT-base model, Vanilla CNN, Vanilla LSTM, and Vanilla BiLSTM on Large Movie Review Dataset v1.0.

We can see from Table 4.5 that BERT-Base model achieves the highest sentiment analysis accuracy compared to Vanilla CNN, Vanilla LSTM, and Vanilla BiLSTM.

We have mentioned in Chapter 1 some successful pre-trained models that achieve state-of-the-art results on Large Movie Review Dataset v1.0 such as XLNet-Large that achieves an accuracy

[13] `https://ai.stanford.edu/~amaas/data/sentiment/`

Table 4.5: Comparison results on Large Movie Review Dataset v1.0

Approach	Accuracy
Vanilla CNN	80.35
Vanilla LSTM	80.72
Vanilla BiLSTM	81.73
BERT base	**88.81**

of 96.21 and BERT-Large that achieves an accuracy of 95.79. However, these models require the combination of GPUs with plenty of computing power and a massive amount of memory.

We test BERT-Base model on our collected dataset. Figure 4.3 represents the accuracy of the model in predicting the sentiment orientation of the collected reviews.

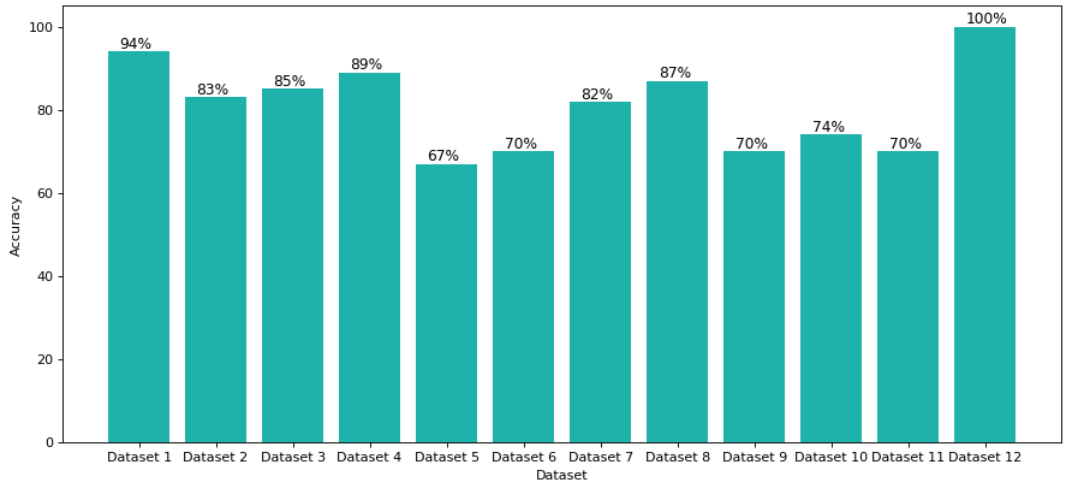


Figure 4.3: Accuracy of BERT-base model in predicting the sentiment orientation of the collected reviews.

We observe from Figure 4.3 that the model achieves good results in predicting the sentiment polarity of the extracted reviews. Even more impressively, the model performs well on datasets 9, 10, 11, and 12 that contain product, hotel, and restaurant reviews, even though it's trained with movie reviews.

4.4.3 Reputation visualization

We propose a new way to visualize reputation by depicting the produced numerical reputation value toward the target entity, opinion categories, top positive review, and top negative review.

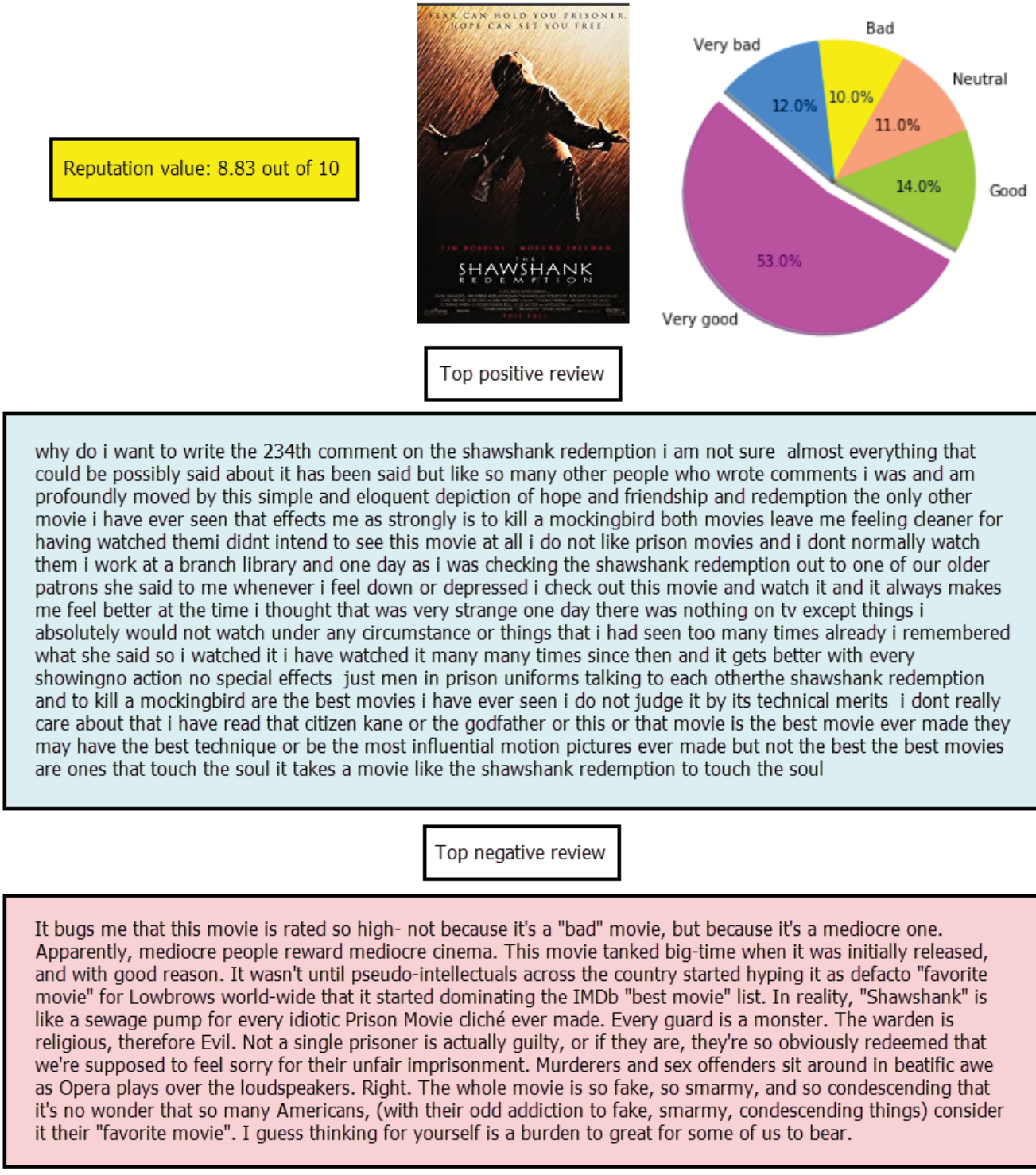


Figure 4.4: Reputation visualization.

As illustrated in Figure 4.4, our system provides users and potential customers with a comprehensive reputation visualization form that shows the numerical reputation value toward the target entity, opinion categories (very good, good, neutral, very bad, and bad) in a pie chart, top positive review (the positive review that holds the highest review score) and top negative review (the negative review that holds the highest review score).

Compared to previous studies on reputation generation [3, 24, 25], our proposed system is the only one that presents all of this helpful information in order to support users and customers during the decision making process in e-commerce websites. Table 4.6 shows comparison results between our system and previous reputation systems in terms of reputation visualization.

Table 4.6: Comparison results: reputation visualization

Work	Opinion categories	Top positive review	Top negative review
Yan et al. (2017) [3]	✓	✗	✗
Benlahbib & Nfaoui (2019) [24]	✗	✗	✗
Benlahbib & Nfaoui (2020) [25]	✓	✗	✗
This study	✓	✓	✓

4.4.4 System evaluation

Previous studies on reputation generation based on mining user and customer reviews expressed in natural language have mainly focused on exploiting semantic and sentiment relations between reviews to generate reputation values toward various entities. However, customer and user reviews contain a lot of other useful information that could be exploited during the reputation generation phase like review posting time and review helpfulness votes. Unfortunately, up-to-date, no work has incorporated review time, review helpfulness votes, and review sentiment polarity to generate reputation. Therefore, we propose a new reputation system that combines review posting time, review helpfulness votes, and review sentiment orientation in order to generate accurate and reliable reputation values toward different entities. Table 4.7 depicts the difference between previous reputation systems [3, 24, 25] and our proposed reputation system.

Since there are no standard evaluation metrics to assess the effectiveness and robustness of reputation systems, we conduct a user and expert survey as adopted in many research papers [98].

Table 4.7: Comparison results: review attributes exploited by recent reputation systems

Work	Semantic	Sentiment	Review helpfulness	Review time
Yan et al. (2017) [3]	✓	✗	✗	✗
Benlahbib & Nfaoui (2019) [24]	✓	✗	✗	✗
Benlahbib & Nfaoui (2020) [25]	✓	✓	✗	✗
This study	✓	✓	✓	✓

We have invited 32 users and 3 experts to rate four reputation generation systems: System 1 (our reputation system), system 2 [3], system 3 [24] and system 4 [25]. Each user and expert assigns a satisfaction score to each reputation system. The score ranges between 1 and 10.

The 32 users are from different backgrounds: 6 computer science PhD students, 2 math PhD students, an electrical engineer, an undergraduate student in mathematics, 2 computer science engineers, a physics teacher, 4 mathematics teachers, a research engineer in computer science, an electronic engineering student, an information systems engineer, a third-year student at the National School of Commerce and Management, a quality control technician, a sixth-year medical student, a housewife, 7 second-year medical students, and a software engineer.

Table 4.8 presents the average satisfaction score for each reputation system given by thirty-two users.

The formula of the average satisfaction score is: $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$ where $\{x_1, x_2, \ldots, x_N\}$ are the observed values of the sample items and N is the number of observations in the sample. The standard deviation is a measure of the amount of variation or dispersion of a set of values [99]. The formula for the standard deviation is: $\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$ where $\{x_1, x_2, \ldots, x_N\}$ are the observed values of the sample items, μ is the mean value of these observations, and N is the number of observations in the sample.

We can see from Table 4.8 that 31 users favor our reputation system over the three other systems in terms of helpfulness and effectiveness in generating reputation and visualization since it achieves the highest average satisfaction score and the lowest standard deviation of satisfaction scores. Moreover, only one user (user 19) favors system 2 [24]. System 2 takes second place by achieving an average satisfaction score of 7.83. System 4 [3] comes next with a 7.01 average satisfaction score, which sounds very reasonable since the main goal of system 2 was to improve system 4 by exploiting both sentiment and semantic analysis techniques. System 3 [24] takes the

Table 4.8: User satisfaction comparison

Systems	Ours	Benlahbib & Nfaoui [25]	Benlahbib & Nfaoui [24]	Yan et al. [3]
User 1	**9**	8.5	6	8
User 2	**9.5**	8	7.5	6.9
User 3	**9**	8.5	8	8
User 4	**9.5**	8	4.5	8
User 5	**8**	6.5	6	6.5
User 6	**9.5**	8	5.5	7
User 7	**9**	7	4	7
User 8	**9**	8	7	8
User 9	**9**	7	5	7
User 10	**9**	6	4	6
User 11	**9**	8	5.5	8
User 12	**9**	8	6	7
User 13	**8**	6.5	5	6.5
User 14	**8**	5	3	5
User 15	**9.5**	8.5	5.5	7
User 16	**8**	7	4	7
User 17	**9.5**	8	7	7.5
User 18	**9**	8.5	6	7.5
User 19	8	**9**	2	5
User 20	**8**	7	6.5	7
User 21	**9**	**9**	6	6
User 22	**9**	8	6	7.5
User 23	**10**	8	5	6
User 24	**9**	7	5	6
User 25	**10**	8.5	6	7
User 26	**10**	9	8	8.5
User 27	**10**	8	5	7
User 28	**9**	**9**	7	8
User 29	**9**	8	6	7
User 30	**9.75**	8	5	6.5
User 31	**9**	8	6	7
User 32	**10**	9	7	8
Average	**9.07**	7.83	5.625	7.01
Standard Deviation	**0.63**	0.93	1.32	0.84

last place by achieving an average satisfaction score of 5.625. System 3 doesn't provide users and customers with sufficient information to support their decision, since only providing them with reputation value alone isn't enough to help them make a judgment about a target item, indeed, the customers need more helpful information that could support them during the decision-making process such as opinion categories, top positive review and top negative review.

We enrich our experiment results by inviting 3 experts to rate each reputation system with a satisfaction score. Expert 1 is a former owner of an e-commerce website whose main field of interest is natural language processing and machine learning, while expert 2 is an active e-commerce buyer and seller with more than 9 years of experience. As for expert 3, he is a third-year PhD student in economics sciences. Table 4.9 presents the average satisfaction score for each reputation system given by the three experts.

Table 4.9: Expert satisfaction comparison

Systems	Ours	Benlahbib & Nfaoui [25]	Benlahbib & Nfaoui [24]	Yan et al. [3]
Expert 1	**10**	6	5	6
Expert 2	**8.5**	7.5	5	5.5
Expert 3	**9**	8	7	7
Average	**9.17**	7.17	5.67	6.17

Based on the average satisfaction score given by the three experts (Table 4.9), reputation system 1 takes first place with an average satisfaction score of 9.17, preceded by system 2, system 4, and system 3 comes in last place with 5.67 as average satisfaction score. Figure 4.5 combines the results of Table 4.8 and Table 4.9.

Figure 4.5 shows that both users and experts choose system 1 as the best in terms of reputation generation and visualization. System 2 holds second place, preceded by system 4. system 3 comes in fourth place.

We asked the three experts to share their opinions about system 1 strengths and weaknesses. Table 4.10 contains expert reviews toward system 1.

4.4.5 Further discussion

In summary, our reputation system exhibits the following advantages:

Table 4.10: Experts reviews

Expert	Review
Expert 1	The present work proposes a new method for reputation generation and visualization by incorporating helpfulness votes and time review features to sentiment and semantic features. In addition, the system outputs the item reputation value, top positive review, top negative review, and a pie chart that shows the distribution of sentiment over the reviews. Based on all these properties, i give a 10 to system 1. I think that this work can be enhanced by mining the top reviews according to the system's user. For instance, if two reviews have been selected as top positive reviews (the same reviews score), which review should the system display? By analyzing the users' data and behavior, the system will output the most accurate review with regard to user preferences.
Expert 2	I choose system 1 as the best because it covers many important criteria neglected by other systems. On the one hand, the numerical value and opinion categories alone only reveal the general impact of the film (product) on users (good film, nice film, bad film) and ignore the in-depth details that are essential for any kind of reputation. On the other hand, System 1 takes into account all four attributes of the evaluation, which will have a positive effect on the accuracy and credibility of the reputation value. I give it 8.5/10, because, in my opinion, the method still has some shortcomings since the best positive/negative reviews can contain spoilers for movies or bad personal experiences for a service, and this would create some confusion for the user.
Expert 3	I underline the importance of this study in terms of its usefulness and interest. It is an issue that seeks to put in place a system that allows subjective analysis, and thus scrutinizes and makes the opinions of viewers more concrete. And so, after having an idea of the different reputation systems, I consider the first system to be the most efficient in terms of reputation generation since it takes into consideration more variables/determinants such as review helpfulness votes and review time.

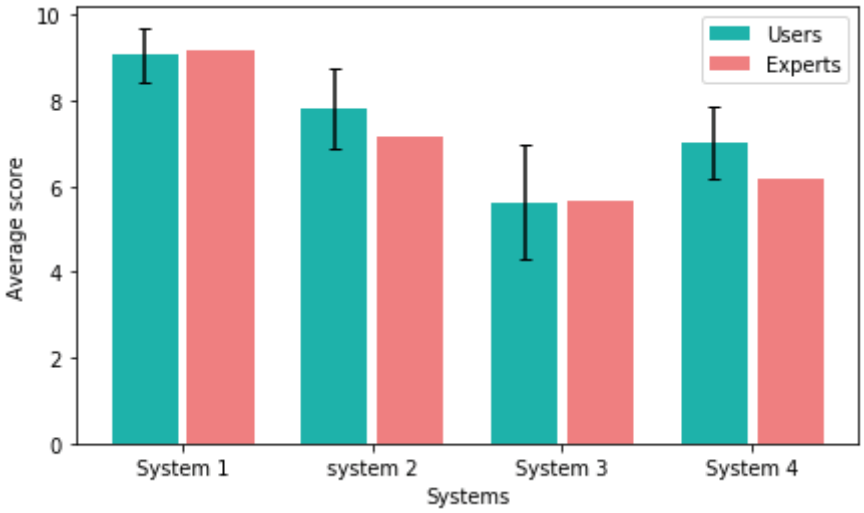


Figure 4.5: Users and experts average satisfaction score for each reputation system.

- **Accuracy**: The system incorporates review helpfulness, review time, review sentiment orientation probability, and review attached rating in order to generate an accurate reputation value.

- **Holistic**: The system proposes a new form of reputation visualization that depicts numerical reputation value, opinion categories, top positive review, and top negative review. The system also can output the top-k positive reviews and the top-k negative reviews. This new form of reputation visualization provides customers with sufficient information toward the target item in order to make an informed decision (buying, renting, booking) toward it.

- **Generality**: The system can be applied in any website that allows web users to (1) post their reviews expressed in natural languages, (2) share their numerical or star ratings, and (3) vote for helpful reviews. Furthermore, the system can be applied to various domains (products, movies, services, hotels).

- **Usefulness**: The system is very useful in terms of supporting web customers during the decision-making process in E-commerce by instantly providing them with sufficient information toward the target item, saving them from spending both their time and effort on reading thousands of online reviews.

However, our reputation system suffers from:

- **Safety**: Due to the openness of the Internet, many malicious users post fake reviews (false

positive/false negative) aiming to impact the popularity and credibility of online products. Therefore, our system should incorporate a filtering phase in order to detect and remove fake and irrelevant reviews.

4.5 Conclusion

In this chapter, we have proposed a reputation system that generates reputation toward various items (products, movies, TV shows, hotels, restaurants, services) by mining customer and user reviews expressed in natural language. The system incorporates four review attributes: review helpfulness, review time, review sentiment polarity, and review rating. The system also provides a comprehensive reputation visualization form by depicting the numerical reputation value, opinion group categories, top-k positive reviews, and top-k negative reviews. To better evaluate the effectiveness of our reputation system, 32 users and 3 experts were invited to assign a score of one (least satisfaction) to ten (highest satisfaction) to four reputation generation systems. Our reputation system achieved the highest average satisfaction score given by both users and experts. The three experts were also invited to share their point of view toward the proposed system in terms of reputation generation and visualization.

We believe that the proposed system represents an interesting online reputation system, full of fascinating insights into customer's decision-making process in e-commerce websites.
Future studies will focus on:

- exploiting further features including user credibility (prolific reviewers) and user's online behavior as suggested by expert 1 (Table 4.10).
- detecting and removing fake and irrelevant reviews by applying a filtering phase, and therefore reducing the processing time, and increasing the efficiency of the system at once, since only relevant and useful reviews will be taken into account.
- incorporating aspect-based opinion mining during the phase of reputation generation and visualization. As a result, the reputation visualization will be enhanced. Indeed, the system will depict more useful information toward the target entity E such as its features ($E_{featureX}$, $E_{featureY}$, $E_{featureZ}$ ...), the number of positive reviews toward feature $E_{featureX}$, and the number of negative reviews toward feature $E_{featureY}$ ...

The next chapter will describe AmazonRep, a reputation system that extends the system proposed in this chapter by exploiting review rating, review helpfulness votes, review time, review sentiment orientation, and user credibility to support Amazon's customer decision-making process.

Chapter 5

Reputation Generation and Visualization for Amazon's Products

5.1 Introduction

Over the last few years, e-commerce has been significantly growing and expanding [22,89,90,93,98]. According to [1,91,92,100,101], the number of reviews attached to an online entity (product, movie, hotel, service) may exceed thousands, which makes it impossible for a potential customer to read them all in order to make a decision (buying, booking, renting) toward the target item [102]. Thus, the need for the right tools and technologies to help in such a task becomes a necessity for customers.

During the last five years, few systems have been proposed to generate and visualize reputation by mining user and customer reviews expressed in natural language [3, 24–27, 95]. However, none of them has combined review time, review polarity, review helpfulness votes, and user credibility for the purpose of generating and visualizing reputation.

In this chapter, we propose AmazonRep, a reputation system to support Amazon's customer decision making process by incorporating review rating, review helpfulness votes, review time, review sentiment orientation, and user credibility. The system also provides a comprehensive reputation visualization form by depicting the numerical reputation value, opinion group categories, top-k positive reviews, and top-k negative reviews in order to help potential customers make an informed decision on whether to purchase the product.

5.2 Problem definition

This section covers the necessary background for understanding the remainder of this chapter, including the problem definition.

In this chapter, we face the problem of generating reputation for Amazon's products by aggregating review time, review helpfulness votes, user credibility, review sentiment orientation, and review attached rating. Given a set of reviews $R_j = \{r_{1j},\ r_{2j},\ \ldots,\ r_{nj}\}$ expressed for a product E_j, written by users $U = \{u_1,\ u_2,\ \ldots,\ u_n\}$, the set of their numerical ratings $V_j = \{v_{1j},\ v_{2j},\ \ldots,\ v_{nj}\}$ where $v_{ij} \in [1,5]$, the set of their attached helpfulness votes $RH_j = \{rh_{1j},\ rh_{2j},\ \ldots,\ rh_{nj}\}$ where $rh_{ij} \in \mathbb{N}^*$, the set of their posting time $RT_j = \{rt_{1j},\ rt_{2j},\ \ldots,\ rt_{nj}\}$, the set of the total number of helpful votes that users U have received $UH = \{h_1,\ h_2,\ \ldots,\ h_n\}$, and the set of the total number of reviews written by users U in Amazon $N = \{n_1,\ n_2,\ \ldots,\ n_n\}$. The goal is to compute a review score for each review $RS_j = \{rs_{1j},\ rs_{2j},\ \ldots,\ rs_{nj}\}$ based on its helpfulness votes, its posting time, and its user credibility, and finally, to compute a reputation value Rep for a product j by averaging the product of reviews score and reviews attached rating. Table 5.1 presents the descriptions of notations used in the rest of this chapter.

Table 5.1: Symbol denotation

Symbol	Description
R_j	The set of reviews expressed for the product j
V_j	The set of ratings expressed for the product j
U	The set of users that have expressed their opinions toward product j
RH_j	The set of reviews' helpfulness votes expressed for the product j
RT_j	The set of reviews' posting time expressed for the product j
RS_j	The set of reviews score expressed for the product j
UH	The set of the total number of helpful votes that users U have received
N	the set of the total number of reviews written by users U
E_j	The target product j
O_j	The total number of reviews expressed for the target product j
Rep	The reputation value

5.3 Proposed approach

5.3.1 System overview

Our approach consists mainly on four steps:

- Firstly, we collect 1300 product reviews from Amazon[1], and, we preprocess them.
- Secondly, we compute review helpfulness score, review time score, and user credibility score.
- Thirdly, we compute a review score based on the pre-computed scores (helpfulness score, time score, and user credibility score).
- Finally, we generate a numerical reputation value toward the target product. Then, we visualize reputation by depicting numerical reputation value, opinion categories, the top-k positive reviews with the highest scores, and the top-k negative reviews with the highest scores.

Figure 5.1 describes the pipeline of our reputation system.

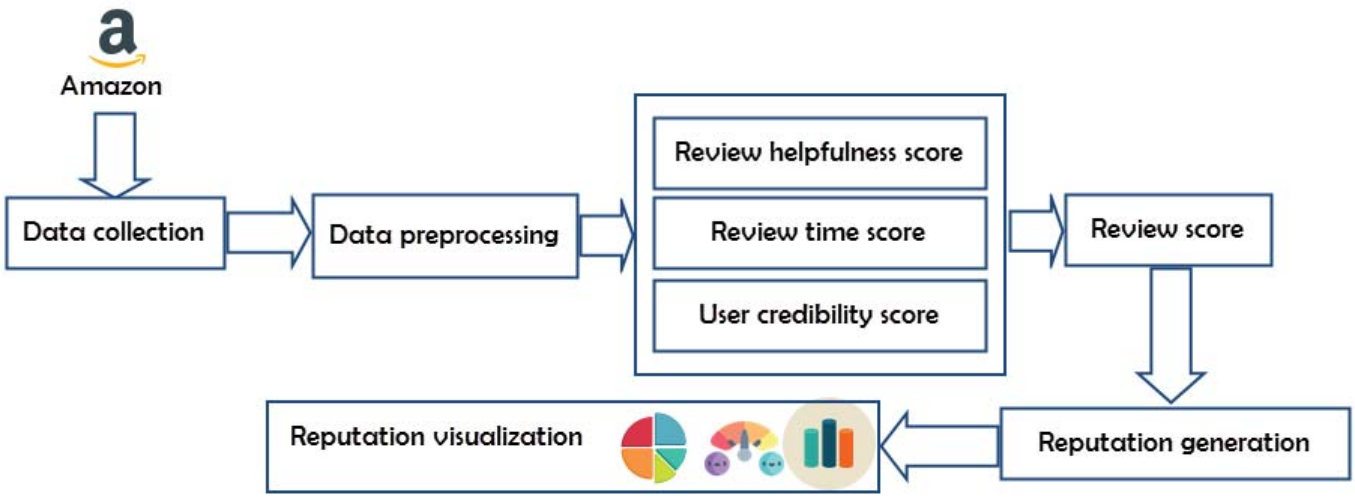


Figure 5.1: AmazonRep pipeline.

[1] https://www.amazon.com

5.3.2 Review structure

We gather online reviews from Amazon[2] website with respect to the following structure: **textual review, review helpfulness votes, review posting time, number of user's reviews, and number of user's helpful votes**. Figure 5.2 describes the structure of Amazon's reviews.

5.3.3 Review helpfulness

According to [26]: The number of helpfulness votes attached to a review indicates how informative it is, which implies that reviews that receive higher votes from other users typically provide more valuable information. Thus, we design equation (5.1) to assign a helpfulness score to each review.

$$H(r_{ij}) = \begin{cases} 0.8 & \text{if } rh_{ij} = 0 \text{ or } \frac{\log_{10}(rh_{ij})}{\log_{10}(N_j)} \leqslant 0.8 \\ \log_{N_j}(rh_{ij}) & \text{Otherwise} \end{cases} \tag{5.1}$$

We denote:

r_{ij} : Review number i expressed for the entity j.
$H(r_{ij})$: Helpfulness score of review r_{ij}.
rh_{ij} : The number of helpfulness votes attached to review r_{ij}.
N_j : The number of helpfulness votes attached to the most voted review toward the entity j.

The review helpfulness score is comprised between 0.8 and 1 because we don't want to assign a low score to reviews with a small number of helpfulness votes.

We mention that $\frac{\log_{10}(rh_{ij})}{\log_{10}(N_j)} \leqslant 0.8$ means that $\log_{N_j}(rh_{ij}) \leqslant 0.8$ due to the fact that:
$\log_N(n) = \frac{\log_{base}(n)}{\log_{base}(N)} = \frac{\log_{10}(n)}{\log_{10}(N)}$

By applying equation (5.1), the most voted review will receive a review helpfulness score of 1 since $\log_{N_j}(N_j) = 1$. Also, reviews with high helpfulness votes will receive a high review helpfulness score and reviews with low helpfulness votes will receive a low review helpfulness score since for $x \in [1,N]$ and $y \in [1,N]$: $x \leq y$ implies that $log_N(x) \leq log_N(y)$.

[2] `https://www.amazon.com`

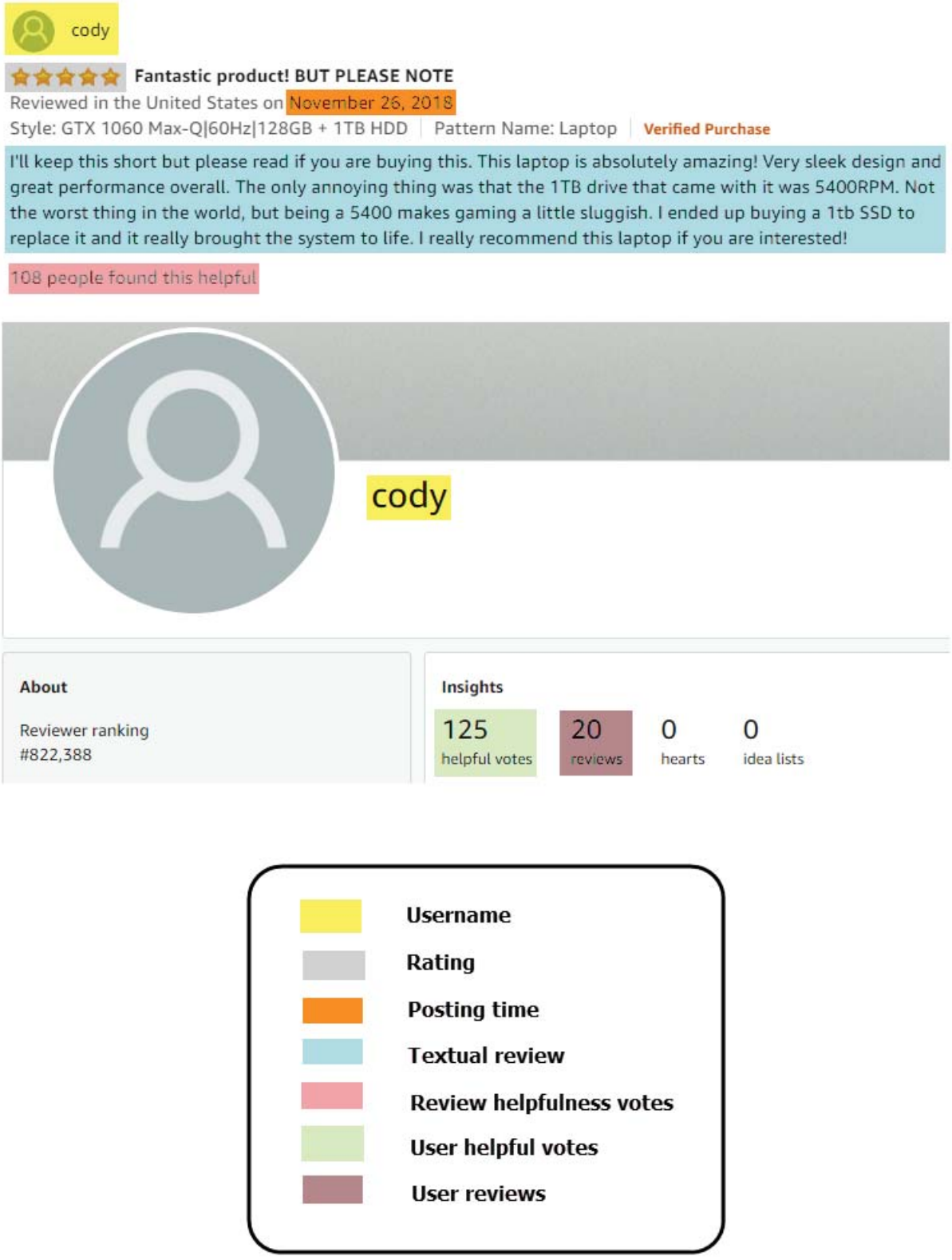


Figure 5.2: Amazon's review structure.

5.3.4 Review Time

More recent reviews generally provide users with more up-to-date information. Therefore, we design formula (5.2) to assign a time score to each review.

$$T(r_{ij}) = \begin{cases} 0.8 & \text{if } y - rt_{ij} \geqslant 100 \\ 1 - (y - rt_{ij}) \times 0.002 & \text{Otherwise} \end{cases} \tag{5.2}$$

We denote:

$T(r_{ij})$: Time score of review r_{ij}.
rt_{ij} : Publication year of review r_{ij}.
y : Current year.

The time score for a review ranges between 0.8 and 1, which implies that a higher time score is assigned to the most recent reviews.

5.3.5 User credibility

We propose equation (5.3) to compute a credibility score for users (customers) based on the number of their helpfulness votes and the number of their reviews.

$$C(u_i) = \frac{1}{1 + e^{-\frac{h_i}{n_i}}} \tag{5.3}$$

We denote:

$C(u_i)$: Credibility score of user u_i.
u_i : The author of review r_{ij}.
h_i : The number of helpful votes of user u_i.
n_i : The number of reviews written by user u_i.

The user credibility score is computed by applying the sigmoid function to the ratio of h_i to n_i where $sigmoid(x) = \sigma(x) = \frac{1}{1+e^{-x}}$. Figure 5.3 shows the plot of the sigmoid function.

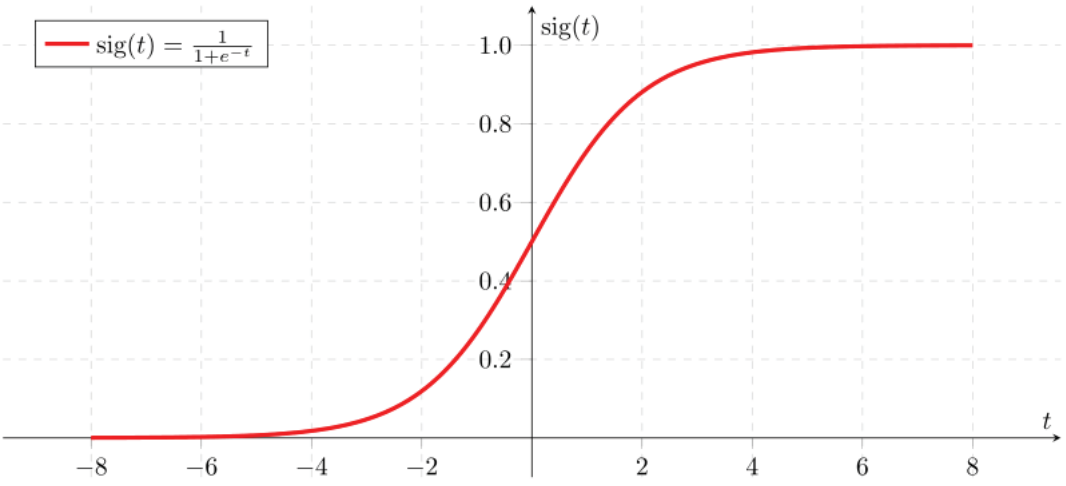


Figure 5.3: Plot of the sigmoid function.

From Figure 5.3, we can see that $0 \leq \sigma(x) \leq 1$ and $\sigma(0) = 0.5$.
Since $\frac{h_i}{n_i} \geq 0$, the user credibility score is ranging between 0.5 and 1.

5.3.6 Review score

Based on the above scores, we propose formula (5.4) to assign a numerical score to each review:

$$RS(r_{ij}) = \frac{\alpha_1 \times H(r_{ij}) + \alpha_2 \times T(r_{ij}) + \alpha_3 \times C(u_i)}{\alpha_1 + \alpha_2 + \alpha_3} \tag{5.4}$$

We denote:

$RS(r_{ij})$: Review score for review r_{ij}.
$H(r_{ij})$: Helpfulness score of review r_{ij}.
$T(r_{ij})$: Time score of review r_{ij}.
$C(u_i)$: Credibility score of user u_i.

α_1, α_2 and α_3 are constants that represent the impact of each score (review helpfulness score, review time score, and user credibility score) on the reputation value.

After conducting several experiments, we have found that AmazonRep generates accurate reputation values toward products when $\alpha_1 = 0.4$, $\alpha_2 = 0.35$, and $\alpha_3 = 0.25$. Consequently, we set $\alpha_1 = 0.4$, $\alpha_2 = 0.35$, and $\alpha_3 = 0.25$. Table 5.2 represents an example results of review score.

Table 5.2: Example results of review score

Review	Review helpfulness score	Review time score	User credibility score	Review sentiment orientation	Review score
Review 1	1	0.994	1	Negative	0.9979
Review 2	0.8	0.994	0.97861669	Negative	0.91255417
Review 3	0.8	0.994	0.99991620	Negative	0.91787905
Review 4	0.8	0.996	1	Positive	0.9186
Review 5	0.96668280	1	0.95257412	Negative	0.97481665

5.3.7 Reputation generation

We propose formula (5.5) to compute a single reputation value toward the target entity using review score $RS(r_{ij})$ and review attached rating v_{ij}:

$$Rep(E_j) = \frac{\sum_{i=1}^{O_j} RS(r_{ij}).v_{ij}}{O_j} \tag{5.5}$$

We denote:

E_j : Target entity j.

$Rep(E_j)$: Reputation value toward the target entity j.

$RS(r_{ij})$: Review score for review r_{ij}.

v_{ij} : Attached numerical rating to review r_{ij}.

O_j : Total number of reviews expressed for the target entity j.

The reputation value varies from 1 to 5 since Amazon website uses a five-star rating system.

5.3.8 Reputation visualization

Our system provides Amazon's customers with sufficient information toward the target product for the purpose of supporting them during the decision making process. The system depicts numerical reputation value, opinion categories, top positive review, and top negative review. Figure 5.4 shows an example of reputation visualization.

5.4 Experiment results

5.4.1 Data collection

We collected 1300 customer reviews for 13 products (1 book, 2 laptops, 3 smartphones, 2 washing machines, 2 refrigerators, 2 movies, and 1 video game) from Amazon[3]. The attributes related to the review (rating, posting time, helpfulness votes, and textual review) were collected using a web scraping tool called ScrapeStorm[4]. The attributes related to the user (number of user's reviews and number of user's helpful votes) were manually collected.

After collecting the reviews, we preprocess them by removing word segmentation and numbers. Also, we remove the commoner morphological and inflexional endings from words in English by applying Porter stemming algorithm [5].

5.4.2 Sentiment analysis

We train a Bidirectional Gated Recurrent Units (Bi-GRU) [65] with Large Movie Review Dataset v1.0 [97][6] in order to determine the sentiment orientation (positive/negative) of the collected reviews. We split the dataset into 45000 reviews for training and 5000 reviews for validation. We set the number of epochs to 5, the batch size to 512. We apply Adam optimizer [103] with a learning rate 0.001. Table 5.3 shows the Bi-GRU model classification result on the validation set.

[3] https://www.amazon.com

[4] https://www.scrapestorm.com/

[5] https://tartarus.org/martin/PorterStemmer/

[6] https://ai.stanford.edu/~amaas/data/sentiment/

Table 5.3: Bi-GRU model classification result on Large Movie Review Dataset v1.0

	Precision	Recall	F1 score	Accuracy
Bi-GRU	0.89	0.89	0.89	0.888

5.4.3 Reputation generation and visualization

Previous studies on reputation generation [3, 24–26, 95] have disregarded incorporating user credibility with other factors during the phase of reputation generation and visualization. Therefore, we propose AmazonRep that combines review time, review polarity, review helpfulness votes, and user credibility in order to generate and visualize reputation. Table 5.4 and Table 5.5 depict the comparison results between AmazonRep and five other reputation systems in terms of reputation generation and visualization.

Table 5.4: Comparison results: reputation visualization

Work	Opinion categories	Top positive review	Top negative review
Yan et al. (2017) [3]	✓	✗	✗
Benlahbib & Nfaoui (2019) [24]	✗	✗	✗
Benlahbib et al. (2019) [95]	✗	✗	✗
Benlahbib & Nfaoui (2020) [25]	✓	✗	✗
Benlahbib & Nfaoui (2020) [26]	✓	✓	✓
This study	✓	✓	✓

Table 5.5: Comparison results: review attributes exploited by recent reputation systems

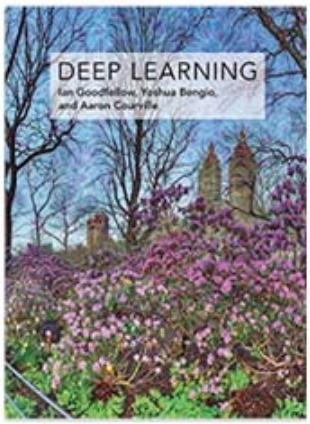
Work	Semantic	Sentiment	Review helpfulness	Review time	User credibility
Yan et al. (2017) [3]	✓	✗	✗	✗	✗
Benlahbib & Nfaoui (2020) [24]	✓	✗	✗	✗	✗
Benlahbib et al. (2019) [95]	✓	✓	✗	✗	✗
Benlahbib & Nfaoui (2020) [25]	✓	✓	✗	✗	✗
Benlahbib & Nfaoui (2020) [26]	✓	✓	✓	✓	✗
This study	✓	✓	✓	✓	✓

5.5 Conclusion

In this chapter, we have designed and built AmazonRep, a reputation system to support Amazon's customer decision-making process based on mining product reviews and their attributes. The proposed system combines review sentiment orientation, review helpfulness votes, review time, and user credibility in order to generate accurate reputation values toward Amazon's products. The system is very useful since it instantly provides customers with sufficient information toward the target product (numerical reputation value, opinion categories, top positive review, and top negative review), saving them from spending both their time and effort on reading thousands of online reviews.

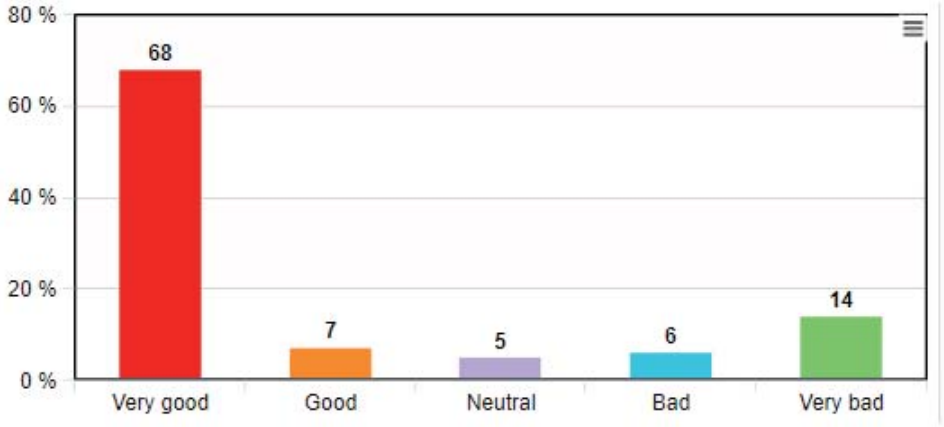
Future studies will focus on incorporating aspect-based opinion mining [104, 105] during the phase of reputation generation and visualization in order to improve the efficiency of the system.

Reputation value: 3.5 out of 5



Exploited features

* Review sentiment orientation
* Review helpfulness votes
* User credibility
* Review time



Top positive review

Very clear exposition, does the math without getting lost in the details. Although many of the concepts of the introductory first 100 pages can be found elsewhere, they are presented with remarkable cut-to-the-chase clarity.

Top negative review

I am surprised by how poorly written this book is. I eagerly bought it based on all the positive reviews it had received. Bad mistake. Only a few of the reviews clearly state the obvious problems of this book. Oddly enough, these informative reviews tend to attract aggressively negative comments of an almost personal nature. The disconnect between the majority of cloyingly effusive reviews of this book and the reality of how it is written is quite flabbergasting. I do not wish to speculate on the reason for this but it does sometimes does occur with a first book in an important area or when dealing with pioneer authors with a cult following.

Figure 5.4: Reputation visualization.

Conclusions

In this dissertation, I introduced four reputation systems that can automatically provide E-commerce customers with valuable information to support them during their online decision-making process by mining online reviews expressed in natural language.

The first chapter describes and examines previous research work done in the area of natural language processing (NLP) techniques for decision making in E-commerce, document-level sentiment analysis, and fine-grained sentiment analysis. The chapter also covers the necessary background for understanding Bidirectional Encoder Representations from Transformers (BERT) model since we employed it to determine the sentiment orientation of customer and user reviews.

Chapter 2 presents a reputation system that incorporates sentiment analysis, semantic analysis, and opinion fusion to generate accurate reputation values toward online items.

The next chapter describes MTVRep, a movie and TV show reputation system that exploits fine-grained sentiment analysis and semantic analysis for the purpose of generating and visualizing reputation toward movies and TV shows.

Chapter 4 presents a reputation system that incorporates four review attributes: review helpfulness, review time, review sentiment polarity, and review rating in order to generate reputation toward various online items (products, movies, TV shows, hotels, restaurants, services). The system also provides a comprehensive reputation visualization form to help a potential customer make an informed decision by depicting the numerical reputation value, opinion group categories, top-k positive reviews, and top-k negative reviews.

Chapter 5 describes AmazonRep, a reputation system that extends the system proposed in chapter 4 by exploiting review rating, review helpfulness votes, review time, review sentiment

orientation, and user credibility for the purpose of supporting Amazon's customer decision making process.

Future studies will focus on:

- exploiting further features including user's online behavior.
- detecting and removing fake and irrelevant reviews by applying a filtering phase, and therefore reducing the processing time, and increasing the efficiency of the system at once, since only relevant and useful reviews will be taken into account.
- incorporating aspect-based opinion mining during the phase of reputation generation and visualization. As a result, the reputation visualization will be enhanced. Indeed, the system will depict more useful information toward the target entity E such as its features ($E_{featureX}$, $E_{featureY}$, $E_{featureZ}$ ...), the number of positive reviews toward feature $E_{featureX}$, and the number of negative reviews toward feature $E_{featureY}$ ...

Bibliography

[1] A. Kangale, S. K. Kumar, M. A. Naeem, M. Williams, and M. K. Tiwari, "Mining consumer reviews to generate ratings of different product attributes while producing feature-based review-summary," *International Journal of Systems Science*, vol. 47, no. 13, pp. 3272–3286, 2016. [Online]. Available: https://doi.org/10.1080/00207721.2015.1116640

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[3] Z. Yan, X. Jing, and W. Pedrycz, "Fusing and mining opinions for reputation generation," *Inf. Fusion*, vol. 36, no. C, p. 172–184, Jul. 2017. [Online]. Available: https://doi.org/10.1016/j.inffus.2016.11.011

[4] R. Paul and Z. Richard, *Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system*. Emerald Group Publishing Limited, 2002.

[5] J. Schneider, G. Kortuem, J. Jager, S. Fickas, and Z. Segall, "Disseminating trust information in wearable communities," *Personal Ubiquitous Comput.*, vol. 4, no. 4, pp. 245–248, Jan. 2000. [Online]. Available: http://dx.doi.org/10.1007/PL00000012

[6] F. Garcin, B. Faltings, and R. Jurca, "Aggregating reputation feedback," in *Proceedings of the First International Conference on Reputation: Theory and Technology*, vol. 1, no. CONF, 2009, pp. 62–74.

[7] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[8] C. S. Leberknight, S. Sen, and M. Chiang, "On the volatility of online ratings: An empirical study," in *E-Life: Web-Enabled Convergence of Commerce, Work, and Social Life*, M. J. Shaw, D. Zhang, and W. T. Yue, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 77–86.

[9] J. Cho, K. Kwon, and Y. Park, "Q-rater: A collaborative reputation system based on source credibility theory," *Expert Systems with Applications*, vol. 36, no. 2, Part 2, pp. 3751 – 3760, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417408001486

[10] A. Abdel-Hafez, Y. Xu, and D. Tjondronegoro, "Product reputation model: An opinion mining based approach," in *Proceedings of the 1st International Workshop on Sentiment Discovery from Affective Data [CEUR Workshop Proceedings Volume 917]*, M. Cocea, C. Labbe, S. Weibelzahl, E. Menasalvas, and M. M. Gaber, Eds. http://ceur-ws.org/: CEUR Workshop Proceedings / Sun SITE Central Europe (CEUR), 2012, pp. 16–27. [Online]. Available: https://eprints.qut.edu.au/58118/

[11] N. Makadia, "Feature-based opinion summarization: A survey," *International Journal of Engineering Development and Research*, vol. 8, no. 2, 2016.

[12] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '04. New York, NY, USA: ACM, 2004, pp. 168–177. [Online]. Available: http://doi.acm.org/10.1145/1014052.1014073

[13] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*, ser. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, pp. 487–499. [Online]. Available: http://dl.acm.org/citation.cfm?id=645920.672836

[14] G. A. Miller, "Wordnet: A lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995. [Online]. Available: http://doi.acm.org/10.1145/219717.219748

[15] A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, ser. HLT '05. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 339–346. [Online]. Available: https://doi.org/10.3115/1220575.1220618

[16] L. Garcia-Moya, H. Anaya-Sánchez, and R. Berlanga-Llavori, "Retrieving product features and opinions from customer reviews," *IEEE Intelligent Systems*, vol. 28, no. 3, pp. 19–27, 2013.

[17] A. Abbasi, S. France, Z. Zhang, and H. Chen, "Selecting attributes for sentiment classification using feature relation networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 3, pp. 447–462, 2010.

[18] X. Chi, T. P. Siew, and E. Cambria, "Adaptive two-stage feature selection for sentiment classification," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2017, pp. 1238–1243.

[19] L. Zhuang, F. Jing, and X.-Y. Zhu, "Movie review mining and summarization," in *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, ser. CIKM '06. New York, NY, USA: ACM, 2006, pp. 43–50. [Online]. Available: http://doi.acm.org/10.1145/1183614.1183625

[20] S. Homoceanu, M. Loster, C. Lofi, and W. Balke, "Will i like it? providing product overviews based on opinion excerpts," in *2011 IEEE 13th Conference on Commerce and Enterprise Computing*, Sep. 2011, pp. 26–33.

[21] S. Blair-Goldensohn, K. Hannan, R. McDonald, T. Neylon, G. Reis, and J. Reynar, "Building a sentiment summarizer for local service reviews," in *WWW Workshop on NLP Challenges in the Information Explosion Era (NLPIX)*, 2008.

[22] Y.-H. Hu, Y.-L. Chen, and H.-L. Chou, "Opinion mining from online hotel reviews a text summarization approach," *Inf. Process. Manage.*, vol. 53, no. 2, p. 436–449, Mar. 2017. [Online]. Available: https://doi.org/10.1016/j.ipm.2016.12.002

[23] A. M. El-Halees and D. Salah, "Feature-based opinion summarization for arabic reviews," in *2018 International Arab Conference on Information Technology (ACIT)*, Nov 2018, pp. 1–5.

[24] A. Benlahbib and E. H. Nfaoui, "An unsupervised approach for reputation generation," *Procedia computer science*, vol. 148, pp. 80–86, 2019. [Online]. Available: https://doi.org/10.1016/j.procs.2019.01.011

[25] A. Benlahbib and E.-H. Nfaoui, "A hybrid approach for generating reputation based on opinions fusion and sentiment analysis," *Journal of Organizational Computing and Electronic Commerce*, vol. 30, no. 1, pp. 9–27, 2020. [Online]. Available: https://doi.org/10.1080/10919392.2019.165435

[26] A. Benlahbib and E. H. Nfaoui, "Aggregating customer review attributes for online reputation generation," *IEEE Access*, vol. 8, pp. 96 550–96 564, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2996805

[27] A. Benlahbib and E. Nfaoui, "Mtvrep: A movie and tv show reputation system based on fine-grained sentiment and semantic analysis," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 2, pp. 613–1626, 2021. [Online]. Available: http://dx.doi.org/110.11591/ijece.v11i2.pp1613-1626

[28] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 55–60. [Online]. Available: https://www.aclweb.org/anthology/P14-5010

[29] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana:

Association for Computational Linguistics, Jun. 2018, pp. 2227–2237. [Online]. Available: https://www.aclweb.org/anthology/N18-1202

[30] O. Ahlgren, "Research on sentiment analysis: The first decade," in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, Dec 2016, pp. 890–899.

[31] S. Sun, C. Luo, and J. Chen, "A review of natural language processing techniques for opinion mining systems," *Inf. Fusion*, vol. 36, no. C, pp. 10–25, Jul. 2017. [Online]. Available: https://doi.org/10.1016/j.inffus.2016.10.004

[32] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, ser. EMNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 79–86. [Online]. Available: https://doi.org/10.3115/1118693.1118704

[33] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep 1995. [Online]. Available: https://doi.org/10.1007/BF00994018

[34] M. E. Maron, "Automatic indexing: An experimental inquiry," *J. ACM*, vol. 8, no. 3, pp. 404–417, Jul. 1961. [Online]. Available: http://doi.acm.org/10.1145/321075.321084

[35] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, "A maximum entropy approach to natural language processing," *Comput. Linguist.*, vol. 22, no. 1, pp. 39–71, Mar. 1996. [Online]. Available: http://dl.acm.org/citation.cfm?id=234285.234289

[36] A. Kennedy and D. Inkpen, "Sentiment classification of movie reviews using contextual valence shifters," *Computational Intelligence*, vol. 22, no. 2, pp. 110–125, 2006. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8640.2006.00277.x

[37] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, Barcelona, Spain, Jul. 2004, pp. 271–278. [Online]. Available: https://www.aclweb.org/anthology/P04-1035

[38] M. Koppel and J. Schler, "The importance of neutral examples for learning sentiment," *Computational Intelligence*, vol. 22, pp. 100–116, May

2006. [Online]. Available: https://www.microsoft.com/en-us/research/publication/the-importance-of-neutral-examples-for-learning-sentiment/

[39] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[40] X. Yan and X. G. Su, *Linear Regression Analysis*. WORLD SCIENTIFIC, 2009. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/6986

[41] E. Frank and M. Hall, "A simple approach to ordinal classification," in *Proceedings of the 12th European Conference on Machine Learning*, ser. EMCL '01. London, UK, UK: Springer-Verlag, 2001, pp. 145–156. [Online]. Available: http://dl.acm.org/citation.cfm?id=645328.649997

[42] S. Wang and C. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 90–94. [Online]. Available: https://www.aclweb.org/anthology/P12-2018

[43] R. Jing, Y. Yu, and Z. Lin, "How service-related factors affect the survival of b2t providers: A sentiment analysis approach," *Journal of Organizational Computing and Electronic Commerce*, vol. 25, no. 3, pp. 316–336, 2015.

[44] Ł. Augustyniak, P. Szymański, T. Kajdanowicz, and W. Tuligłowicz, "Comprehensive study on lexicon-based ensemble classification sentiment analysis," *Entropy*, vol. 18, no. 1, p. 4, 2016.

[45] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[46] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

[47] Y. Freund, R. E. Schapire *et al.*, "Experiments with a new boosting algorithm," in *icml*, vol. 96. Citeseer, 1996, pp. 148–156.

[48] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.

[49] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proceedings of the 7th ACM Conference on Recommender Systems*, ser. RecSys '13. New York, NY, USA: ACM, 2013, pp. 165–172. [Online]. Available: http://doi.acm.org/10.1145/2507157.2507163

[50] P. D. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 417–424. [Online]. Available: https://doi.org/10.3115/1073083.1073153

[51] P. D. Turney, "Mining the web for synonyms: Pmi-ir versus lsa on toefl," in *Proceedings of the 12th European Conference on Machine Learning*, ser. EMCL '01. London, UK, UK: Springer-Verlag, 2001, pp. 491–502. [Online]. Available: http://dl.acm.org/citation.cfm?id=645328.650004

[52] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, 2011. [Online]. Available: https://www.aclweb.org/anthology/J11-2001

[53] S. Vashishtha and S. Susan, "Fuzzy rule based unsupervised sentiment analysis from social media posts," *Expert Systems with Applications*, vol. 138, p. 112834, 2019.

[54] M. Fernández-Gavilanes, T. Álvarez-López, J. Juncal-Martínez, E. Costa-Montenegro, and F. J. González-Castaño, "Unsupervised method for sentiment analysis in online texts," *Expert Systems with Applications*, vol. 58, pp. 57–75, 2016.

[55] R. Moraes, J. F. Valiati, and W. P. GaviaO Neto, "Document-level sentiment classification: An empirical comparison between svm and ann," *Expert Syst. Appl.*, vol. 40, no. 2, pp. 621–633, Feb. 2013. [Online]. Available: https://doi.org/10.1016/j.eswa.2012.07.059

[56] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ser. ICML'14. JMLR.org, 2014, pp. II–1188–II–1196. [Online]. Available: http://dl.acm.org/citation.cfm?id=3044805.3045025

[57] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13. USA: Curran Associates Inc., 2013, pp. 3111–3119. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999792.2999959

[58] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, May–Jun. 2015, pp. 103–112. [Online]. Available: https://www.aclweb.org/anthology/N15-1011

[59] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[60] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[61] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford University Press, Inc., 1995.

[62] G. Mesnil, T. Mikolov, M. Ranzato, and Y. Bengio, "Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews," 2014.

[63] K. Baktha and B. K. Tripathy, "Investigation of recurrent neural networks in the field of sentiment analysis," in *2017 International Conference on Communication and Signal Processing (ICCSP)*, April 2017, pp. 2047–2050.

[64] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[65] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural*

Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. [Online]. Available: https://www.aclweb.org/anthology/D14-1179

[66] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation," *arXiv preprint arXiv:1904.12848*, 2019.

[67] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Ro{bert}a: A robustly optimized {bert} pretraining approach," 2020. [Online]. Available: https://openreview.net/forum?id=SyxS0T4tvS

[68] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 5753–5763. [Online]. Available: http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding.pdf

[69] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=H1eA7AEtvS

[70] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:1910.10683*, 2019.

[71] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=r1xMH1BtvB

[72] P. Xu, A. Madotto, C.-S. Wu, J. H. Park, and P. Fung, "Emo2Vec: Learning generalized emotion representation by multi-task training," in *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 292–298. [Online]. Available: https://www.aclweb.org/anthology/W18-6243

[73] J. Mu, S. Bhat, and P. Viswanath, "All-but-the-top: Simple and effective postprocessing for word representations," *arXiv preprint arXiv:1702.01417*, 2017.

[74] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1631–1642. [Online]. Available: https://www.aclweb.org/anthology/D13-1170

[75] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, "Sentiment analysis by capsules," in *Proceedings of the 2018 World Wide Web Conference*, ser. WWW '18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018, p. 1165–1174. [Online]. Available: https://doi.org/10.1145/3178876.3186015

[76] Y. Yang, "Convolutional neural networks with recurrent neural filters," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 912–917. [Online]. Available: https://www.aclweb.org/anthology/D18-1109

[77] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," in *Advances in Neural Information Processing Systems*, 2017, pp. 6294–6305.

[78] M. Munikar, S. Shakya, and A. Shrestha, "Fine-grained sentiment classification using bert," in *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, vol. 1. IEEE, 2019, pp. 1–5.

[79] Z. Sun, C. Fan, Q. Han, X. Sun, Y. Meng, F. Wu, and J. Li, "Self-explaining structures improve nlp models," 2020.

[80] S. Brahma, "Improved sentence modeling using suffix bidirectional lstm," *arXiv preprint arXiv:1805.07340*, 2018.

[81] D. Shen, G. Wang, W. Wang, M. R. Min, Q. Su, Y. Zhang, C. Li, R. Henao, and L. Carin, "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms," *arXiv preprint arXiv:1805.09843*, 2018.

[82] R. Farmer and B. Glass, *Building Web Reputation Systems*, 1st ed. USA: Yahoo! Press, 2010.

[83] Z. Yan, *Trust Management in Mobile Environments: Autonomic and Usable Models*, 1st ed. Hershey, PA, USA: IGI Global, 2013.

[84] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decis. Support Syst.*, vol. 43, no. 2, pp. 618–644, Mar. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.dss.2005.05.019

[85] J. Shawe-Taylor and S. Sun, "A review of optimization methodologies in support vector machines," *Neurocomputing*, vol. 74, no. 17, pp. 3609 – 3618, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231211004371

[86] J.-z. Wang, Z. Yan, L. T. Yang, and B.-x. Huang, "An approach to rank reviews by fusing and mining opinions based on review pertinence," *Inf. Fusion*, vol. 23, no. C, pp. 3–15, May 2015. [Online]. Available: http://dx.doi.org/10.1016/j.inffus.2014.04.002

[87] K. Bafna and D. Toshniwal, "Feature based summarization of customers' reviews of online products," *Procedia Computer Science*, vol. 22, pp. 142–151, 2013.

[88] L. Zhuang, F. Jing, and X.-Y. Zhu, "Movie review mining and summarization," in *Proceedings of the 15th ACM international conference on Information and knowledge management*. ACM, 2006, pp. 43–50.

[89] T. Hou, B. Yannou, Y. Leroy, and E. Poirson, "Mining customer product reviews for product development: A summarization process," *Expert Systems with Applications*, vol. 132, pp. 141–150, 2019.

[90] M. Hu and B. Liu, "Mining opinion features in customer reviews," in *Proceedings of the 19th National Conference on Artifical Intelligence*, ser. AAAI'04. AAAI Press, 2004, p. 755–760.

[91] J. Lovinger, I. Valova, and C. Clough, "Gist: General integrated summarization of text and reviews," *Soft Comput.*, vol. 23, no. 5, p. 1589–1601, Mar. 2019. [Online]. Available: https://doi.org/10.1007/s00500-017-2882-2

[92] S. Pecar, "Towards opinion summarization of customer reviews," in *Proceedings of ACL 2018, Student Research Workshop*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1–8. [Online]. Available: https://www.aclweb.org/anthology/P18-3001

[93] K. Zhang, R. Narayanan, and A. Choudhary, "Voice of the customers: Mining online customer reviews for product feature-based ranking," in *Proceedings of the 3rd Wonference on Online Social Networks*, ser. WOSN'10. USA: USENIX Association, 2010, p. 11.

[94] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 168–177. [Online]. Available: https://doi.org/10.1145/1014052.1014073

[95] A. Benlahbib, A. Boumhidi, and E. H. Nfaoui, "A logistic regression approach for generating movies reputation based on mining user reviews," in *2019 International Conference on Intelligent Systems and Advanced Computing Sciences (ISACS)*. IEEE, 2019, pp. 1–7. [Online]. Available: https://doi.org/10.1109/ISACS48493.2019.9068916

[96] Y. Wang, Z. Zhou, S. Jin, D. Liu, and M. Lu, "Comparisons and selections of features and classifiers for short text classification," in *IOP Conference Series: Materials Science and Engineering*, vol. 261, no. 1. IOP Publishing, 2017, p. 012018.

[97] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[98] D. Wang, S. Zhu, and T. Li, "Sumview: A web-based engine for summarizing product reviews and customer opinions," *Expert Syst. Appl.*, vol. 40, no. 1, p. 27–33, Jan. 2013. [Online]. Available: https://doi.org/10.1016/j.eswa.2012.05.070

[99] J. M. Bland and D. G. Altman, "Statistics notes: Measurement error," *BMJ*, vol. 312, no. 7047, p. 1654, 1996. [Online]. Available: https://www.bmj.com/content/312/7047/1654

[100] K. Amarouche, H. Benbrahim, and I. Kassou, "Customer product review summarization over time for competitive intelligence," *Journal of Automation Mobile Robotics and Intelligent Systems*, vol. Vol. 12, No. 4, p. 70–82, 2018.

[101] L. García-Moya, H. Anaya-Sánchez, and R. Berlanga-Llavori, "Retrieving product features and opinions from customer reviews," *IEEE Intelligent Systems*, vol. 28, no. 3, pp. 19–27, 2013. [Online]. Available: https://doi.org/10.1109/MIS.2013.37

[102] M. Coavoux, H. Elsahar, and M. Gallé, "Unsupervised aspect-based multi-document abstractive summarization," in *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 42–47. [Online]. Available: https://www.aclweb.org/anthology/D19-5405

[103] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[104] J. Zhu, H. Wang, M. Zhu, B. K. Tsou, and M. Ma, "Aspect-based opinion polling from customer reviews," *IEEE Transactions on Affective Computing*, vol. 2, no. 1, pp. 37–49, 2011. [Online]. Available: https://doi.org/10.1109/T-AFFC.2011.2

[105] A.-D. Vo, Q.-P. Nguyen, and C.-Y. Ock, "Automatic knowledge extraction for aspect-based sentiment analysis of customer reviews," in *Proceedings of the 10th International Conference on Computer Modeling and Simulation*, ser. ICCMS 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 110–113. [Online]. Available: https://doi.org/10.1145/3177457.3177462