O IMPACTO DO RUÍDO DE ATRIBUTO NA CLASSIFICAÇÃO DA POLARIDADE DE CRÍTICAS DE FILMES

KAREN STÉFANY MARTINS

O IMPACTO DO RUÍDO DE ATRIBUTO NA CLASSIFICAÇÃO DA POLARIDADE DE CRÍTICAS DE FILMES

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Pedro Olmo Stancioli Vaz de Melo
Coorientador: Rodrygo Luis Teodoro Santos

Belo Horizonte, MG
Outubro de 2020

KAREN STÉFANY MARTINS

ON THE IMPACT OF ATTRIBUTE NOISE ON MOVIE REVIEW POLARITY CLASSIFICATION

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Pedro Olmo Stancioli Vaz de Melo
Co-Advisor: Rodrygo Luis Teodoro Santos

Belo Horizonte, MG
October 2020

© 2020, Karen Stéfany Martins.
. Todos os direitos reservados

Martins, Karen Stéfany

M386i O impacto do ruído de atributo na classificação da polaridade de críticas de filmes [manuscrito] / Karen Stéfany Martins. — 2020.
xx, 67 f. il.; 29 cm.

Orientador: Pedro Olmo Stancioli Vaz de Melo.
Corrientador: Rodrygo Luis Teodoro Santos.
Dissertação (mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.
Referências: f. 55-67

1. Computação – Teses. 2. Aprendizado profundo – Teses . 3. Mineração de opinião – Teses. 4. Critica cinematografica – Teses. I. Melo, Pedro Olmo Stancioli Vaz de II. Santos,Rodrygo Luis Teodoro. III.Título.

CDU 519.6*82(043)

Ficha catalográfica elaborada pela bibliotecária Belkiz Inez Rezende Costa
CRB 6ª Região nº 1510

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

ON THE IMPACT OF ATTRIBUTE NOISE ON MOVIE REVIEW POLARITY CLASSIFICATION

KAREN STEFANY MARTINS

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

Prof. Pedro Olmo Stancioli Vaz de Melo - Orientador
Departamento de Ciência da Computação - UFMG

RLSantos
Prof. Rodrygo Luis Teodoro Santos - Coorientador
Departamento de Ciência da Computação - UFMG

Helena de M. Caseli
Profa. Helena de Medeiros Caseli
Departamento de Computação - UFSCAR

Prof. Adriano Alonso Veloso
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 16 de Outubro de 2020.

Acknowledgments

First and foremost, I would like to thank God. Without his blessings, this achievement would not have been possible. My parents, Dorinha and José, for never going to any lengths to allow me to have a quality education and encourage me to follow my dreams. In particular, my mother for the unconditional love and support. My brother, Alex, for being a great inspiration in my life and supporting me in all my choices. My niece, Laura, for bringing joy to our lives. My boyfriend, Gianlucca, for encouraging me not to give up in the most difficult moments.

I would also like to express my gratitude to my advisor, Pedro Olmo, for guiding me throughout this master's journey and always being available. My co-advisor, Rodrygo Santos, for accepting to be part of this work and for always presenting important ideas during all discussions. My colleagues from Wisemap/WiseDados Lab for the friendship and support. Finally, I also thank the entire Computer Science Department (DCC) at UFMG.

Abstract

With the growth of the internet, movie review websites have changed the cinematography industry. It has been affecting the movie's box office, for example. The review polarity is very important in several applications. Some of them use machine learning classifiers to define the review polarity. However, these classifiers are not perfect. They are often criticized for the lack of explanation of their successes and failures. This work helps to fill this gap by proposing a methodology to characterize, identify, and measure the impact of problematic instances in the task of polarity classification of movie reviews. We characterize such instances by two types of attribute noise: *neutrality*, where the review text does not convey a clear polarity, and *discrepancy*, where the polarity of the text does not match the polarity of its rating. To do that, we propose a human classifier which is composed of three independent human annotators. Each annotator classifies the reviews on two levels. On the first level, they classify the review in relation to its polarity, that is, positive or negative. Next, on the second level, they answer whether they are confident or not about their classification and why. Then, we aggregate their answers using the majority vote. Finally, we test state-of-the-art machine learning classifiers on these reviews. From these steps, we quantify the amount of attribute noise in polarity classification of movie reviews and provide empirical evidence about the need to pay attention to such problematic instances, as they are much harder to classify, for both machine and human classifiers. Our proposed methodology is simple and can be easily applied to other classification tasks. To the best of our knowledge, this is the first systematic analysis of the impact of attribute noise in polarity detection from well-formed textual reviews.

Keywords: Attribute Noise; Deep Learning; Explainability; Opinion Reviews; Opinion Mining; Movie Reviews.

Resumo

A partir do crescimento da Internet, sites de críticas de filmes mudaram o setor cinematográfico. Eles podem afetar as bilheterias dos filmes, por exemplo. A polaridade dessas críticas é muito importante em várias aplicações. Algumas delas usam classificadores baseados em aprendizado de máquina para definir a polaridade. No entanto, esses classificadores não são perfeitos. Eles são frequentemente criticados pela falta de explicação dos seus sucessos e fracassos. Este trabalho ajuda a preencher essa lacuna, propondo uma metodologia para caracterizar, identificar e medir o impacto de instâncias problemáticas na tarefa de classificação da polaridade de críticas de filmes. Caracterizamos essas instâncias por dois tipos de ruído de atributo: neutralidade, quando o texto da crítica não transmite uma polaridade clara e discrepância, quando a polaridade do texto não corresponde à polaridade definida pelo autor. Para fazer isso, propomos um classificador humano composto por três juízes humanos independentes. Cada juíz classifica as críticas em dois níveis. No primeiro nível, eles classificam em relação à sua polaridade, isto é, positiva ou negativa. Em seguida, no segundo nível, eles respondem se estão confiantes ou não sobre a sua classificação e o por quê. Em seguida, agregamos suas respostas usando o voto da maioria. Por fim, testamos os classificadores baseados em aprendizado de máquina nessas críticas. A partir dessas etapas, quantificamos a quantidade de ruído em atributo na classificação de polaridade de críticas de filmes e fornecemos evidências empíricas sobre a necessidade de prestar atenção a essas instâncias problemáticas, pois são muito mais difíceis de classificar, tanto para os classificadores máquinas quanto para os humanos. Nossa metodologia proposta é simples e pode ser facilmente aplicada a outras tarefas de classificação. Até onde sabemos, esta é a primeira análise sistemática do impacto do ruído de atributo na detecção de polaridade a partir de críticas textuais bem formadas.

Palavras-chave: Ruído em Atributo; Aprendizado Profundo; Explicabilidade; Críticas de opinião; Mineração de Opinião; Críticas de Filmes.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Movie review websites are popular online services that allow users to evaluate movies using a score and a text explaining the score. In addition, users can also view other users' reviews and start discussions. Several studies revealed that these reviews can influence other people's behavior, affecting the audience and consequently the cinematography industry [Nagamma et al., 2015; Wankhede and Thakare, 2017]. Many people prefer to read the reviews about a movie before deciding whether to watch it or not [Topal and Ozsoyoglu, 2016]. In the United States, for example, about 36% of moviegoers check the site's reviews often before seeing a movie.[1] Furthermore, research has shown that 7 out of 10 people are less interested in watching a movie if its Rotten Tomatoes score is between 0 and 25 points.[2] Therefore, reviewers have the power to increase or decrease the audience of a movie [Boatwright et al., 2007].

The problem is that reviewers have their personal biases and own quality standards [Xiaojing Shi and Xun Liang, 2015]. In a scale from 0 to 100, a score of 70 for a highly demanding reviewer may have the same meaning as a 90 for a more tolerant one. This can also happen when reviewers value movie aspects, such as acting, script, lighting, costumes, soundtrack, photography, special effects, among others, differently. For instance, one is very critical towards the movie script, while the other disregards the quality of the script at the expense of good action scenes. Thus, the textual content of reviews is generally much more meaningful and rich to assess the quality and to serve as basis for recommendations than their numerical scores [Xiaojing Shi and Xun Liang, 2015]. Because of that, and allied to the fact that many reviews do not have a numerical score associated with them, several solutions exist in the literature

[1] Available in: https://www.latimes.com/business/hollywood/la-fi-ct-rotten-tomatoes-20170721-htmlstory.html. Accessed: 06-13-2020.

[2] Available in: https://www.hollywoodreporter.com/news/studios-fight-back-withering-rotten-tomatoes-scores-1025575. Accessed: 06-13-2020.

to automatically classify the polarity of movie reviews [Ouyang et al., 2015; Lai et al., 2015; Hassan and Mahmood, 2018; Zhou et al., 2015; Wang et al., 2016].

A common approach to solve this problem is through supervised machine learning, i.e., the target function to output the polarity of a review is estimated using labeled data [Zhu et al., 2003]. The problem is that if the training data is not representative nor reliable, most likely the trained model will not perform well in production. Imagine, for instance, a training set containing a significant number of examples in which human specialists, when presented to them, do not fully (or need more information to) agree with their labels. Again, models trained with this data will most likely present high error rates. Under these circumstances, a challenging problem is to identify *whether* and *which* particular sets of training data led the model to perform poorly. Identifying such sets of data is of fundamental importance to build better and more general models that can cope with large portions of noisy data in the problem of review classification.

In fact, a huge challenge for all the existing supervised machine learning solutions is data noise, that is, anything that obscures the relationship between the features and the class of a given instance [Frenay and Verleysen, 2014; Beigman and Klebanov, 2009; Beigman Klebanov and Beigman, 2014]. There are two classes of noise in the training data: class noise and attribute noise [Gupta and Gupta, 2019; Frenay and Verleysen, 2014; Nettleton et al., 2010; Van Hulse et al., 2007; Zhu and Wu, 2004; Teng, 1999]. Class noise occurs when the training data contains instances that are wrongly labeled [Gupta and Gupta, 2019; Frenay and Verleysen, 2014; Beigman and Klebanov, 2009; Beigman Klebanov and Beigman, 2014]. For instance, a negative review is labeled as positive by mistake by the reviewer. On the other hand, attribute noise occurs when the training data contains one or more attributes with wrong, incomplete or missing values [Gupta and Gupta, 2019; Van Hulse et al., 2007]. A simple example addressed in this work is a review that does not have an explicit opinion. A more complex one is a review of a highly acclaimed movie where only negative points are highlighted to justify a positive but different from perfect score. These attribute noises may dramatically affect the effectiveness of the supervised learning solutions [Gupta and Gupta, 2019].

1.1 Thesis Statement

Besides degrading the performance of supervised learning solutions [Gupta and Gupta, 2019], attribute noise is hard to characterize and identify, especially in textual data [Van Hulse et al., 2007]. Arguably, human-generated unstructured textual data is inherently prone to attribute noise, which can take the form of grammar errors, dialects, slangs, profanities/slurs, humour, off-topic content, irony and sarcasm, among others [Michel and Neubig, 2018; Eisenstein, 2013]. Several studies have characterized and measured the impact of naturally occurring noisy text inputs. However, such studies either inject synthetic noise to the attributes [Agarwal et al., 2007] or focus on simpler forms of noise like ungrammatical constructs [Baldwin et al., 2013; Michel and Neubig, 2018; Dey and Haque, 2009], which can be easily identified and automatically corrected. To the best of our knowledge, no previous study has systematically characterized attribute noise in real samples of well-written text reviews or its impact on polarity classification.

Other important problem that affects machine learning classifiers is explainability. Despite their potential in many tasks, these solutions have a lack of explanation of their outputs [Lipton and Steinhardt, 2019]. Generally, how the model makes a decision is not clear or simple to understand. For this reason, they are known as black boxes models. This lack of explanation makes humans insecure about their output and utilization. Many works were proposed to open these black boxes and understand their behaviour [Raaijmakers et al., 2017; Park et al., 2017; Lundberg and Lee, 2017]. A common way to do it in the literature is through the design of controlled test datasets. In other words, some authors usually create these datasets to evaluate how the model behaviour in a specific problem. Then, they compare the results with original ones, generally created by human annotators [Poliak et al., 2018; Marvin and Linzen, 2018; Conneau et al., 2018], to understand and analyze their performance.

In this master thesis, we join these two problems: attribute noise and explainability. We propose a methodology to characterize, quantify and measure the impact of attribute noise in polarity classification tasks. We demonstrate its usefulness in the task of movie review polarity classification. Our goal is to investigate the impact of attribute noise of real samples of well-written text reviews on the performance of human and machine classifiers. Also, in terms of explainability, we propose a human classifier able to identify noisy instances and generate a controlled test set to evaluate the behaviour of state of the art classifiers on noisy textual data.

1.2 Our Solution

In order to quantify the amount and impact of attribute noise, we collected 415.867 movie reviews from Metacritic[3], a website that contains user-generated reviews on many domains, including movies. One advantage of using Metacritic is that the meaning of the scores is clearly stated to the users when a review is being submitted: positive

[3] Available in: https://www.metacritic.com/movie. Accessed: 06-13-2020.

reviews are between 61 and 100, mixed are between 40 and 60, and negative from 0 to 39. Because of that, class noise and biases should be rare, i.e., a user who liked (disliked) a movie will very unlikely give a negative (positive) score to it. To make this error even less prone to occur, we collected only positive and negative reviews. Thus, we are basically left with only attribute noise, which we assign into two disjoint categories: *neutrality* and *discrepancy*. A *neutral* review does not have a clear polarity and a *discrepant* review has a human-perceived polarity that is different from its associated score. Note that this categorization is complete, i.e., every instance that, for a human, does not reveal its class clearly has one of these two types of attribute noise.

To formally define *neutral* and *discrepant* reviews, we propose a methodology based on a well-defined human classifier, which differently from machine classifiers, uses human reasoning to infer the class of the example. Our proposed human classifier is composed by three independent human annotators and predicts the polarity based on the majority vote among these annotators. When the class assigned by the human classifier is incorrect, we label the review as *discrepant*, i.e., the human-perceived polarity of the text is different from its associated score. When the human classifier is not confident about its prediction, we label the review as *neutral*. In total, the human classifier labeled $1,200$ reviews and found 198 *neutral* and 64 *discrepant* reviews. We trained state of the art methods and the best classifiers achieved an accuracy of approximately 90% [Devlin et al., 2019]. Then, we tested the machine classifiers on these reviews and results revealed that attribute noise can significantly decrease their performances.

1.3 Contributions

In summary, the main contributions of this thesis are:

- A simple and reproducible methodology based on a well-defined human classifier to characterize and identify attribute noise for polarity classification tasks;
- A thorough analysis of the impact of attribute noise in the task of movie review polarity classification;
- Publicly available datasets of movie reviews describing the expected amounts of five classes (*discrepant* and *neutral*, which can be of four types: *factual*, *mixed opinions*, *contextual* and *undefined*) of attribute noise.

1.4 Chapter organization

The remainder of this thesis is organized as follow: Chapter 2 is dedicated to fundamentals and related work. We review works on polarity classification of documents, attribute noise, movie reviews polarity classification and machine learning explainability. In Chapter 3, we describe our proposed methodology for identifying and measuring the impact of attribute noise in the task of movie review polarity classification. Next, in Chapter 4, we describe the experimental setup used to apply our proposed methodology, including dataset, machine classifiers and model training. In Chapter 5, we present the results. Finally, in Chapter 6, we present concluding remarks and future work.

Chapter 2

Fundamentals and Related Work

This Chapter presents the related works to this thesis. Section 2.1 defines the polarity classification task of documents. Next, Section 2.2, we explore polarity classification of movie reviews using supervised machine learning. Section 2.3 describes a common problem, known as attribute noise, that affects classifiers. Finally, we discuss the lack of explanation of machine learning classifiers in Section 2.4.

2.1 Polarity Classification

With the rapid growth of the internet, the volume of online reviews available increased considerably. Thereby, sentiment analysis of reviews has gained focus [Vinodhini and Chandrasekaran, 2012; Zhang et al., 2018; Tang et al., 2009]. Sentiment analysis (SA), also known as opinion mining, is one of the most active research areas in natural language processing (NLP), data mining and information retrieval [Zhang et al., 2018; Tang et al., 2009]. Its main goal is to identify the mood about a particular product or a topic [Vinodhini and Chandrasekaran, 2012]. Due to the fact that people do not express opinions in a same way and an opinion word can be considered positive or negative depending on the situation, there are many challenges in this field [Vinodhini and Chandrasekaran, 2012; M. et al., 2016]. In addition, SA is studied in three levels of granularity: document level, sentence level, and aspect level [Medhat et al., 2014; Zhang et al., 2018; Baldania, 2017; Vinodhini and Chandrasekaran, 2012; M. et al., 2016]. This work focuses on document level analysis.

Document level sentiment classification consists of classifying the polarity of a document, which must (or should) contain opinions about a single entity (e.g. a book) [Zhang et al., 2018; Medhat et al., 2014]. For instance, given a book review, the system determines whether the review text expresses an overall positive, negative, or neutral opinion about the book. Although polarity classification naturally suits to analyze consumer opinions about products and services [Zhou et al., 2015; Gui et al., 2017; Fang and Zhan, 2015], it is also well suited to various types of applications, such as to infer votes in elections [Goldberg et al., 2007], civilian sentiment during terrorism scenarios [Cheong and Lee, 2011], citizens' perception of government agencies [Arunachalam and Sarkar, 2013] and recommendation systems [Zhang, 2015]. In this work, the entities are movies and the documents, reviews. Movie reviews can be associated with scores from various scales (e.g. 0 to 100, for Metacritic, and 0 to 10, for IMDB) as a regression task, but also with a single binary opinion (e.g. positive or negative, for Rotten tomatoes) that ignores neutral sentiment as a binary classification task [Zhang et al., 2018].

The study of SA dates back to the late 1990s. In spite of that, it only became a major sub field of the information management discipline in the early 2000s [Tang et al., 2009; Vinodhini and Chandrasekaran, 2012]. Since then, several works have been published defining different techniques. Nowadays, the literature indicates that SA techniques can be divided in two categories. The first category considers approaches based on machine learning. On the other hand, the second category considers approaches based on the lexicon [Vinodhini and Chandrasekaran, 2012; Lu et al., 2018; M. et al., 2016; Shelke et al., 2012; Medhat et al., 2014]. Some authors also consider a third one, called hybrid approach, which is a combination of the first two [Lu et al., 2018; Medhat et al., 2014; Rezaeinia et al., 2019].

Machine learning (ML) is a subset of Artificial Intelligence (AI) that studies computer algorithms that improve automatically through experience [Mitchell et al., 1997]. These algorithms are widely used in SA to classify the polarity of a document. In this context, they can be subdivided into two groups: supervised and unsupervised learning methods. In supervised learning, it is necessary that the training documents contain labels. Considering movie reviews, for example, the label can has two levels (positive or negative) or more (for instance, 1 to 5). There are different kinds of supervised methods in the literature such as probabilistic, linear and decision tree classifiers. While supervised methods depend on labeled training documents, unsupervised methods do not have this need [Medhat et al., 2014]. These methods try to infer the label from the data. It is important when we need to classify documents but the training set is unlabeled. One common approach is to cluster the data into categories based on their statistical properties [Baldania, 2017].

Lexicon-based approach mainly consists of creating a sentiment lexicon to identify the sentiment polarity and strength of words and phrases. It is a list of words where each one is assigned with their respective positive or negative score [Tang et al., 2015].

There are two approaches to build it: Dictionary-based and Corpus-based. In the first one, a set of opinion words is selected manually with known polarity. Next, this set is expanded with their synonyms and antonyms using well known corpora like WordNet [Miller et al., 1990]. After added these new words to the seed list, the process is repeated until no new words are found [Medhat et al., 2014]. A major drawback of this approach is the inability of finding the domain and context specific orientations of opinion words. This problem is solved by the Corpus-based approach which is based on syntactic patterns or patterns that occur together [Medhat et al., 2014].

2.2 Movie Reviews Polarity Classification

In this section, we focus on polarity classification in the context of movie reviews. We will briefly describe the most common supervised machine learning approaches giving examples of works that used movie reviews as dataset.

2.2.1 Probabilistic Classifiers

Probabilistic classifiers, also known as generative classifiers, are constructed from generative models which enables to analyze complex domains [Garg and Roth, 2001]. Naive Bayes (NV), Bayesian Network (BN) and Maximum Entropy Classifier (ME) are among the most popular probabilistic classifiers [Bhavitha et al., 2017; Medhat et al., 2014]. The first classifier estimates the posterior probability of a class, based on the distribution of the words in the document, assuming word independence [Medhat et al., 2014; Vinodhini and Chandrasekaran, 2012]. The second classifier is a directed acyclic graph where nodes are variables and edges correspond to conditional dependency [Bhavitha et al., 2017; Medhat et al., 2014]. However, due to its expensive computation, it is not usually used in text classification [Bhavitha et al., 2017]. Lastly, ME converts labeled feature sets to vectors through encoding which is used to calculate combined weights for each feature to determine the most likely label for a feature set [Medhat et al., 2014]. Pang et al. [2002] compared NV and ME in movie reviews domain. According to them, NB tends to have the worst performance.

2.2.2 Linear Classifiers

Linear classifiers use linear functions in the classification as a separating hyperplane between different classes [Medhat et al., 2014]. Support Vector Machines (SVM) is an example of linear classifier that was originally proposed by Boser et al. [1992]. The idea behind it is to find hyper-planes with maximal distance which can separate different classes [Bhavitha et al., 2017]. Pang et al. [2002] also compared SVM in their work. The results showed that SVM tend to do best than NV and ME, regardless of the small difference. Neural Network (NN) is another example of linear classifier which is composed of multiple neurons as an elemental unit. Anyhow, Multilayer neural network are used for non-linear margins, in this case, the output of the previous layer is used as input of the next one and the training is more complicated due to the fact that the errors have to be back-propagated over the layers [Medhat et al., 2014; Bhavitha et al., 2017]. Moraes et al. [2013] compared SVM with Artificial Neural Networks (ANN) in different domains for balanced and unbalanced datasets. For movie reviews, ANN outperformed SVM significantly in both cases.

2.2.3 Decision Tree Classifiers

Many variations of Decision trees (DT) have been developed over the years. The core idea behind this concept is to combine a sequence of simple tests where each test is a comparison of an attribute against a value. The value can be a threshold, if it is a numerical attribute, or a set of possible values, if it is a nominal attribute. Unlike neural networks, this method is very easy to interpret [Kotsiantis, 2013]. However, it has difficulty handling noisy data and overfitting problems [Bhavitha et al., 2017]. Palkar et al. [2016] compared NB, ME, SVM, DT and Random Forest (RF), which is constructed from several DT. They evaluated the algorithms in binary classification using three movie reviews datasets. For each algorithm, they also evaluated pre-processing based on whether it was carried out or not. Although they used three datasets, two of them were too large to NB, DT and DF operate without pre-processing the input. Considering only the other dataset, NB performed quite poorly, mainly without pre-processing. SVM and ME outperformed the other algorithms. Last, DT and RF also produced average results with and without pre-processing. These classifiers were also compared by Yasen and Tedmori [2019]. In their work, RF got the best accuracy.

2.2.4 Deep Learning

Deep learning has become widely used in recent years. Its architectures have improved the state-of-the-art of sentiment classification of documents [Tang et al., 2015]. In the past, researchers began to abandon the study of neural networks due to the high computational cost of training networks with more than one or two layers, that is, deep neural networks. Howbeit, the hardware advancement, the huge amount of training data available and the powerful learning representations enabled the growth of these networks [Zhang et al., 2018]. Tang et al. [2015] summarized deep learning as follows:

> Deep learning is a kind of representation learning approach. It learns multiple levels of representation with nonlinear neural networks, each of which transforms the representation at one level into a representation at a higher and more abstract level. The learned representations can be naturally used as features and applied for detection or classification tasks.

Traditionally, documents can be represented using bag of words (BOW) or Term frequency-inverse document frequency (TF-IDF). In BOW, the document is represented as a vector of words where each position contains the word occurrence (whether the word occurs or not in the document) or the word frequency [Kusner et al., 2015; Zhang et al., 2018]. In TF-IDF, each position contains the TF-IDF score which evaluates how relevant a word is in a document. In other words, a term that appears in many documents is less important than one that appears just a few times. Nevertheless, the vector of words in these representations is the size of the vocabulary in the dataset, which makes the vector sparse and with high computational cost [Zhang et al., 2011]. Furthermore, the order and semantics of words are ignored.

To alleviate these problems, a new technique called word embedding was created. Word embedding is used as input for deep learning methods. This technique converts vocabulary words to vectors of continuous real numbers. The vectors, known as dense vectors, are low dimensional. Unlike BOW and TF-IDF, this technique is not only concerned with whether the words occur or not in the document. It is also capable of learning the semantic and syntactic of words. The document representation is generated from word embedding using neural networks [Zhang et al., 2018]. Mikolov et al. [2013a,b] proposed Word2Vec, a model that learns high-quality word vector representations from huge datasets. Recently, Devlin et al. [2019] proposed a new method of pre-training language representations called BERT (Bidirectional Encoder Representations from Transformers) that considers left and right context of the word. This method obtained new state-of-the-art results on eleven natural language processing tasks.

Two deep learning models have achieved great successes recently in text classification: convolutional neural network (CNN) and Long short-term memory (LSTM) [Zhou et al., 2015]. CNN was first invented for computer vision, then it also proved to be effective for NLP tasks. It uses word sequences as input and it is capable of capturing local correlations of spatial or temporal structures. Moreover, it applies two operations:

convolution filters and pooling. With convolution filters, it obtains multiple features. Afterwards, it applies pooling operations over the features to select the most important ones, learning short and long-range relations [Kim, 2014; Zhou et al., 2015]. Several architectures were proposed using variations of CNN [Ouyang et al., 2015; Lai et al., 2015; Kim, 2014]. Although CNN is capable of learning local response from temporal and spatial data, it lacks the ability to capture learning sequential correlations [Liu and Guo, 2019; Zhou et al., 2015]. For this purpose, Recurrent Neural Network (RNN) is a neural network class that has a kind of memory formed through the directed cycle between neuron connections. This memory is able to process a sequence of inputs where the outputs are dependent on all previous computations, forming a remembering mechanism.

Despite the fact that RNN can learn contextual information, for long data sequences, some traditional implementations cause shortcomings like exploding. LSTM is a kind of RNN architecture created for this purpose. Thus, it is capable of learning long-term dependencies, becoming a powerful approach to extract the high-level text informations [Zhang et al., 2018; Liu and Guo, 2019; Zhou et al., 2015]. Many works proposed architectures combining these models with pre-trained word embeddings, achieving great results [Hassan and Mahmood, 2018; Zhou et al., 2015; Wang et al., 2016; Liu and Guo, 2019; Zhou et al., 2016]. Wang et al. [2016], for instance, combined CNN with Gated Recurrent Units (GRU), which is a slight variation of LSTM with pre-trained word embeddings. Different from their work, Zhou et al. [2015] used LSTM instead of GRU.

2.3 Attribute Noise

Supervised machine learning is one of the most common and successful approaches for polarity classification [Jochim and Schütze, 2014; Pozzi et al., 2016; Lee et al., 2018; Deriu et al., 2017]. The problem with this approach is that the quality of training and test data can significantly influence the results. These data may contain noise generated during the data collection and preprocessed phase from human error while translating information or limitations of the measurement equipment [Nettleton et al., 2010]. The noise can occur in the attribute values (attribute noise) or in the class values (class noise). Class and attribute noise can increase the learning complexity and, consequently, reduce classification accuracy [Zhu and Wu, 2004; Gupta and Gupta, 2019; Beigman Klebanov and Beigman, 2014].

Class noise is considered to be more harmful than attribute noise [Frenay and

Verleysen, 2014], but it is easier to detect [Van Hulse et al., 2007]. Therefore, class noise is more addressed in the literature [Zhu and Wu, 2004; Gupta and Gupta, 2019], and several studies analyzed the impact of class noise in classification tasks [Jamison and Gurevych, 2015; Beigman and Klebanov, 2009; Beigman Klebanov and Beigman, 2014] and proposed approaches to deal with this problem [Fefilatyev et al., 2012; Hendrycks et al., 2018; Toledo et al., 2015; Sukhbaatar et al., 2015; Frenay and Verleysen, 2014; Natarajan et al., 2013; Prati et al., 2019; Barbosa and Feng, 2010; Younes et al., 2010; Liu et al., 2017; Rehbein and Ruppenhofer, 2017; Jindal et al., 2019]. Beigman Klebanov and Beigman [2014], for example, showed that is important to pay attention in instances from annotated data that are problematic and disagreeable because they can be disruptive to the classifier. On the other hand, some works proposed a classifier that is robust to label noise [Jindal et al., 2019; Younes et al., 2010; Barbosa and Feng, 2010; Sukhbaatar et al., 2015; Hendrycks et al., 2018]. There are also approaches to detect it automatically [Rehbein and Ruppenhofer, 2017; Toledo et al., 2015; Fefilatyev et al., 2012] and to correct it dynamically [Liu et al., 2017; Toledo et al., 2015]. Different from those, Sáez et al. [2016] considered that the performance and noise robustness of the classifiers are two different concepts. To minimize the impact of considering these two concepts separately, they proposed a new measure to determine the expected behavior of a classifier against class noise.

Despite of being less harmful, attribute noise can also bring severe problems to data analysis [Gupta and Gupta, 2019] and it should not be ignored [Zhu and Wu, 2004]. In addition, many studies suggests that a good idea to deal with class noise is eliminating the instances that contain it, which may not be a good strategy for attribute noise, given that other attributes of the instance can contain valuable information. Besides, in real-world data, the class information is usually cleaner than attributes. In other words, during the pre-processing phase of the data the attributes need more attention [Zhu and Wu, 2004].

Most of the studies that investigated the impact of attribute noise and proposed solutions to deal with it did so by inserting synthetic noise in the attributes [Nettleton et al., 2010; Teng, 1999; Pujara et al., 2017; Vaibhav et al., 2019; Zhu and Wu, 2004; Mannino et al., 2009; Agarwal et al., 2007]. Nettleton et al. [2010], for example, tested classifiers on a synthetic dataset perturbed with different proportions of attribute noise and class noise to compare the effect of noise in both training and test datasets. The datasets contain only numerical attributes. To generate attribute noise, they altered some values according to Gaussian and uniform distributions. Results showed that the type and percentage of noise in the dataset influences the behavior of classifiers. In addition, noise in the training dataset is more harmful. Following the same idea, Zhu and Wu [2004] concluded that correcting test set attribute noises can improve the classification accuracy even if the training set has noise, and the lowest classification accuracy occurs when both training and test set are corrupted.

Instead of comparing the classifiers behaviour on different amounts of noise, Teng [1999] focused on correcting the noisy instances. They proposed an approach to handle this problem in the training data by identifying possible noisy attributes and classes. Their approach consists of two phases. First, it identifies possible noisy attributes, trying, for each attribute, to predict it using the class and the other attributes as features with the goal to exploit the interdependence between attributes. Then, using the grouped results for each attribute, it selectively replaces some of the attribute and classes values to obtain a better fit of the instance, preserving much of the original information. To test it, they artificially corrupted the training set with random noise and tested it on a classifier. As a result, the accuracy of the classifier has increased using the polished data. Unlike the last two works, Pujara et al. [2017] introduced noise in the dataset to determine how performance degrades in the context of graphs.

According to Van Hulse et al. [2007], there are two main problems of inserting synthetic noise in the attributes into datasets. First, it is not known whether the original dataset is noise-free or clean. Adding noise to a dataset that is already noisy makes it difficult to analyze the performance of the algorithms. Second, the synthetic noise may not be a good representation of the real-world dataset for a given domain.

Different from the last cited works about synthetic noise, Valdivia et al. [2019] showed that ratings in TripAdvisor reviews are not strongly correlated with sentiment scores given by unsupervised sentiment analysis methods. They observed that users tend to choose a high score even when they write the review using negative words in some sentences. These negative words suggest a not-so-high score, potentially making the score and review inconsistent. As a consequence, sentiment analysis methods predict the review as neutral or negative. To solve this problem, they proposed a unified index that aggregates the review and score polarities. Basically, the model consists of a geometric mean between the polarities that is capable of fixing the mismatch between humans and unsupervised sentiment analysis methods.

Furthermore, Li et al. [2020] investigated the impact of the reviews' textual quality on sentiment analysis task of movie reviews using deep learning approaches. They used two measures: word count and readability. Results showed that short length and high readability achieved the best performance.

Regarding attribute noise in textual data, many studies have analyzed how they affect computational tasks [Baldwin et al., 2013; Michel and Neubig, 2018; Agarwal et al., 2007; Arora and Kansal, 2019; Contractor et al., 2010; Dey and Haque, 2008, 2009; Lopresti, 2005; Bermingham and Smeaton, 2010; Subramaniam et al., 2009; Esuli and Sebastiani, 2013; Vinciarelli, 2005; Florian et al., 2010]. For instance, Bermingham and Smeaton [2010] showed that it is easier to classify sentiment in short documents (e.g. tweets) than in longer ones, as short documents have less non-relevant information. Other works [Vinciarelli, 2005; Subramaniam et al., 2009; Lopresti, 2005] analyzed the impact text generated through automatic recognition processes (e.g optical character recognition (OCR), automatic speech recognition (ASR) systems) have on text processing algorithms. Although these texts contain several forms of noise (e.g. deletions and insertions of character or words), many techniques exist to circumvent their negative effects [Subramaniam et al., 2009].

The same is true for attribute noise in the form of errors in language rules, such as typos, grammatical errors, improper punctuation, irrational capitalization and abbreviations, which are very common but easy to deal with [Lourentzou et al., 2019; Agarwal et al., 2007; Arora and Kansal, 2019; Contractor et al., 2010; Dey and Haque, 2008, 2009; Subramaniam et al., 2009; Florian et al., 2010; Michel and Neubig, 2018]. Contractor et al. [2010], for instance, presented an unsupervised method to translate noisy text in clean text. Moreover, there are also works that proposed a sentiment analysis framework with a pre-processing phase to reduce these linguistic noises [Dey and Haque, 2008; Arora and Kansal, 2019]. Different from those, Esuli and Sebastiani [2013] assumed that the noise is in the class instead of the text which became a class noise problem.

In addition to us, other studies have proposed systematic processes to identify attribute noise and quantify its impact on classifiers [Baldwin et al., 2013; Van Hulse et al., 2007; Khoshgoftaar and Van Hulse, 2009; Michel and Neubig, 2018; Dey and Haque, 2009; Agarwal et al., 2007]. Agarwal et al. [2007] measured the impact noisy documents have on automatic text classifiers by inputting synthetic noise on their features. To inject synthetic noise and estimate its effect, they artificially introduced spelling errors and noisy from automatic speech recognition transcription in different levels. They found that the performance was not very affected even with high noise levels. As we previous mentioned, using synthetic data is not reliable [Van Hulse et al., 2007]. Dey and Haque [2009] investigated how noise introduced due to incorrect English affects the performance of opinion mining techniques and proposed a framework that is able to effectively handle such noisy inputs. Baldwin et al. [2013] analyzed YouTube, Twitter, web user forum, blogs and Wikipedia to investigate how linguistically noisy social media are. They compared these sources with more conventional texts using statistical and linguistic analyses, like language distribution, lexical analysis, grammaticality and similarity.

Similarly, Van Hulse et al. [2007] used a software engineering expert to manually identify instances with noise in one or more attributes in a real-world software measurement dataset. They used the software engineering expert in two different phases. In the first phase, they applied an unsupervised clustering technique to partitioned the dataset into clusters. Then, the expert labeled instances that he was completely confident. Finally, they compared the labels assigned by the expert with the actual labels and verified the instances that did not have matching labels. These instances were considered with class noise and removed from the dataset, remaining only naturally occurring attribute noise. In the second phase, they proposed a approach for detecting instances with attribute noise and used the expert again to evaluate its effectiveness. Their approach is interesting because it can be used with or without knowledge of class labels, unlike other noise approaches. In their other work, instead of identifying noisy instances, they proposed a technique to rank attributes from most to least noisy also using software measurement dataset and software engineering expert for inspection [Khoshgoftaar and Van Hulse, 2009].

More related to our work, Michel and Neubig [2018] proposed a benchmark dataset for Machine Translation of Noisy Text (MTNT), composed by noisy comments posted on Reddit with their respective professionally sourced translations. They considered common social media types of noise such as abbreviations, typographical errors, obfuscated profanities, inconsistent capitalization, internet slang and emojis. To Identify noisy English texts, they pre-filtering the texts, eliminating comments that contain URL, automated comments from bots and comments in another language. Next, they selected comments that contain at least one out-of-vocabulary word which is a indication of noise. Their last step is to compute, for each comment, the normalized log probability of each of its lines in a language model, selecting those with low probability and labeling them as noisy. The authors also showed that existing machine translation models are heavily impacted by attribute noise. This dataset was used later to design an MT system resilient to such type of noise [Vaibhav et al., 2019].

Unlike this work, these approaches focused on noise that proved to be easy to detect, easy to deal with, or both, such as noise in the form of errors in language rules and in communication. Here we are interested in the attribute noise characterized by well-written texts that are not conveying their true classes clearly, e.g., attribute noise that hides the true polarity of textual reviews. To the best of our knowledge, we are the first to characterize, identify and measure the impact of such type of noise on classifiers.

Finally, it is important to point out that problems in the annotation of instances are related but very different from the problem tackled in this work. Several studies have shown that noisy (or hard) instances can significantly affect the annotation process [Beigman and Klebanov, 2009; Sharoff et al., 2010], which can potentially degrade the measures of inter-annotator agreement [Artstein and Poesio, 2008]. In this work, our proposed labeling process is not affected by noisy instances, but instead serves to identify them. In other words, the labels defined in this work (*neutrality* and *discrepancy*) are not directly associated with the class labels, but come from the labeling process. If the annotator is uncertain, the instance is marked as *neutral*. If certain, but the assigned label (e.g. polarity) is incorrect, the instance is marked as *discrepant*. Note that this process can be applied to practically any labeling task to identify attribute noise in instances, no matter the labels available to the annotators, and even when only one annotator is available, i.e., no inter-annotator agreement can be computed.

In our previous work [Martins and Vaz de Melo, 2019], we started to analyze and quantify the discrepancies in movie reviews through a sentiment analysis task. We defined that a text is considered discrepant from the score when the classifier fails. To do that, we applied a state of the art deep learning architecture on a large collection of movie reviews posted on Metacritic. We also proposed a metric capable of differentiating discrepancies between scores and text of movie reviews. Our results revealed that the score and text are usually not compatible. Unlike this work, we did not take into account that actually the classifier can fail because of the learning problem and noise present in the data, that is, the classifier error is mixed with the discrepancies and neutrality. In this thesis, we changed the definitions and proposed a different methodology to improve our analysis, considering both errors. Our work offers an alternative approach to characterize and quantify the types of noise in textual data. More specifically, we propose a well defined human classifier capable of identifying (and labeling) two *mutually exclusive* types of attribute noise in movie reviews relevant to the task of polarity classification, namely *discrepancy* and *neutrality*. We show that the performance of machine classifiers on such data degrade substantially, what corroborates with the labels given by the human classifier.

2.4 Machine Learning Explainability

Machine learning solutions, especially deep learning models, are often criticized for the lack of explanation of their successes and failures [Lipton and Steinhardt, 2019; Pelevina et al., 2016]. These models are known as black boxes due to the fact that it is not clear how they arrived at a given output from an input [Samek et al., 2017].

The explanation is important to ensure confidence of the model. Gilpin et al. [2018] defined explainability as “models that are able to summarize the reasons for neural network behavior, gain the trust of users, or produce insights about the causes of their decisions.” An explainable model can be appraised according to its interpretability and completeness, which are hard to achieve simultaneously. Interpretability is defined by Gilpin et al. [2018] as “the science of comprehending what a model did (or might have done)”. It aims to “describe the internals of a system in a way that is understandable to humans.” On the other hand, the objective of completeness is to “describe the operation of a system in an accurate way.” Although explainability and interpretability are often used interchangeably, there is a distinction between their definitions. While explainable models must be interpretable, interpretable models are not always explainable [Gilpin et al., 2018; Došilović et al., 2018].

There are many works that address explainability. Tenney et al. [2019] investigated Bert’s layers to understand how syntactic and semantic information is maintained in the structure. Results showed that the initial layers keep basic syntactic information and higher layers keep the semantic information. Besides, the model uses layers as hierarchies to deal with complex interactions. Similarly, Raaijmakers et al. [2017] developed a mechanism to identify statistical patterns of neighbor similarity across hidden layers in deep text mining with the aim of understanding the internal semantic representations, interpreting the hidden layer. Park et al. [2017] applied rotation algorithm on high-dimensional word vectors to improve their interpretability. Unlike those, Lundberg and Lee [2017] presented SHAP (SHapley Additive exPlanations) which is a game theoretic approach to interpret the output of complex models. In the polarity classification task of reviews, for example, it is able to understand the most important words for the machine classifiers. In other words, for a given review, it can describe which words led the classifier to choose the output.

More associated to our work, some authors designed test datasets to evaluate models behavior and understand what they are capturing in different aspects such as their ability to represent types of reasoning [Poliak et al., 2018], grammar of the language [Marvin and Linzen, 2018] and linguistic features of sentences [Conneau et al., 2018]. These works also used human annotators to compare if their errors are similar to the models. Furthermore, two of them aggregated these annotations using majority voting [Poliak et al., 2018; Conneau et al., 2018].

In this work, we designed a controlled test set to evaluate the behaviour of state of the art classifiers on noisy textual data. To identify noisy instances, we proposed a human classifier which consists of the majority vote of three human annotators. Our test set contains three main labels to indicate whether the review is neutral (the annotators are uncertain), discrepant (the annotators are certain, but the assigned polarity is incorrect) or no noise (the annotators are certain and the assigned polarity is correct). Then, we analyzed its impact on machine classifiers.

2.5 Human vs. Machine Classifiers

According to researchers, AI will probably outperforming humans in all tasks in 45 years [Grace et al., 2017]. In this way, humans are considered to be upper bound in terms of performance in many tasks such as classification. Many works compared the performance of humans, also known as human classifiers, and machine classifiers. Most of them focused on image classification task [Stallkamp et al., 2012; Wichmann et al., 2004; Graf and Wichmann, 2003; Ciresan et al., 2012; Geirhos et al., 2018; Han et al., 2015; Dodge and Karam, 2017]. Ciresan et al. [2012] evaluated their architecture, which achieved near-human performance, on object recognition benchmarks such as digits, Latin and Chinese characters, and traffic signs. Graf and Wichmann [2003] and Wichmann et al. [2004] analyzed gender classification in frontal views of human faces. Geirhos et al. [2018] compared the robustness of humans and convolutional deep neural networks on object recognition under image degradations. Similarly, Dodge and Karam [2017] and Geirhos et al. [2017] compared deep neural networks under image quality distortions and degradations. Han et al. [2015] investigated automatic demographic estimation, that is, estimation of race, gender and age of a person from his face image. Finally, Taigman et al. [2014] compared human performance in face recognition and Goodfellow et al. [2013] in arbitrary multi-digit numbers recognition.

Different from those, Mesaros et al. [2017] compared human and machine performance in acoustic scene classification. Tsapatsoulis and Djouvas [2017] focused on feature extraction for sentiment classification of tweets. They proposed a human-created index of terms that were generated by humans during tweet annotation. Then, they compared these tokens with automatically extracted features, under a machine learning framework in different classifiers. Results showed that human indicated tokens have the best tweet classification performance. In their other work, Tsapatsoulis and Djouvas [2019], they compared features indicated by humans with features extracted through deep learning in sentiment classification of short texts.

In this work, we proposed a human classifier which is able to identify attribute noise. Then, we compare human and machine classifiers in polarity classification of movie reviews. Our goal is to identify how they deal with attribute noise of well-written text reviews.

Chapter 3

Methodology

This chapter introduces the methodology used to investigate the effects of attribute noise on movie reviews classification. The problem setting is defined in Section 3.1. In Section 3.2, we define two possible hypotheses for attribute noise in reviews. The methodology to identify attribute noise will be described in Section 3.3.

3.1 Problem Setting

In this work, we focus on the problem of document-level polarity detection. More formally, in a dataset $\mathcal{D} = (X, Y)$ composed by a set of textual movie reviews X and their corresponding binary scores Y, each review $x_i \in X$ is associated with a score $y_i \in Y$ that can be either 0 (*positive*) or 1 (*negative*). For the purposes of this work, it is important that $\mathcal{D}$ does not contain any movie reviews that have been explicitly associated with a neutral score by their author, e.g. a score between 40 and 60 on *Metacritic*, to isolate attribute noise from the unclear descriptors of neutral reviews, avoiding class noise and biases.

To assess the impact of attribute noise, we test two types of classifiers to infer the polarity of textual movie reviews, a machine classifier f_M and a human classifier f_H. A classifier is defined as a function $f(x_i)$ that receives a textual movie review x_i as input and returns its polarity $\hat{y}_i \in \{0, 1\}$. Both classifiers will be explained in the next sections.

We use the human classifier to assign a label l_i to a large set of movie reviews x_i to indicate whether x_i has attribute noise or not. This label can be one (and only one) of a set L of manually defined labels that indicate the absence of attribute noise (*"no noise"*) or a characteristic of x_i (i.e., a type of attribute noise) that can contribute to (or cause) prediction errors. With that, we will be able to quantify the impact of attribute noise on machine classifiers and provide explanations about why they occur and how to avoid them in order to improve machine classifiers' accuracy. More specifically, for a machine classifier f_M and for all labels $l \in L$, *"no noise"* included, we will calculate the probabilities $P(l_i = l | y_i \neq \hat{y}_i)$ and $P(y_i = \hat{y}_i | l_i = l)$.

3.2 Types of Attribute Noise

A strong premise of this work is that the dataset $\mathcal{D}$ has no (or negligible) class noise, i.e., all polarity scores $y_i \in Y$ reflect the real opinion of the reviewer. To guarantee that, one needs to construct $\mathcal{D}$ using movie reviews from systems like *Metacritic* or *Rotten Tomatoes*, which have well defined meanings for the scores that are always visible to the reviewers, as will be discussed in Chapter 4. Thus, every time the polarity of text x_i is inconsistent with its score y_i, we assume x_i contains attribute noise. More specifically, we define two possible hypotheses explaining inconsistencies in the text, i.e., two disjoint types of attribute noise: (1) the text does not have a clear polarity, namely *neutrality*, and (2) the text has a clear polarity, but its score is the opposite one, namely *discrepancy*. These two types of attribute noise can be easily and unequivocally identified by our proposed human classifier described in the following section. In Section 4.2, we also discuss possible directions on how this can be done using automated methods.

A movie review x_i has attribute noise of type *neutrality* when its polarity is not clear. In particular, we define four labels for this type of attribute noise: *neutral_mixed* (text has mixed opinions), *neutral_factual* (text is purely factual), *neutral_contextual* (polarity needs context) and *neutral_undefined* (reasons are unclear). The *neutral_mixed* label considers reviews that have both positive and negative points about the movie but the overall opinion is not clearly stated. For instance: *"As dumb as the film is, the actors escape relatively unscathed."* The *neutral_factual* label defines non-opinionated reviews, that is, the review only describes facts about the movie. For instance: *"It is a movie about the World War II and its consequences on the lives of those who survived."* The label *neutral_contextual* characterizes reviews where context is needed to understand its polarity, including those containing irony and sarcasm. For instance: *"Ultimately, Collin's film is one of forgiveness and that's not the usual way great tragedies end."* Finally, the label *neutral_undefined* is given to reviews where the reasons for the lack of polarity are not clear. For instance: *"Wow, can't believe the critics on this one."*

The second type of attribute noise, namely *discrepancy*, is given to reviews where the polarity of its text is the opposite of the polarity of its score. For this type, we define a single label: *discrepant* (polarity of text and score are discrepant). For instance, consider a highly acclaimed movie of a prestigious director, such as Martin Scorsese. Now, consider a reviewer who liked this movie, but unlike the vast majority of critics, found many points that prevent her from giving it a perfect score. Thus, the text will mostly be about its negative points to justify why she is not giving the expected perfect score. Consequently, the text review will appear negative although the score is positive. For instance, consider the following review, which has a negative polarity, but its score is positive: *"Thoroughly predictable from start to finish."*

3.3 Human Classifier

A fundamental building block of our methodology is the human classifier f_H. Human classifiers are often considered to be the upper bound in terms of performance of classification tasks [Stallkamp et al., 2012; Wichmann et al., 2004; Graf and Wichmann, 2003; Ciresan et al., 2012; Geirhos et al., 2018]. This means that when it makes a prediction error, machine classifiers will most likely also miss. Moreover, when a human classifier working on its full capacity makes a mistake, and the class label is correct (i.e. no class noise), then what caused the error is most likely attribute noise [Zhu and Wu, 2004]. We use this premise to define the two types of attribute noise discussed in the previous section.

In the task of polarity classification of movie reviews, a human classifier mistake on movie i can be due to two causes: **(C1)** the text of the review x_i is not clear about its polarity y_i, or **(C2)** the score y_i is different from the (clearly) perceived polarity of x_i. In other words, the human classifier f_H can be characterized by two binary features when executing this task: whether it is confident about its prediction **(F1)** and whether it correctly classified the polarity of the review x_i **(F2)**. Thus, when it makes a mistake, if it was not confident, an error of type **C1**, occurs, and when it was confident, an error of type **C2** occurs. The first one **(C1)** is associated with attribute noise of type *neutrality* and the second one **(C2)** is associated with attribute noise of type *discrepancy*. Also, while the second only occurs when the human classifier f_H makes a mistake, the first occurs every time f_H is not confident, i.e., it is independent of the prediction $\hat{y}_i$. Instances for which the human classifier is not confident can be seen as the *hard cases* defined by Beigman and Klebanov [2009], where labels are unreliable due to some difficulty faced by the annotator in giving such labels. This difficulty can make the labels reflects the human classifier preferences and biases which can cause annotation noise.

With the aforementioned rationale, we are ready to propose a well-defined human classifier f_H that is able to identify attribute noise in movie reviews. First, and in order to construct a robust classifier, f_H is an ensemble composed by three independent human classifiers f_{h1}, f_{h2} and f_{h3}. In other words, we will use three annotators to label a movie review x_i in terms of its polarity and attribute noise. Each annotator $j \in \{1, 2, 3\}$ is asked to classify the reviews in two levels. First, they are asked to make a prediction $\hat{y}_i^j$, i.e., to classify the polarity of the review x_i as *positive* or *negative* toward the movie in question. Second, they are asked to indicate whether they are *confident* or not about their classification $\hat{y}_i^j$. We denote the confidence of annotator j on review x_i by $c_i^j \in \{0, 1\}$, where $c_i^j = 1$ if j is confident and $c_i^j = 0$ otherwise. If $c_i^j = 0$, then we assume that x_i does not contain sufficient information for j to infer its polarity, that is, x_i has attribute noise of type *neutrality*. So, annotator j is asked to choose one label l_i^j that fits best to the *neutrality* attribute noise present in x_i, which can be either *neutral_mixed*, *neutral_factual* or *neutral_contextual*. If $c_i^j = 1$, then l_i^j is set to *"no noise"*. This process is illustrated in Figure 3.1. Of course, each annotator j cannot see the others' responses while giving their own annotations $\hat{y}_i^j$, c_i^j and l_i^j.

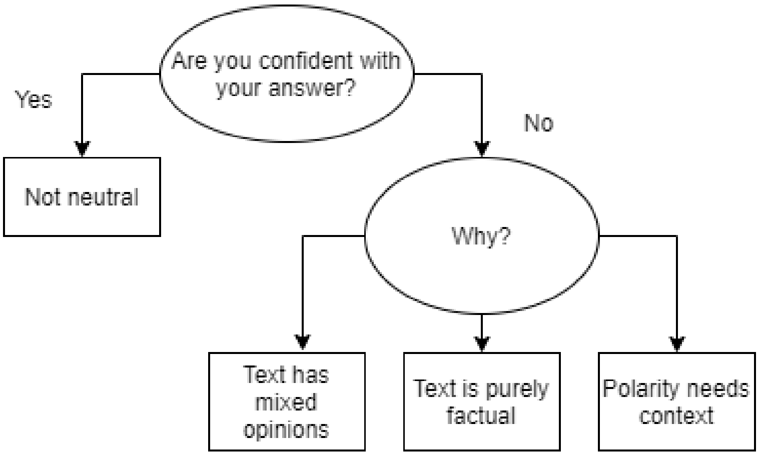

Figure 3.1: Confidence diagram.

At the end of this process, for each instance x_i, we will have three annotation triples $(\hat{y}_i^j, c_i^j, l_i^j)$, where $\hat{y}_i^j \in \{0, 1\}$ (*positive* or *negative*), $c_i^j \in \{0, 1\}$ (*not confident* or *confident*) and $l_i^j \in \mathcal{L}_N = \{$ *neutral_mixed*, *neutral_factual*, *neutral_contextual*, *"no noise"* $\}$. Assuming that all annotators are equally skilled, we aggregate these annotations using majority voting to set the outputs of our human classifier f_H. For the polarity $\hat{y}_i$ and the confidence c_i, the aggregation is straightforward, as described in Equations 3.1 and 3.2, respectively:

$$\hat{y}_i = \begin{cases} 0 & \textbf{if } \sum_{j=1}^{3} \hat{y}_i^j \leq 1 \\ 1, & \text{otherwise}, \end{cases} \tag{3.1}$$

$$c_i = \begin{cases} 0 & \textbf{if } \sum_{j=1}^{3} c_i^j \leq 1 \\ 1, & \text{otherwise}. \end{cases} \tag{3.2}$$

Setting the final attribute noise label l_i of review x_i is more involved. Let $\mathcal{L}_i = [l_i^1, l_i^2, l_i^3]$ be the **list** of labels l_i^j given by the annotators to review x_i (e.g. $\mathcal{L}_1 =$ [*neutral_mixed*, *neutral_mixed*, *"no noise"*]) and $N(l, \mathcal{L}_i)$ the number of elements of $\mathcal{L}_i$ that are equal to label l (e.g. $N(\textit{neutral\_mixed}, \mathcal{L}_1) = 2$). Then, l_i is the majority vote if at least two annotators gave that label to x_i and, if not, l_i is set to *neutral_undefined*, indicating no consensus. Thus, the label *neutral_undefined* only occurs when the third annotator does not agree with the first two annotators. This process is formally described by Equation 3.3:

$$l_i = \begin{cases} \arg\max_{l \in \mathcal{L}_N} N(l, \mathcal{L}_i) & \textbf{if } N(l, \mathcal{L}_i) \geq 2 \\ \textit{neutral\_undefined}, & \text{otherwise}. \end{cases} \tag{3.3}$$

Finally, when the human classifier is confident about its classification of x_i ($c_i = 1$), but it makes a mistake ($\hat{y}_i \neq y_i$), we update the label l_i of x_i to *discrepant*. In other words, the human classifier did not agree with the author's original score. Note that we are comparing the human classifier results c_i and $\hat{y}_i$, not each annotator results. It is easy to see that this update step will be executed only if l_i was previously set to *"no noise"*, i.e., it will not overwrite a *neutrality* label. Equation 3.4 formally defines the *discrepancy* update step:

$$l_i = \textit{discrepant} \ \textbf{ if } \hat{y}_i \neq y_i \ \textbf{ and } c_i = 1. \tag{3.4}$$

It is important to note that the noise labels are completely dependent on human classification. In other words, they are generated by the human classifier. Whenever the human classifier has doubts about an instance or mistakes it, it is considered noisy. Also, the other way around, every time an instance has noise is because the human classifier had doubts or made a mistake.

Chapter 4

Experimental Setup

This chapter presents the experimental setup to evaluate the impact of attribute noise on machine classifiers. The dataset collected from Metacritic is described in Section 4.1 and the three machine classifiers used in the analysis in Section 4.2. Then, in Section 4.3, we presented our model training.

4.1 Data Set

According to our problem formulation described in Section 3.1, it is important that the dataset $\mathcal{D}$ does not contain any movie reviews that have been explicitly associated with neutral scores. Because of that, we collected data from Metacritic[1], a website that publishes reviews about movies, television shows, music albums, video games and books. Movie reviews on Metacritic can be authored by *regular users* and *experts*, i.e., people working in the movie industry or important communication channels (e.g. *The New York Times*). In case of *experts*, the review provided by Metacritic is actually a short summary of the original review and, as we show in Chapter 5, this can be a problem for polarity classifiers. Also, each *experts* review is associated with a score ranging from 0 to 100, where scores from 0 to 39 are *negative*, from 40 to 60 are *neutral*, and from 61 to 100 are *positive*. Movies also have their *metascore*, an overall score ranging from 0 to 100 that is calculated from the expert's scores using a weighted average. On the other hand, reviews made by *regular users* are produced by any person that has an account and are associated with a score ranging from 0 to 10, where scores between 0 and 3 are *negative*, between 4 and 6 are *neutral*, and over 7 are *positive*. Because of these thick and well defined lines that separate *positive* from *negative* reviews, Metacritic is highly appropriate for the construction of $\mathcal{D}$.

[1] Available in: `https://www.metacritic.com/movie`. Accessed: 06-28-2020.

In total, we collected 415,867 reviews for 8,170 different movies, where 227,348 of those are from *regular users* and 188,519 from *experts*. Our data collection was executed using the following steps. First, we collected the most popular *experts* from the website, as provided by Metacritic [2]. Then, we generated a list of all movies reviewed by the top 10 experts. From this list, which contains 8,170 movies, we collected all reviews from *experts* and *regular users* that were posted until August, 2018, using the BeautifulSoup library [3]. For the purpose of this work, we avoided reviews that do not have a clear polarity (*neutral* reviews), i.e., we only considered *positive* and *negative* reviews.

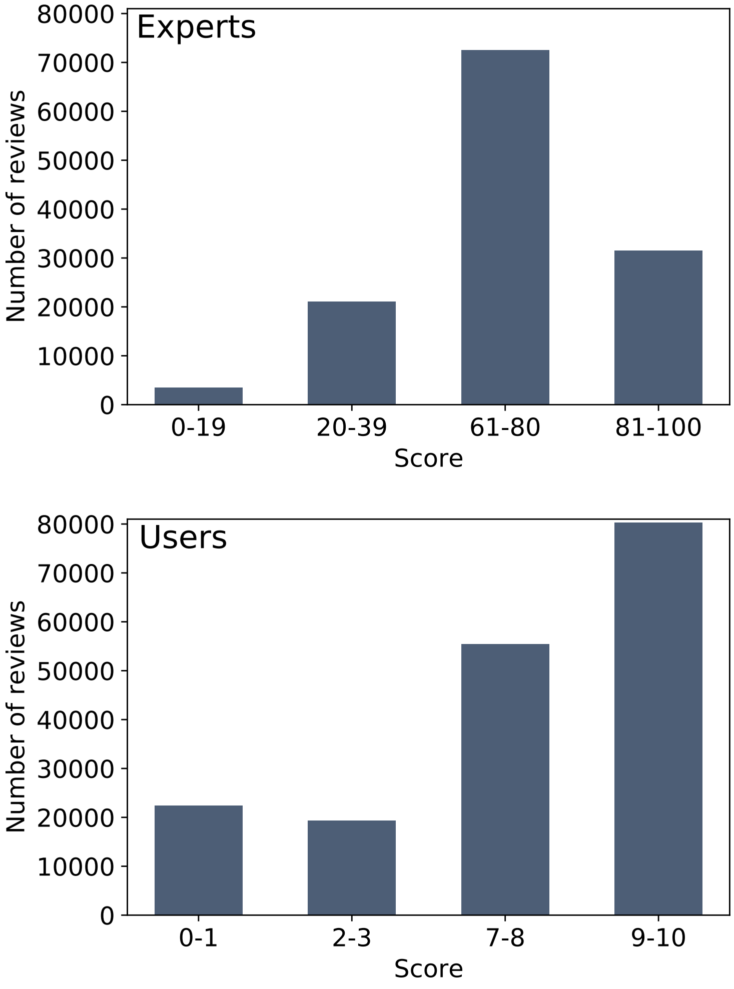

Figure 4.1: Score distribution.

Figure 4.1 shows the histogram of our data grouped by score and user type after

[2] Available in: `https://www.metacritic.com/browse/movies/critic/popular`. Accessed: 06-28-2020.

[3] Available in: `https://www.crummy.com/software/BeautifulSoup/bs4/doc/`. Accessed: 06/28/2020.

selecting non-neutral reviews with English text. Note that we have much more positive than negative reviews in both cases. In addition, usually a movie contains more *regular users* reviews than *experts*. For *experts*, there are more reviews with score between 61 and 80 and just a few with score between 0 and 19. While, for *regular users*, there are more reviews with score between 9 and 10 and the amount of reviews for the two negative groups is similar.

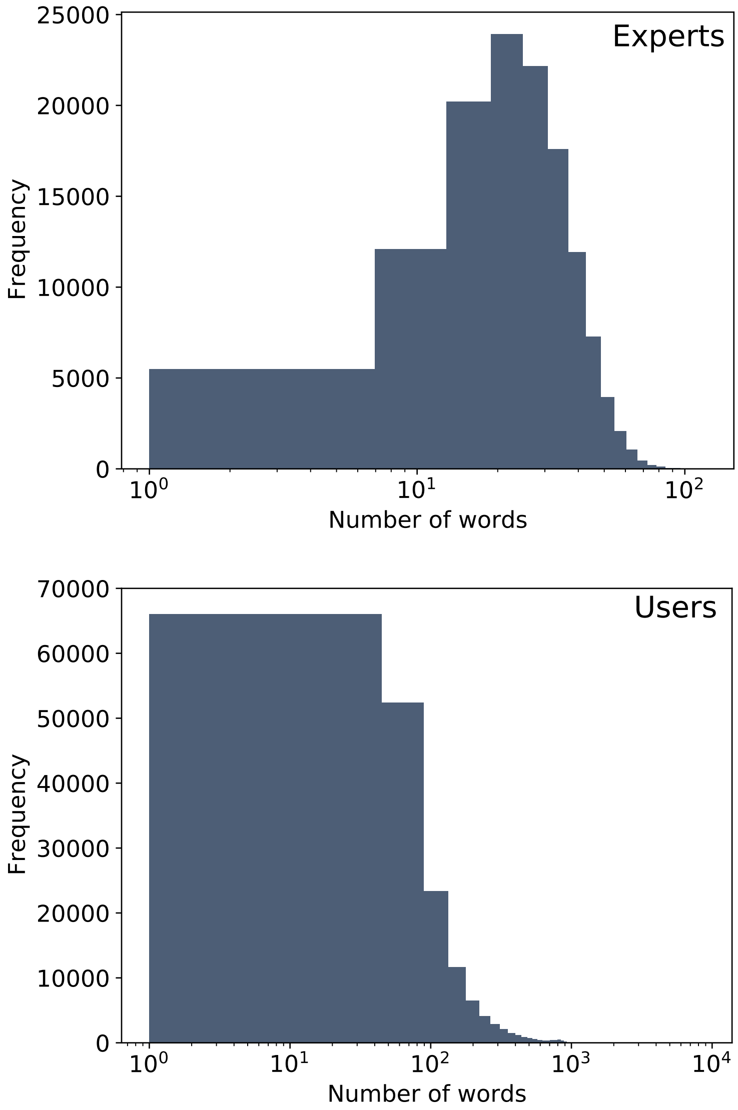

Figure 4.2: Number of words per review for *expert* and *regular user*.

As we mentioned, reviews from *experts* are actually a short summary of the original review. Thus, they are usually shorter than reviews from *regular users*, containing an average of 26 words (std. dev. of 13) against an average of 100 words (std. dev. of 129) for reviews by *regular users*. Figure 4.2 shows the histogram of the number of words. Note that the distribution is very different for *regular users* and *experts*. For *experts*, it is similar to a normal distribution which means that the number of words is symmetric in relation to the mean. For *regular users*, it is similar to a power law distribution. In other words, many reviews have up to 200 words and an smaller amount has more words than that. In addition, we observed that *experts* use a more elaborate language. We also calculated the entropy of words for both users. The entropy for *experts* is $11,08$ and $10,53$ for *regular users*. This result show that *experts* are more unpredictable than *regular users*. Because of these differences, we will condition our analyses on the type of user (*experts* or *regular users*) and score polarity (*positive* or *negative*).

4.2 Machine Classifiers

Our goal is to measure the impact of attribute noise in the performance of machine learning classifiers in the task of movie review polarity classification. That said, a fundamental step of our methodology is to choose state-of-the-art models that are able to detect the polarity of movie reviews, that is, to classify a review as positive or negative toward the movie in question. Thus, we selected three supervised deep learning architectures with reported success in the task of polarity detection of movie reviews: BERT [Devlin et al., 2019], CNN-GRU [Wang et al., 2016] and C-LSTM [Zhou et al., 2015].

The C-LSTM architecture is represented in Figure 4.3. It utilizes a CNN to extract a sequence of higher-level phrase representations, which are then fed into a LSTM unit to obtain the sentence representation. The network is initialized with pre-trained Word2vec vectors from Google News Dataset[4] and words not found in the vocabulary were initialized with a uniform distribution [-0.25, 0.25]. The output $\hat{y}_i$ is given by a dense layer with a sigmoid function. For regularization, we add two dropouts to prevent co-adaptation, one after the embedding layer and another after the LSTM layer.

Similarly, CNN-GRU, Figure 4.4, connects a CNN with a GRU to extract local and global features. Their model consists of an embedding layer initialized with Word2vec vectors from Google News Dataset in the same way as C-LSTM. In addition, two convolution layers with max-pooling. Then, a concatenate layer combining these two CNNs. To generate the sentence representation, the next layer is a GRU. Finally, their final representations are connected to two dense layers, with a relu and sigmoid

[4] Available at: `https://code.google.com/archive/p/word2vec/`. Accessed on 06/28/2020

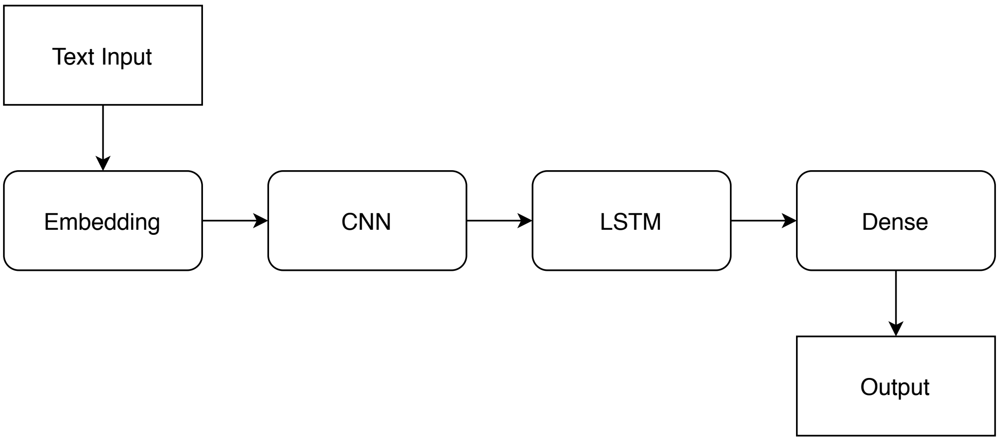

Figure 4.3: C-LSTM architecture.

function, respectively. For regularization, we add three dropouts: after the embedding layer, concatenate and second dense layer.

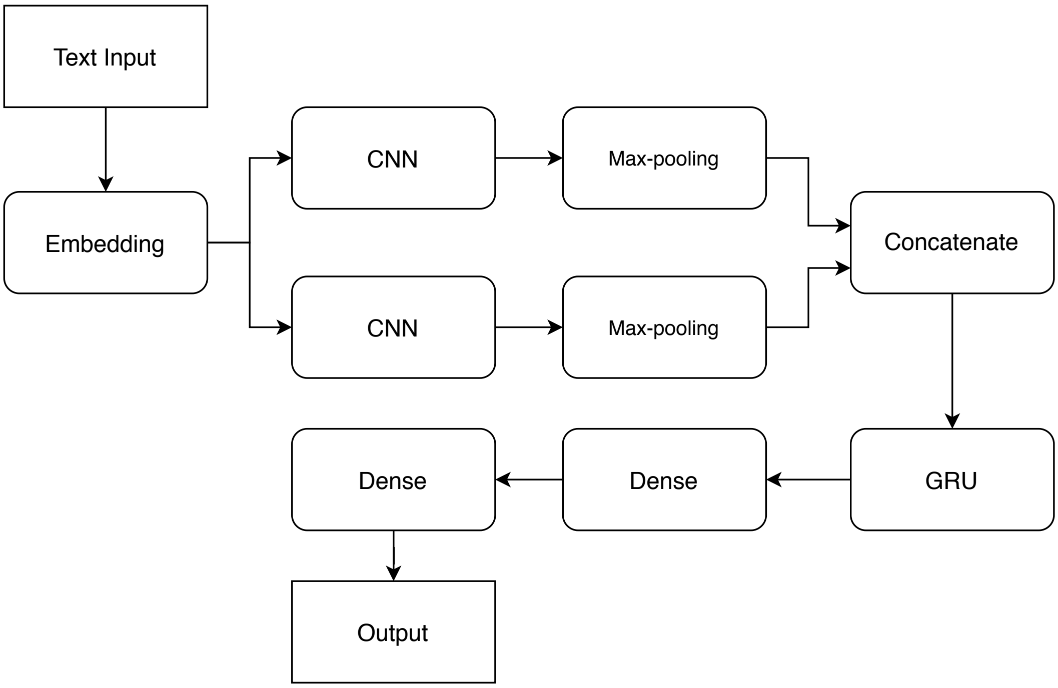

Figure 4.4: CNN-GRU architecture.

Finally, BERT uses a masked language model (MLM) to pre-train deep bidirectional representations from unlabeled text that considers both the left and right context of sentences and words. In this work, we used an architecture composed by BERT embeddings pre-trained with data from Wikipedia connected with two dense layers. The first dense layer used a relu function and the second one a sigmoid function.

The code of the three machine classifiers used in this work are publicly available

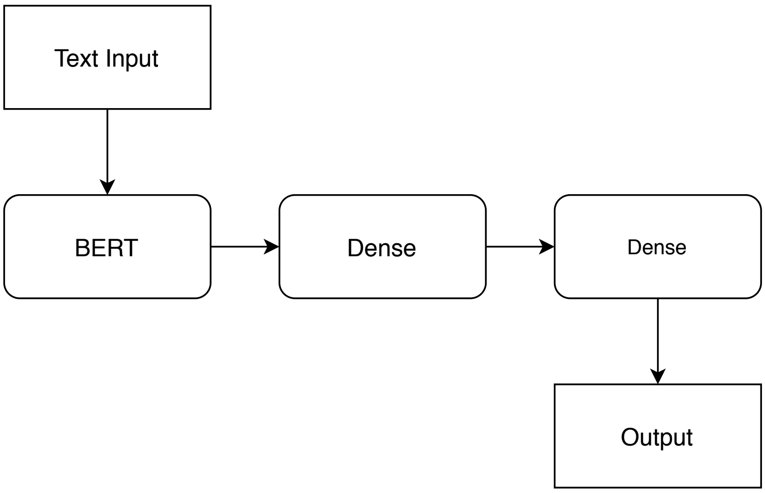

Figure 4.5: BERT architecture.

in the Internet. CNN-GRU and BERT were published by their authors and C-LSTM by researchers who used this method in their work Elaraby and Abdul-Mageed [2018]. We made small modifications in the codes so they are able to process our movie reviews data. We also created a log module to register all the results and changed the final output layer to a sigmoid function, since our problem is a binary classification. We also made BERT use the Keras library [5] just to facilitate our comparisons, but this is not a necessary step to reproduce our results. The link to each repository is listed bellow:

- C-LSTM: `https://github.com/EngSalem/TextClassification_Off_the_shelf`;
- CNN-GRU: `https://github.com/ultimate010/crnn`;
- BERT: `https://github.com/google-research/bert`;

4.3 Model Training

To train the machine classifiers, we randomly generated two balanced partitions of our data with the same size, one for *experts* and other for *regular users*. Each training dataset contains $4,398$ *positive* and $4,398$ *negative* reviews, for a total of $8,796$ reviews. It is important to note that these datasets do not contain any review labeled by the human classifier. After that, we performed a 5-fold cross-validation to choose the best hyperparameters for our data. The set of hyperparameter configurations we tested were the same used in the original articles [Wang et al., 2016], [Zhou et al., 2015] and

[5] Available at: https://faroit.github.io/keras-docs/1.0.1/. Accessed on 06/28/2020

[Devlin et al., 2019]. Since the BERT architecture is very simple, it has only a single hyperparameter, the batch size, for which we tested values of 16, 32 and 64. For C-LSTM, we tested layers with 100, 150 and 200 filters, and filters of size 2, 3 and 4, memory dimensions of size 100, 150 and 200, and batch size of 16, 32 and 64. Finally, for CNN-GRU, we tested layers with 100 and 200 filters, filters of size 3 and 4, GRU dimensionality of 100 and 150, pool sizes of 2 and 3, and batch sizes of 16 and 32. To run our experiments, we use a computer with the following configuration: 32 RAM, Intel Core i7 CPU 3.40 GHz and NVIDIA GeForce GTX GPU.

After executing cross-validation, we selected the best hyperparameters for each architecture and type of users comparing their F1-Score. We use 256 words for models trained with *expert* data and 512 for those trained with *regular user* data in all architectures. BERT achieved the best results using a batch size of 16 for both user types. For *experts*, C-LSTM uses a batch size of 32, 100 filters with size 3 in the convolutional layer, and 200 as memory dimension for LSTM. For *regular users*, the hyperparameters are the same, except in the LSTM layer, where a memory dimension of 100 was used. For *experts*, CNN-GRU uses 100 filters with size 5 as filter length and 3 as pool size for both CNNs. In the GRU, we used dimensionality of 150 and batch size of 16. For *regular users*, the differences are that we used a dimensionality of 100 in the GRU layer, size 3 as filter length and 2 as pool size for both CNNs. For both C-LSTM and CNN-GRU the differences in the hyperparameters are explained by the fact that our *expert* reviews are significantly shorter than the ones wrote by *regular users*.

After selecting the best hyperparameters, we trained two models for each architecture, one for *experts* and other for *regular users*. Also, each result reported in this work is the average of five runs, where for each run the model is trained from start using the whole training dataset. With that, we can measure their parameter sensitivity and calculate confidence intervals for the results. In addition, C-LSTM and CNN-GRU took approximately half a day to train and BERT one day. Finally, we noted that the performance of all three models were not significantly affected by the hyperparameter configurations we tested.

Chapter 5

Results

In this Chapter we quantify the presence of attribute noise in movie reviews (Section 5.1) and its impact on the task of polarity classification (Section 5.2). We also created additional experiments (Section 5.3).

5.1 Amount of Attribute Noise

The first question we need to answer is: how much attribute noise exists in movie reviews? In the context of our Metacritic dataset $\mathcal{D}$, the answer to this question can be influenced by two factors: (1) the type of user and (2) the polarity of their rating. Thus, the following results are conditioned on whether the authors are *experts* or *regular users* and whether the reviews are *positive* or *negative*. Because of that, we sampled a collection D_H of 800 movie reviews from $\mathcal{D}$ that is both balanced in terms of user type and score polarity, i.e., this collection has 200 reviews for each of the four combinations of user type and score polarity.

In order to quantify the amount of attribute noise in D_H, we use our proposed human classifier f_H described in Section 3.3 to label every review $x_i \in D_H$. Since this annotation process is very expensive, in this work, only two annotators were used initially. The third annotator was called to classify instance x_i if, and only if, the first two had any kind of disagreement, i.e., a disagreement regarding the polarity y_i, the confidence c_i, or attribute noise label l_i. In order not to influence the third annotator only with difficult instances, we randomly select other reviews and mix with the disagreement instances. As annotators, we selected three people who are fluent in the English language. Despite not having practice in movie labeling, they often watch movies. Again, each annotator cannot see the others' responses while giving their own annotations.

Recall that f_H assigns a polarity $\hat{y}_i \in \{\textit{positive}, \textit{ negative }\}$ to x_i and, more important to our purpose here, a label l_i, which can be either *"no noise"* (absence of attribute noise), *discrepant* (the polarity of the text is different from the score polarity), or one of the four *neutrality* labels: *neutral_ mixed* (text has mixed opinions), *neutral_factual* (text is purely factual), *neutral_ contextual* (polarity needs context) and *neutral_ undefined* (reasons are unclear). Also, let $u_i \in \{\textit{expert}, \textit{ regular user }\}$ be the user type of the author of review x_i. Our goal with the following results is to estimate the probability $P(l_i = l \mid y_i = y, u_i = u)$ for the four combinations of score polarity y and user type u.

label (l_i)	***positive***	***negative***	**total**
experts			
no noise	146(36.5%)	162(40.5%)	**77%**
discrepant	20(5%)	3(0.8%)	**5.8%**
neutral	34(8.5%)	35(8.8%)	**17.3%**
mixed	10(2.5%)	7(1.8%)	**4.3%**
factual	14(3.5%)	3(0.8%)	**4.3%**
contextual	7(1.8%)	20(5%)	**6.8%**
undefined	3(0.8%)	5(1.3%)	**2%**
regular users			
no noise	177(44.3%)	187(46.8%)	**91%**
discrepant	3(0.8%)	2(0.5%)	**1.3%**
neutral	20(5%)	11(2.8%)	**7.8%**
mixed	16(4%)	7(1.8%)	**5.8%**
factual	1(0.3%)	2(0.5%)	**0.8%**
contextual	0(0%)	1(0.3%)	**0.3%**
undefined	3(0.8%)	1(0.3%)	**1%**

Table 5.1: Amount of attribute noise in reviews.

In Table 5.1 we show the number and proportion of movie reviews with and without attribute noise for *experts*. From the 400 labeled reviews, almost one quarter (92) contains attribute noise. From those, note that *neutral* reviews are more common than *discrepant* ones, but while the first is equally present in both *positive* and *negative* reviews, *discrepant* noise is significantly more present in *positive* reviews. In such cases, the author gave a positive score to the movie, but its review demonstrates the opposite sentiment. This often occurs when the *expert* is using the review to justify a good, but far from perfect score, to a critically acclaimed movie. As for the *neutral* reviews, the most predominant type is *neutral_ contextual* (6.8%), followed equally by *neutral_ mixed* (4.3%) and *neutral_factual* (4.3%). Also, *neutral_ contextual* noise is more common in *negative* reviews, when *experts* often use figures of speech (e.g. irony, simile) together with external knowledge to create humour. In the example listed in Table 5.2, the *expert* uses irony in the review. Finally, *neutral_factual* noise is more present in *positive* reviews, where often the *experts* simply describe some characteristic of the movie that impressed them without explicitly saying that. Table 5.2 shows real examples of reviews posted by *experts* for each type of attribute noise.

class label (l_i)	**Example**
discrepant	"Figgis's film doesn't match its reach." (Positive)
mixed	"Pleasant but dull formula film." (Negative)
factual	"Without trivializing the disease, the film challenges AIDS' stigma (albeit for heterosexuals) at a moment when it was still considered a death sentence." (Positive)
contextual	"Disheveled tripe pieced together with the good intentions." (Negative)
undefined	"More interesting as history, re-written, than as the moral parable this true story became." (Positive)

Table 5.2: Examples of *experts* reviews with attribute noise.

Also, in Table 5.1 we show the number and proportion of movie reviews with and without attribute noise for *regular users*. First, note that the number of reviews with attribute noise significantly decreased in comparison with the ones written by *experts*. From the 400 labeled reviews, only 36(9%) contains attribute noise, of which 31 are *neutral* and only 5 are *discrepant*. Different from what was verified for *experts*, the most predominant noise label for *regular users* was *neutral_mixed*, which occurred significantly more in *positive* reviews. For the other labels, their occurrences were fairly balanced between *negative* and *positive* reviews. We observed that *regular users* use a much more direct and simple language to state their opinions than *experts*. Because of that, most of the attribute noise is concentrated in cases where the author lists both the negative and positive aspects of the movie without stating their final opinions about the movie, which is the definition of *neutral_mixed*. Table 5.3 shows real examples of reviews posted by *regular users* for each type of attribute noise.

A note about the human classifier. For the first two annotators, they agreed on 91.13% of the polarity scores, on 90.5% of their confidence levels and on 88% of their attribute noise labels. Regarding the third annotator, only 1.5% of the instances were not in total agreement with at least one of the first two annotators. The Cohen's kappa coefficient for the first two annotators was 0.82 in relation to polarity scores, 0.58 on their confidence levels and 0.49 on their attribute noise labels.

class label (l_i)	**Example**
discrepant	“The actors try their best with the lines they are given, but the "movie about a real bank robbery" is on auto-pilot most of the time. It greatly resembles a 70’s film by letting the characters drive the story. As a result there’s a lot of dialog. But its not very interesting dialog. It is an instantly forgettable film.” (Positive)
mixed	“I think the director did an incredible job. I loved the way it was shot. The scifi world they created was also awesome. But I think the story was way too subtle and wasn’t clear enough.” (Positive)
factual	“(...) The 1953 film about a provincial old couple’s pilgrimage to the big city provokes sympathy for the mother and father, who are so frail, so gentle, and yet are treated so badly by their urbanized son and daughter. (...)” (Positive)
contextual	“Only go if you’re interested in seeing Bening’s new facelift.” (Negative)
undefined	“Wow, can’t believe the critics on this one.” (Positive)

Table 5.3: Examples of *regular users* reviews with attribute noise.

5.2 Impact of Attribute Noise

In this section, we quantify the impact of attribute noise in machine classifiers. Also, by putting these results in perspective with what was achieved by the human classifier, we hope to provide an accurate assessment on how distant machine classifiers are with respect to human performance in the task of polarity detection of movie reviews. We guide our analyses by the following questions:

1. What are the probabilities of a correct and a misclassification given the label l? In other words, we want to estimate the probabilities $P(\hat{y}_i = y_i \mid l_i = l)$ and $P(\hat{y}_i \neq y_i \mid l_i = l)$ for all labels $l \in L$.
2. What are the probabilities of label l given that the classifier was correct and that it made a mistake? In other words, we want to estimate the probabilities $P(l_i = l \mid \hat{y}_i \neq y_i)$ and $P(l_i = l \mid \hat{y}_i = y_i)$ for all labels $l \in L$.

To address these questions, we test the three classifiers described in Chapter 4 in the labeled dataset D_H (see Section 5.1), which contains 800 reviews. Because this labeled dataset is completely balanced, we created two balanced training datasets, one containing solely reviews from *experts*, namely $D_T^{experts}$, and another containing solely reviews from *regular users*, namely D_T^{users}. Each training dataset contains $8,796$ reviews, $4,398$ of each polarity. Again, this dataset is completely balanced and solely

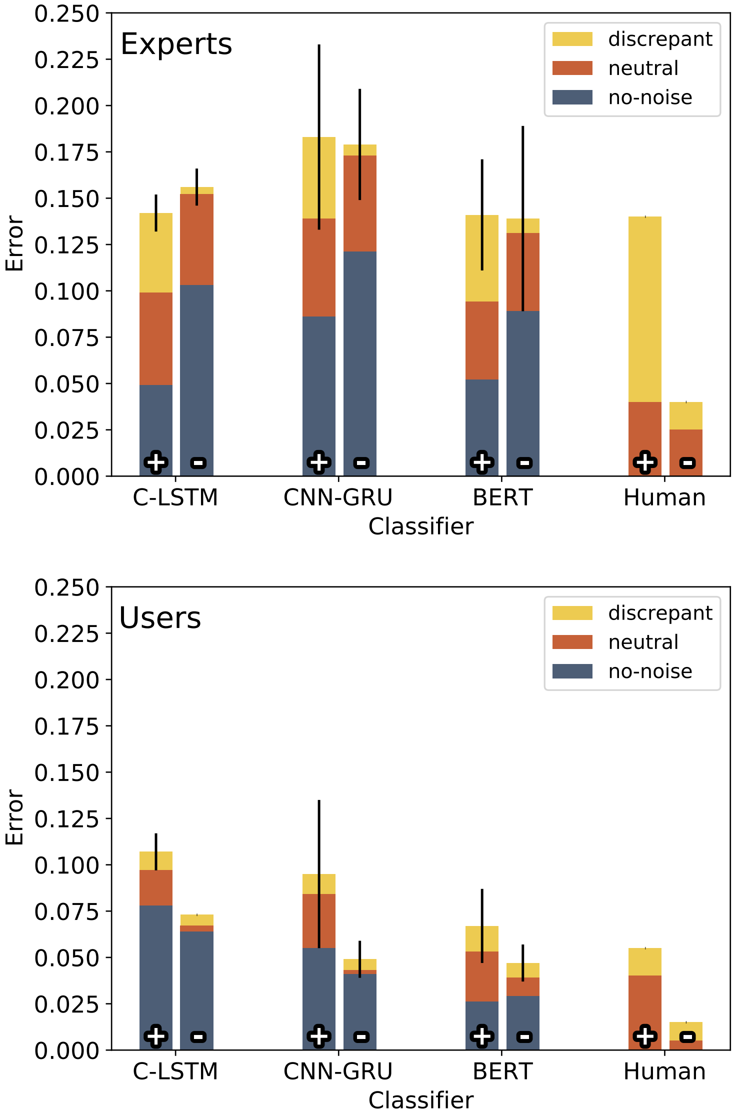

Figure 5.1: Classification error for all classifiers.

used to train the machine classifiers. Because these classifiers are sensitive to initialization parameters, we trained and tested them 5 times and the corresponding error bars are shown in Figure 5.1. Finally, recall that y_i refers to the author's original polarity score and $\hat{y}_i$ refers to the polarity predicted by the classifiers, including the human classifier. Thus, our golden standard is the original score given by the author, not the human classifier polarity.

Figure 5.1 shows the classification error (with their respective error bars) for the three machine classifiers and for the human classifier in the labeled dataset D_H. The classification error is simply the proportion of instances that were misclassified. Each bar is also colored according to the labels' proportion in the misclassified instances. For each classifier, the left (right) bar shows the error with respect to *positive* (*negative*) instances. In general, the human classifier was the one that achieved the smallest error, followed by BERT and C-LSTM. Also, the errors are always higher for *experts*, as these reviews have significantly less words (see Section 4.1) and much more noise. The latter is also one of the main reasons for the error being almost always higher for *positive* instances than for *negative* instances. For *expert* reviews, while *negative* instances always have more *"no noise"* instances, *positive* instances have almost twice more noisy instances, particularly *discrepant* ones. For *regular user* reviews, *positive* instances also have more noisy instances, but the difference in terms of *neutral* reviews is more significant. Note that, for both *experts* and *regular users*, this difference in the instances misclassified by the human classifier is striking.

For a more precise assessment of the impact of attribute noise, we show in Table 5.4 the accuracy of the classifiers considering instances of each label separately. In other words, these results provide estimates for the probabilities of our first question, $P(\hat{y}_i = y_i \mid l_i = l)$ and $P(\hat{y}_i \neq y_i \mid l_i = l)$. First, note that for all classifiers the accuracy significantly degrades in instances with *neutral* noise and get even worse in instances with *discrepant* noise. Recall that a *discrepant* review is a review where the human classifier was sure about its polarity, but the originally assigned polarity is the opposite. Thus, by definition, the human classifier accuracy on *discrepant* reviews is zero. For *neutral* instances, the human classifier always outperforms the machine classifiers. However, the machine classifiers are not always tricked by *discrepant* reviews as the human classifier is, although their performances are not better than a coin toss. Considering the specific *neutral* labels, note that BERT achieves human level performance for *neutral_ contextual*, which is coherent with the nature of this classifier, given that its embeddings are supposed to carry much more contextual information in comparison with the embeddings used in C-LSTM and CNN-GRU. The most inconclusive results refer to *neutral_ undefined*, which is also the label with the least instances, 12 out of 800.

	C-LSTM	CNN-GRU	BERT	Human
no noise	0.91	0.91	0.94	**1**
discrepant	**0.55**	0.52	0.45	0
neutral	0.76	0.73	0.76	**0.78**
mixed	0.75	0.72	**0.76**	0.75
factual	0.78	0.77	0.71	**0.80**
contextual	0.67	0.64	**0.79**	**0.79**
undefined	**0.97**	0.90	0.77	0.83

Table 5.4: Accuracy of the classifiers considering only instances of a particular label.

To answer our second question, related to the probabilities $P(l_i = l \mid \hat{y}_i \neq y_i)$ and $P(l_i = l \mid \hat{y}_i = y_i)$, we sample an additional dataset D_H^{error} to be labeled by our human classifier f_H. First, we run the BERT classifier, which was the one that achieved the best results, on two new balanced sets of reviews extracted from $\mathcal{D}$, one containing $2,752$ reviews from *experts* and the other $2,752$ reviews from *regular users*. Again, we used the same BERT classifiers that were trained for generating the results in Figure 5.1, one for each user type. After running BERT, we construct D_H^{error} by sampling 100 misclassified and 100 correctly classified instances authored by each user type, for a total of 400 reviews. Then, we run f_H on D_H^{error} to have a more accurate estimate of $P(l_i = l \mid \hat{y}_i \neq y_i)$ and $P(l_i = l \mid \hat{y}_i = y_i)$.

Table 5.5 shows the percentages of each label for correctly and incorrectly classified instances, which provide estimates for the probabilities of $P(l_i = l \mid \hat{y}_i \neq y_i)$ and $P(l_i = l \mid \hat{y}_i = y_i)$. For both *experts* and *regular users*, it is much more likely to find *neutral* and *discrepant* reviews in misclassified instances. In other words, one easy way to find instances with attribute noise in movie reviews is to run BERT and sample from misclassified instances. Our estimates for the probabilities of finding a misclassified instance with attribute noise is 0.64 for *experts* and 0.56 for *regular users*. Recall from Table 5.1 that we found only 23% of instances with attribute noise in reviews from *experts* and only 9% in reviews from *regular users* in our first balanced sample D_H. The most striking difference is for *discrepant* reviews, where the number of instances increased by one order of magnitude in misclassified instances. Regarding the *neutral* labels, our results reveal that we are at least twice as likely to find *neutral_contextual* noise in misclassified *expert* reviews and *neutral_mixed* noise in misclassified *regular users* reviews.

We investigated misclassified *"no noise"* instances and found two patterns that explain the errors. First, reviews that have positive and negative points, but where humans can easily identify what side has the most weight. Second, reviews that have some "irony" that is clear to humans, but is created using words with the opposite polarity of the final score y_i. Table 5.6 shows real examples of misclassified *"no noise"* reviews with their original polarities given by their authors. The first and last review belong to the second pattern. While, the second review belongs to the first one.

We further investigate these patterns by using SHAP [Lundberg and Lee, 2017], which is a game theoretic approach to explain the output of deep learning models and designed to understand the most important words for the machine classifiers. Figure 5.4 shows the result for the last review in Table 5.6. The words are plotted in descending order according with their importance. Note that all listed words have a positive polarity when they are analyzed separately. As a result, their combination contributes

label (l_i)	$\hat{\mathbf{y}}_\mathbf{i} = \mathbf{y}_\mathbf{i}$	$\hat{\mathbf{y}}_\mathbf{i} \neq \mathbf{y}_\mathbf{i}$
experts		
no noise	96 (78%)	28 (36%)
discrepant	1 (1%)	19 (25%)
neutral	26 (21%)	30 (39%)
mixed	10 (8%)	9 (11%)
factual	6 (5%)	3 (4%)
contextual	8 (6%)	11 (14%)
undefined	2 (2%)	7 (9%)
regular users		
no noise	111 (86%)	31 (44%)
discrepant	2 (2%)	14 (19%)
neutral	16 (12%)	26 (37%)
mixed	13 (10%)	15 (21%)
factual	0 (0%)	1 (1%)
contextual	1 (1%)	6 (8%)
undefined	1 (1%)	5 (6%)

Table 5.5: Percentage of labels in correct ($\hat{y}_i = y_i$) and incorrect ($\hat{y}_i \neq y_i$) predictions by BERT.

Examples
"Michael Bay may think that special effects can substitute for good acting and a good story, but that does not fly around here." (Negative)
"Fun movie if you can suspend your disbelief enough to sit through it. Plot breaks no new ground which means you basically know what you are getting as soon as you walk in to it. Ice Cube and Charlie Day were great at playing extensions of themselves and theres alot of laughs to be had whenever either of them are on screen." (Positive)
"The trailer was promising to me; I expected it to be a really good movie, but instead it was "meh". I didn't really like Cruz; it was heartwarming how Lightning McQueen made a tribute to Doc at the end, but the trailer made it seem action packed; it wasn't as good as I expected." (Negative)

Table 5.6: Examples of misclassified *"no noise"* reviews.

to the classifier misclassify the review. Differently, the classifier was uncertain for the first review. However, the word "good" appeared twice and helped to slightly classify the review as positive which can be seen Figure 5.2. The second review was a little divergent. Although, the most important word was "great", it was not enough to the classifier consider the review as positive. The classifier got confused with the first and second sentence. Nevertheless, it also got confused with a name. Results can be seen in Figure 5.3.

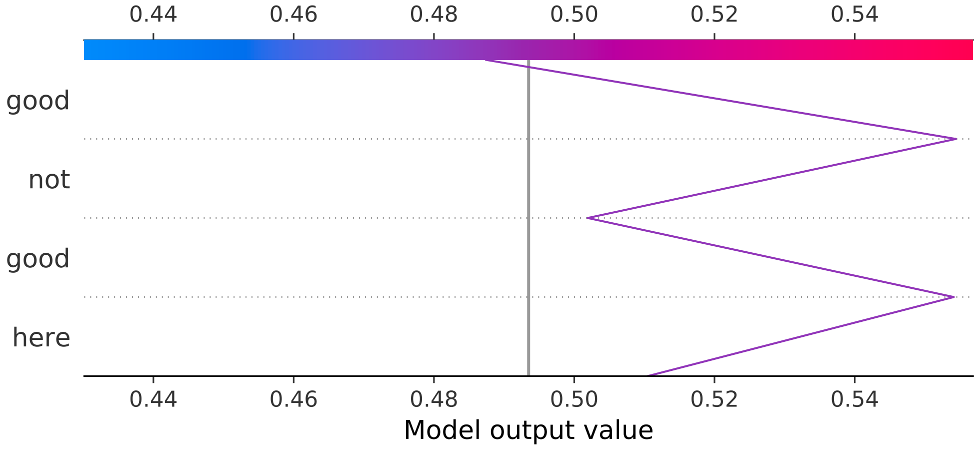

Figure 5.2: SHAP plot for the first review in Table 5.6.

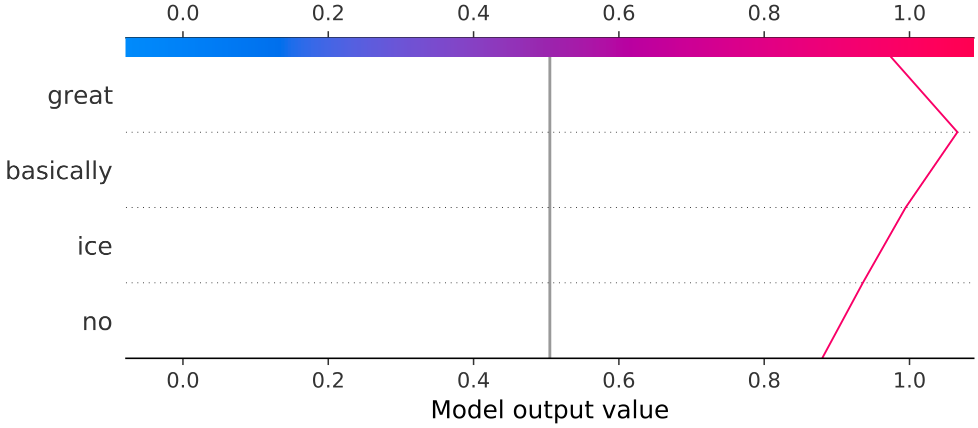

Figure 5.3: SHAP plot for the second review in Table 5.6.

We conjecture that these instances can be correctly classified with extra training and more modern (and complex) architectures. On the other hand, we feel that dealing with instances with attribute noise is not that simple, where more guided and focused approaches are probably needed, such as the one proposed by Valdivia et al. [2019].

5.3 Additional Experiments

We also created additional experiments to further understand the *neutral* and *discrepant* reviews. In Section 5.3.1, we analyzed the BERT output for *neutral* and *discrepant* reviews. Next, in Section 5.3.2, we trained a new model to classify whether a review is neutral or not. Finally, in Section 5.3.3 we analyze a case of study about Star Wars.

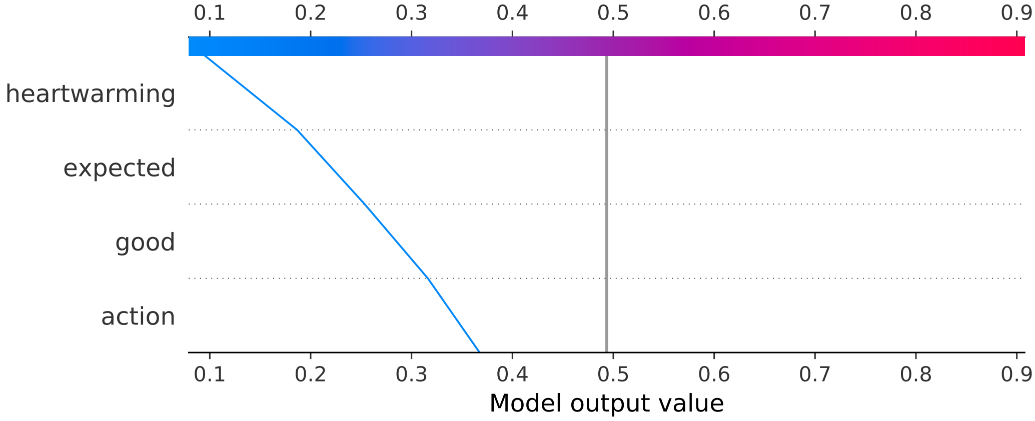

Figure 5.4: SHAP plot for the last review in Table 5.6.

5.3.1 BERT output for *neutral* and *discrepant* reviews

In order to further understand how the classifiers work with *discrepant* and *neutral* reviews, we analyze the BERT output value for these reviews. Since our BERT classifier output is generated by a sigmoid function, the output is a value between 0 and 1, where 0 is positive and 1 is negative. Thus, values close to 0.5 means that the classifier was not sure about the polarity attribute for the review. From that, we used a metric to evaluate how uncertain the classifier was for *discrepant* and *neutral* reviews. For each output, we applied the following:

$$s = 2 * |score - 0.5|. \tag{5.1}$$

From Equation 5.1, the output is uncertain, if it is close to 0, or certain if it is close to 1. The results are shown in Figure 5.5. Note that all medians are above 0.6. For *regular users* reviews, the classifier is more confident. In addition, *neutral* reviews from *experts* generate more uncertainty than the others. Considering the users separately, *neutral* reviews create more distrust than *discrepant*.

We also analyzed a box plot containing only misclassified reviews. The results are represented in Figure 5.6. Unlike the previous result, *neutral* reviews from *experts* have a median less than 0.5. Moreover, the median for *neutral* reviews decreased slightly. On the other hand, the median for *discrepant* reviews increased moderately. In other words, the classifier is more confident about misclassified *discrepant* reviews and less confident about misclassified *neutral* reviews.

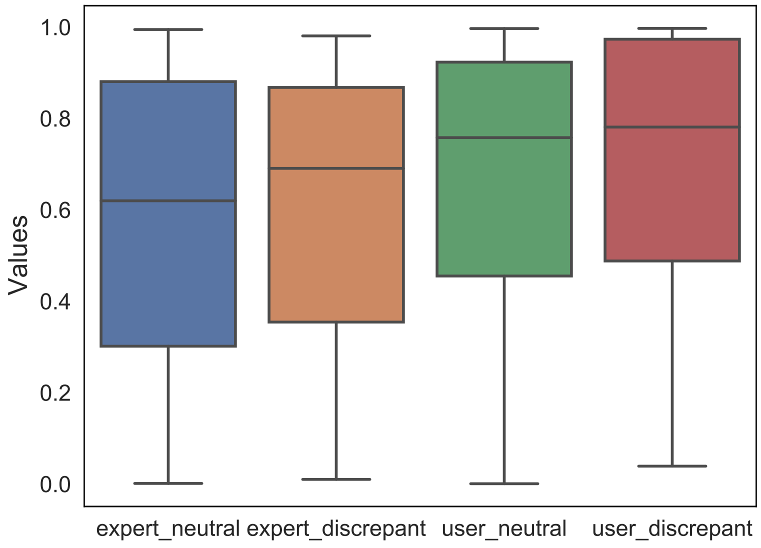

Figure 5.5: BERT output for *neutral* and *discrepant* reviews.

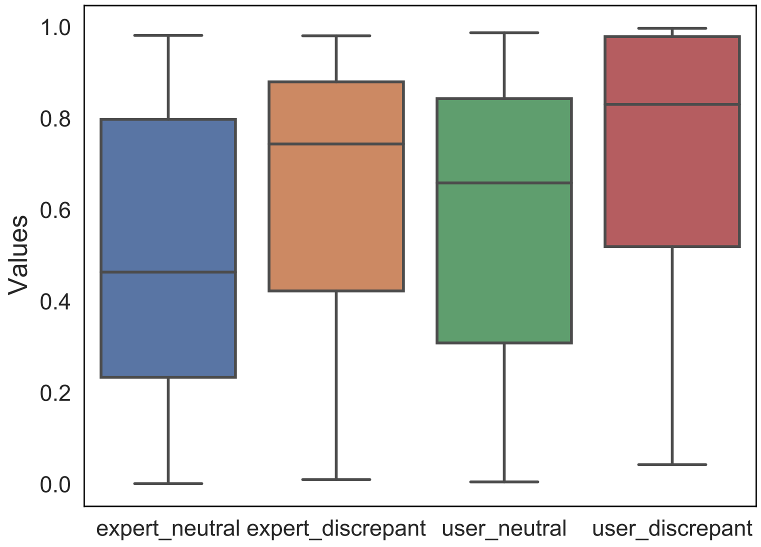

Figure 5.6: BERT output for *neutral* and *discrepant* misclassified reviews.

5.3.2 Neutral classifier

In order to identify reviews with neutral noise, we trained a new BERT classifier to categorize reviews as neutral or non-neutral, using our labeled collection. To do that, we randomly created two balanced training datasets, one containing solely reviews from *experts* and another containing solely reviews from *regular users*. Each dataset contains 4, 000 neutral and 4, 000 non-neutral reviews. For neutral, we selected reviews with score between $40-60$ (*expert*), and $4-6$ (*regular user*). For non-neutral, between $0-39$ and $61-100$ (*expert*), and $0-3$ and $7-10$ (*regular user*). To test the new classifiers, we generated five test sets for each user. Each test set contains the same amount of *neutral* and *"no noise"* reviews, for each neutrality label. The amount of reviews in each class for each combination is defined in Table 5.7.

	expert	*regular user*
neutral + no noise	124	73
mixed + no noise	36	52
factual + no noise	32	4
contextual + no noise	42	8
undefined + no noise	14	9

Table 5.7: Amount of *neutral* and *"no noise"* review in each combination.

Our goal is to evaluate if the classifier is able to correctly classify the reviews with neutrality labels as neutral and *"no noise"* as non-neutral. The results are shown in Table 5.8. Note that the *neutral_mixed* and *"no noise"* have the best F1 Score for *experts*. On the other hand, *neutral_contextual* reviews has the worst result. In additional, the other results were around 0.5. Considering *regular users*, the results were higher, except for *neutral_contextual* reviews. Furthermore, *neutral_mixed* and *"no noise"* also have the best F1 Score.

In order to reduce the uncertainty of *neutral* reviews, we trained other two BERT classifiers, one for *experts* and other for *regular users*. Instead of using training datasets created from a range of neutral scores like in the previous analysis, for these classifiers, the training datasets contain only neutral reviews with score 50 (*experts*) and 5 (*regular users*). The results are also shown in Table 5.8. Note that, comparing with the previous results, the F1 Score improved for most of the test sets for *experts* and *regular users*. Moreover, the neutral class had more impact than the non-neutral. In other words, the classifier is able to hit more neutral and non-neutral reviews when the classifier is trained with a smaller range of scores. The only test sets that got worse F1 Scores were *neutral_mixed* for *experts* and *neutral_factual* for *regular users*.

Additionally, we plotted the confusion matrix. The confusion matrices for *experts* is shown in Figure 5.7. Note that the classifier trained with smaller range of scores (neutral score 50) improved the true positive results. However, true negative decreased a little. The same happened with *regular users* as we can observe in Figure 5.8.

experts				
Neutral score	40-60		50	
Class	Neutral	Non-neutral	Neutral	Non-neutral
neutral + no noise	0.53	0.58	0.58	0.59
mixed + no noise	0.71	0.63	0.67	0.67
factual + no noise	0.45	0.56	0.53	0.62
contextual + no noise	0.4	0.59	0.6	0.64
undefined + no noise	0.5	0.5	0.55	0.51
regular users				
Neutral score	4-6		5	
Class	Neutral	Non-neutral	Neutral	Non-neutral
neutral + no noise	0.63	0.7	0.71	0.73
mixed + no noise	0.71	0.74	0.79	0.76
factual + no noise	0.67	0.8	0.4	0.73
contextual + no noise	0.14	0.33	0.18	0.57
undefined + no noise	0.5	0.6	0.5	0.6

Table 5.8: Neutral classifiers F1 Score results.

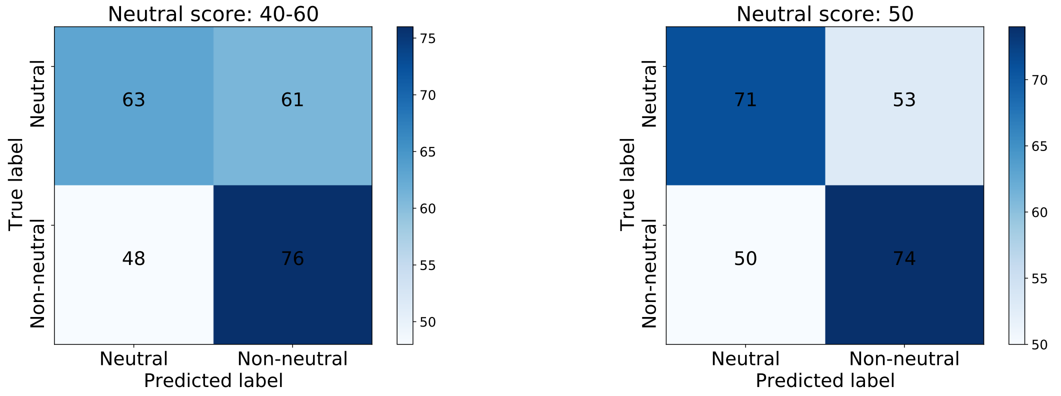

Figure 5.7: Confusion matrix for *experts*.

5.3.3 Star Wars

We mentioned previously that a complex example of attribute noise is a review of a highly acclaimed movie where only negative points are highlighted to justify a positive but different from perfect score. During our experiments, we noticed that the classifier usually misses more often reviews from Star Wars movies. These reviews are usually very long, with average number of words of 37 for *experts* and 142 for *regular users*. From these insights, we decided to further investigate the relationship between the number of words per review and the scores. To do that, we used positive and negative reviews from all nine Star Wars movies. Altogether there are 7, 458 reviews from *regular*

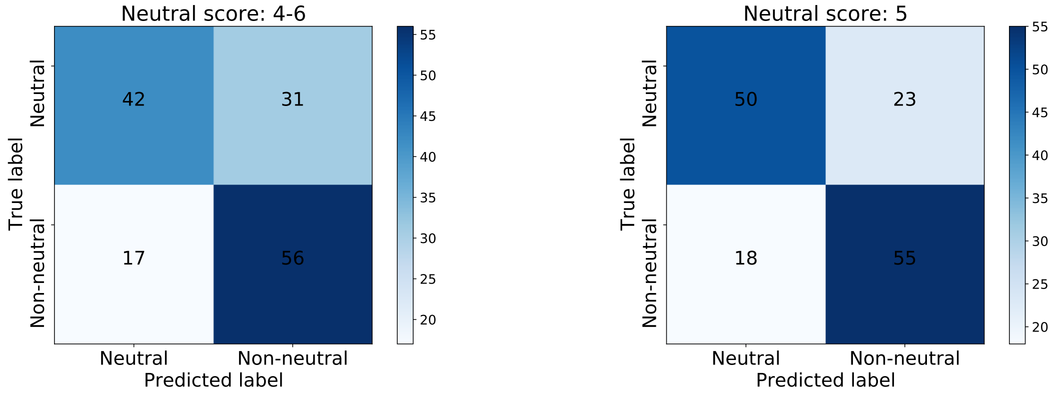

Figure 5.8: Confusion matrix for *regular users*.

users and 257 from *experts*.

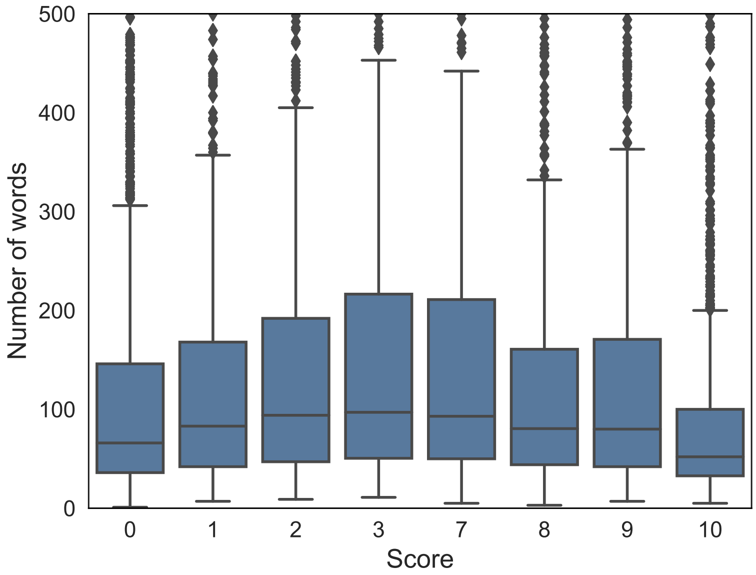

Figure 5.9: Box plot of the number of words per review grouped by score for *regular users*.

The box plot for *regular users* is shown in Figure 5.9. Considering only positive scores, 7, 8, 9 and 10, note that as the scores decreases, the median of the number of words per review increases. In other words, *regular users* usually use more words to justify their positive but not perfect score. On the other hand, for negative scores, 0, 1, 2 and 3, as the scores increases, the median of the number of words per review also increases. The higher the negative score, the greater is the number of words per review. That is, *regular users* usually use a smaller number of words to write about a review very negative. For *experts*, the box plot is shown in Figure 5.10. The number

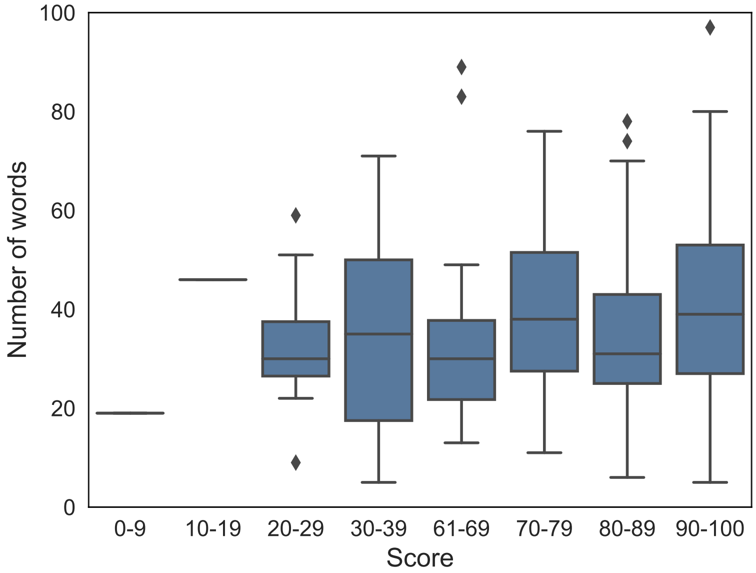

Figure 5.10: Box plot of the number of words per review grouped by score for *experts*.

of reviews written by *experts* is very small, specially comparing with the number of *regular users* reviews. For this reason, it is not possible to analyze some scores such as $0-9$ and $10-19$. Moreover, the pattern noted earlier for *regular users* was not observed for *experts*. However, we can note that the median is greater for positive scores.

The pattern found for *regular users* shows that they are more straightforward to write reviews with perfect positive or negative scores. In other words, they use a smaller number of words. Due to the fact that these reviews usually are more direct and have less context, this form of writing facilitates the understanding of the polarity for machine classifiers. On the other hand, reviews with a greater number of words tend to have more context, mixed opinions and lack of clarity. Consequently, they are more prone of being noisy.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

One of the main criticisms about ML is its lack of explainability for successes and failures [Lipton and Steinhardt, 2019]. In NLP tasks solved by deep learning architectures, this is an even bigger problem, since features are usually encoded into dense vector embeddings that are difficult to interpret [Pelevina et al., 2016]. This work helps to fill this gap by proposing a methodology to characterize, identify and measure the impact of problematic instances in the task of polarity classification of movie reviews. We characterized such instances by two types of attribute noise: *neutrality*, where the textual review does not clearly convey a polarity, and *discrepancy*, where the polarity of the text does not match the polarity of its rating.

Our methodology is summarized in the creation of a human classifier capable of assigning a label to a movie review to indicate whether it has noise or not. The human classifier is composed of three independent human annotators. The first two annotators classified each review in two levels and the third one was called to classify instances in case of disagreement between the first two. Initially, they determined if the review is positive or negative. Then, they declared their reliability in relation to the polarity of the review. If they were not confident, they also chose a reason. To aggregate the classification of the three annotators and create our human classifier, we used the majority vote. Basically, when the human classifier was not confident about its prediction, we labeled the review as *neutral*. Moreover, if the review was no longer classified as neutral and the class assigned by the human classifier was different from the author's rating, we labeled the review as *discrepant*. To test the human classifier, we collected movie reviews posted on Metacritic from *experts* and *regular users*. Altogether, the human classifier classified $1,200$ reviews and found 198 *neutral* and 64 *discrepant*. Finally, we selected three state-of-the-art machine learning classifiers to test on these reviews and measured the impact of *neutral* and *discrepant* reviews.

We analysed the amount of attribute noise that exists in movie reviews considering the influence of two factors: type of user and the polarity of their rating. Results showed that *neutral* reviews are more present for both type of users. After that, we also analysed their influence on human and machine classifiers. We answered two questions: "What are the probabilities of a correct and a miss classification given the label l?" and "What are the probabilities of label l given that the classifier was correct and that it made a mistake?". From this perspective, we provided empirical evidence about the need to pay attention to instances with attribute noise, as they are much harder to be classified, for both machine and human classifiers. We also showed that one easy way to find instances with attribute noise in movie reviews is to run BERT and sample from misclassified instances.

In our proposed methodology, if the annotator was uncertain, the instance was marked as *neutral*. If certain, but the assigned label (e.g. polarity) was incorrect, the instance was marked as *discrepant*. This process is simple and can be easily applied to other classification tasks to identify attribute noise in instances, no matter the labels available to the annotators. We made the dataset containing attribute noise labels publicly available so it can be used as a standard benchmark for robustness to noise in polarity classification tasks, and to potentially foster research on models, datasets and evaluation metrics tailored for this problem.

6.2 Future Work

There are many aspects of this work that can still be explored. Since our proposed methodology is simple and can be easily applied to other classification tasks, an easy approach is to apply it in other contexts. With the growth of the internet, the volume of online reviews available is huge. These reviews are found in several contexts such as books, series, musics, video games and others. A context that interested us a lot and can bring great results are reviews from papers of conferences or journals. These reviewers are known for being inconsistent with their rating, so it would be more likely to find *discrepant* reviews.

Since reviews from *experts* are actually a short summary of the original review, another idea for future work is to summarize the large reviews from *regular users* before the classification task and analyze its impact. In this work, we showed that instances with attribute noise are much harder to be classified. We analyzed its impact in three deep learning models: BERT, C-LSTM and CNN-GRU. As future work, we can explore other models such as SVM and Decision trees. We trained the models using a random balanced dataset that probably contains noise. Another experiment that could be done is to train the models with and without noise and analyze how they behave.

A very important point of our methodology is to choose the annotators. They need to be familiar with the language of the reviews, for example. In our experiments, we used three annotators who are considered fluent in the English language. However, they are not native speakers. For this reason, it is possible that there are differences if the human classifier is formed by other people. Therefore, a next work could be done applying the methodology with native speakers as annotators and comparing with this work.

In addition to the annotators, the main labels (neutrality and discrepancy) are also very important in our methodology. Petrović et al. [2010] focused on event detection on tweets, which may be plagued with spam. Although this is not directly related to our own contribution, it poses an interesting direction for a future extension focused on analyzing the impact of malicious attribute noise, i.e., noise deliberately generated to mislead readers into visiting the spammer's website, for instance. Following the same idea, content generated by robots is also considered noise. In this way, methods used to identify robots can also identify noise. Thus, we can test these methods as baselines.

In this work, we discussed how machine learning methods are known as black boxes and caused lack of confidence in users. Moreover, we run three machine learning classifiers in our labeled reviews to understand their behaviour. In other words, we started to open these black boxes and analyzed how they deal with noise data. Nevertheless, we did not go further to explain each one specifications and limitations. As future work, one idea is to analyze them further. For instance, how each Bert's layers conduct on noise data, similar to Tenney et al. [2019].

Another idea is to consider the results discussed in this work to create more robust classifiers such as proposed by Valdivia et al. [2019]. These classifiers would be able to deal with noise and be less harmed by them. Following the same idea, the chance of finding discrepant and neutral reviews when the classifier makes mistakes increases in order of magnitude. Therefore, it can be used to detect instances with attribute noise. Furthermore, to point out neutral instances, the neutral classifier could be used. Performing a sanitation on the dataset before training the classifier is important in several applications such as recommendation systems. These systems use reviews to recommend movies, for example. If these reviews are noisy, they will cause problems in the recommendation. Consequently, removing or giving less importance to these instances is very important.

Bibliography

Agarwal, S., Godbole, S., Punjani, D., and Roy, S. (2007). How much noise is too much: A study in automatic text classification. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 3–12.

Arora, M. and Kansal, V. (2019). Character level embedding with deep convolutional neural network for text normalization of unstructured data for twitter sentiment analysis. *Social Network Analysis and Mining*, 9.

Artstein, R. and Poesio, M. (2008). Survey article: Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555--596.

Arunachalam, R. and Sarkar, S. (2013). The new eye of government: Citizen sentiment analysis in social media. In *Proceedings of the IJCNLP 2013 workshop on natural language processing for social media (SocialNLP)*, pages 23--28.

Baldania, R. (2017). Sentiment analysis approaches for movie reviews forecasting: A survey. In *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pages 1–6.

Baldwin, T., Cook, P., Lui, M., MacKinlay, A., and Wang, L. (2013). How noisy social media text, how diffrnt social media sources? In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 356--364, Nagoya, Japan. Asian Federation of Natural Language Processing.

Barbosa, L. and Feng, J. (2010). Robust Sentiment Detection on Twitter from Biased and Noisy Data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 36--44, USA. Association for Computational Linguistics.

Beigman, E. and Klebanov, B. B. (2009). Learning with annotation noise. *ACL-IJCNLP 2009 - Joint Conf. of the 47th Annual Meeting of the Association for Com-*

putational Linguistics and 4th Int. Joint Conf. on Natural Language Processing of the AFNLP, Proceedings of the Conf., pages 280--287.

Beigman Klebanov, B. and Beigman, E. (2014). Difficult cases: From data to learning, and back. *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, 2:390--396.

Bermingham, A. and Smeaton, A. F. (2010). Classifying sentiment in microblogs: Is brevity an advantage? In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, page 1833–1836, New York, NY, USA. Association for Computing Machinery.

Bhavitha, B. K., Rodrigues, A. P., and Chiplunkar, N. N. (2017). Comparative study of machine learning techniques in sentimental analysis. In *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 216–221.

Boatwright, P., Basuroy, S., and Kamakura, W. (2007). Reviewing the reviewers: The impact of individual film critics on box office performance. *Quantitative marketing and economics*, 5(4):401--425.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 144–152, New York, NY, USA. Association for Computing Machinery.

Cheong, M. and Lee, V. C. S. (2011). A microblogging-based approach to terrorism informatics: Exploration and chronicling civilian sentiment and response to terrorism events via Twitter. *Information Systems Frontiers*, 13(1):45--59.

Ciresan, D., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642--3649. IEEE.

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. (2018). What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126--2136, Melbourne, Australia. Association for Computational Linguistics.

Contractor, D., Faruquie, T. A., and Subramaniam, L. V. (2010). Unsupervised cleansing of noisy text. In *Coling 2010: Posters*, pages 189--196, Beijing, China. Coling 2010 Organizing Committee.

Deriu, J., Lucchi, A., De Luca, V., Severyn, A., Müller, S., Cieliebak, M., Hofmann, T., and Jaggi, M. (2017). Leveraging Large Amounts of Weakly Supervised Data for Multi-Language Sentiment Classification. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1045--1052, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). 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)*, pages 4171--4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dey, L. and Haque, S. (2008). Opinion mining from noisy text data. *International Journal on Document Analysis and Recognition (IJDAR)*, 12:205–226.

Dey, L. and Haque, S. K. M. (2009). Studying the effects of noisy text on text mining applications. In *Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data*, AND '09, page 107–114, New York, NY, USA. Association for Computing Machinery.

Dodge, S. and Karam, L. (2017). A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th international conference on computer communication and networks (ICCCN)*, pages 1--7. IEEE.

Došilović, F. K., Brčić, M., and Hlupić, N. (2018). Explainable artificial intelligence: A survey. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 0210–0215.

Eisenstein, J. (2013). What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359--369, Atlanta, Georgia. Association for Computational Linguistics.

Elaraby, M. and Abdul-Mageed, M. (2018). Deep models for Arabic dialect identification on benchmarked data. In *Proceedings of the Fifth Workshop on NLP for Similar*

Languages, Varieties and Dialects (VarDial 2018), pages 263--274, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Esuli, A. and Sebastiani, F. (2013). Improving text classification accuracy by training label cleaning. *ACM Trans. Inf. Syst.*, 31(4). ISSN 1046-8188.

Fang, X. and Zhan, J. (2015). Sentiment analysis using product review data. *Journal of Big Data*, 2(1):5. ISSN 2196-1115.

Fefilatyev, S., Shreve, M., Kramer, K., Hall, L., Goldgof, D., Kasturi, R., Daly, K., Remsen, A., and Bunke, H. (2012). Label-noise reduction with support vector machines. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3504–3508, Washington, DC, USA. IEEE.

Florian, R., Pitrelli, J., Roukos, S., and Zitouni, I. (2010). Improving mention detection robustness to noisy input. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 335--345, Cambridge, MA. Association for Computational Linguistics.

Frenay, B. and Verleysen, M. (2014). Classification in the Presence of Label Noise: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845--869. ISSN 2162-237X.

Garg, A. and Roth, D. (2001). Understanding probabilistic classifiers. In *European Conference on Machine Learning*, pages 179--191. Springer.

Geirhos, R., Janssen, D. H., Schütt, H. H., Rauber, J., Bethge, M., and Wichmann, F. A. (2017). Comparing deep neural networks against humans: object recognition when the signal gets weaker. *arXiv preprint arXiv:1706.06969*.

Geirhos, R., Temme, C. R. M., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. (2018). Generalisation in humans and deep neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 7538--7550. Curran Associates, Inc.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89.

Goldberg, A. B., Zhu, X., and Wright, S. (2007). Dissimilarity in graph-based semi-supervised classification. In *Artificial Intelligence and Statistics*, pages 155--162.

Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., and Shet, V. (2013). Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv preprint arXiv:1312.6082*.

Grace, K., Salvatier, J., Dafoe, A., Zhang, B., and Evans, O. (2017). When will AI exceed human performance? evidence from AI experts. *CoRR*, abs/1705.08807.

Graf, A. B. A. and Wichmann, F. A. (2003). Insights from Machine Learning Applied to Human Visual Classification. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, pages 905--12, Cambridge, MA, USA. MIT Press.

Gui, L., Zhou, Y., Xu, R., He, Y., and Lu, Q. (2017). Learning representations from heterogeneous network for sentiment classification of product reviews. *Knowledge-Based Systems*, 124:34--45. ISSN 09507051.

Gupta, S. and Gupta, A. (2019). Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Computer Science*, 161:466 – 474. ISSN 1877-0509. The Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia.

Han, H., Otto, C., Liu, X., and Jain, A. K. (2015). Demographic estimation from face images: Human vs. machine performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(6):1148–1161.

Hassan, A. and Mahmood, A. (2018). Convolutional recurrent deep learning model for sentence classification. *Ieee Access*, 6:13949--13957.

Hendrycks, D., Mazeika, M., Wilson, D., and Gimpel, K. (2018). Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in neural information processing systems*, pages 10456--10465.

Jamison, E. and Gurevych, I. (2015). Noise or additional information? Leveraging crowdsource annotation item agreement for natural language tasks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 291--297, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jindal, I., Pressel, D., Lester, B., and Nokleby, M. (2019). An effective label noise model for dnn text classification. In *Proceedings of the 2019 Conference of the North*

American Chapter of the Association for Computational Linguistics, pages 3246--3256, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jochim, C. and Schütze, H. (2014). Improving Citation Polarity Classification with Product Reviews. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 42--48, Stroudsburg, PA, USA. Association for Computational Linguistics.

Khoshgoftaar, T. M. and Van Hulse, J. (2009). Empirical case studies in attribute noise detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 39(4):379–388.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kotsiantis, S. B. (2013). Decision trees: a recent overview. *Artificial Intelligence Review*, 39(4):261--283.

Kusner, M. J., Sun, Y., Kolkin, N. I., and Weinberger, K. Q. (2015). From word embeddings to document distances. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 957–966. JMLR.org.

Lai, S., Xu, L., Liu, K., and Zhao, J. (2015). Recurrent convolutional neural networks for text classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2267--2273, California. AAAI Press.

Lee, G., Jeong, J., Seo, S., Kim, C., and Kang, P. (2018). Sentiment classification with word localization based on weakly supervised learning with a convolutional neural network. *Knowledge-Based Systems*, 152:70--82. ISSN 09507051.

Li, L., Goh, T.-T., and Jin, D. (2020). How textual quality of online reviews affect classification performance: A case of deep learning sentiment analysis. *Neural Computing and Applications*, 32(9):4387--4415.

Lipton, Z. C. and Steinhardt, J. (2019). Troubling trends in machine-learning scholarship. *Queue*, 17(1):1--15. ISSN 15427749.

Liu, G. and Guo, J. (2019). Bidirectional lstm with attention mechanism and convolutional layer for text classification. *Neurocomputing*, 337:325 – 338. ISSN 0925-2312.

Liu, T., Wang, K., Chang, B., and Sui, Z. (2017). A Soft-label Method for Noise-tolerant Distantly Supervised Relation Extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1790--1795, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lopresti, D. (2005). Performance evaluation for text processing of noisy inputs. In *Proceedings of the 2005 ACM Symposium on Applied Computing*, SAC '05, page 759–763, New York, NY, USA. Association for Computing Machinery.

Lourentzou, I., Manghnani, K., and Zhai, C. (2019). Adapting sequence to sequence models for text normalization in social media. *Proceedings of the International AAAI Conference on Web and Social Media*, 13(01):335–345.

Lu, Y., Rao, Y., Yang, J., and Yin, J. (2018). Incorporating lexicons into lstm for sentiment classification. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, Piscataway, NJ, USA. IEEE Press. ISSN 2161-4407.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 4765--4774. Curran Associates, Inc.

M., V., Vala, J., and Balani, P. (2016). A survey on sentiment analysis algorithms for opinion mining. *International Journal of Computer Applications*, 133:7–11.

Mannino, M., Yang, Y., and Ryu, Y. (2009). Classification algorithm sensitivity to training data with non representative attribute noise. *Decision Support Systems*, 46(3):743 – 751. ISSN 0167-9236. Wireless in the Healthcare.

Martins, K. S. and Vaz de Melo, P. O. S. (2019). Characterization of the discrepancies between scores and texts of movie reviews. In *Proceedings of the 25th Brazillian Symposium on Multimedia and the Web*, WebMedia '19, page 229–236, New York, NY, USA. Association for Computing Machinery.

Marvin, R. and Linzen, T. (2018). Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192--1202, Brussels, Belgium. Association for Computational Linguistics.

Medhat, W., Hassan, A., and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093 – 1113. ISSN 2090-4479.

Mesaros, A., Heittola, T., and Virtanen, T. (2017). Assessment of human and machine performance in acoustic scene classification: Dcase 2016 case study. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 319–323.

Michel, P. and Neubig, G. (2018). MTNT: A Testbed for Machine Translation of Noisy Text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543--553.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111--3119.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235--244.

Mitchell, T. M. et al. (1997). Machine learning.

Moraes, R., Valiati, J. F., and Neto], W. P. G. (2013). Document-level sentiment classification: An empirical comparison between svm and ann. *Expert Systems with Applications*, 40(2):621 – 633. ISSN 0957-4174.

Nagamma, P., Pruthvi, H. R., Nisha, K. K., and Shwetha, N. H. (2015). An improved sentiment analysis of online movie reviews based on clustering for box-office prediction. In *International Conference on Computing, Communication Automation*, pages 933–937, Piscataway, NJ, USA. IEEE Press. ISSN .

Natarajan, N., Dhillon, I. S., Ravikumar, P., and Tewari, A. (2013). Learning with noisy labels. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, page 1196–1204, Red Hook, NY, USA. Curran Associates Inc.

Nettleton, D. F., Orriols-Puig, A., and Fornells, A. (2010). A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review*, 33(4):275--306. ISSN 0269-2821.

Ouyang, X., Zhou, P., Li, C. H., and Liu, L. (2015). Sentiment analysis using convolutional neural network. In *2015 IEEE International Conference on Computer and*

Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, pages 2359–2364, Piscataway, NJ, USA. IEEE Press. ISSN .

Palkar, R. K., Gala, K. D., Shah, M. M., and Shah, J. N. (2016). Comparative evaluation of supervised learning algorithms for sentiment analysis of movie reviews. *International Journal of Computer Applications*, 142(1):20--26.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79--86. Association for Computational Linguistics.

Park, S., Bak, J., and Oh, A. (2017). Rotated word vector representations and their interpretability. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 401--411, Copenhagen, Denmark. Association for Computational Linguistics.

Pelevina, M., Arefiev, N., Biemann, C., and Panchenko, A. (2016). Making Sense of Word Embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 174--183, Stroudsburg, PA, USA. Association for Computational Linguistics.

Petrović, S., Osborne, M., and Lavrenko, V. (2010). Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181--189, Los Angeles, California. Association for Computational Linguistics.

Poliak, A., Haldar, A., Rudinger, R., Hu, J. E., Pavlick, E., White, A. S., and Van Durme, B. (2018). Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 67--81, Brussels, Belgium. Association for Computational Linguistics.

Pozzi, F. A., Fersini, E., Messina, E., and Liu, B. (2016). *Sentiment analysis in social networks*. Morgan Kaufmann.

Prati, R. C., Luengo, J., and Herrera, F. (2019). Emerging topics and challenges of learning from noisy data in nonstandard classification: a survey beyond binary class noise. *Knowledge and Information Systems*, 60(1):63--97.

Pujara, J., Augustine, E., and Getoor, L. (2017). Sparsity and Noise: Where Knowledge Graph Embeddings Fall Short. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1751--1756, Stroudsburg, PA, USA. Association for Computational Linguistics.

Raaijmakers, S., Sappelli, M., and Kraaij, W. (2017). Investigating the interpretability of hidden layers in deep text mining. In *Proceedings of the 13th International Conference on Semantic Systems*, Semantics2017, page 177–180, New York, NY, USA. Association for Computing Machinery.

Rehbein, I. and Ruppenhofer, J. (2017). Detecting annotation noise in automatically labelled data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1160--1170, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rezaeinia, S. M., Rahmani, R., Ghodsi, A., and Veisi, H. (2019). Sentiment analysis based on improved pre-trained word embeddings. *Expert Systems with Applications*, 117:139--147.

Samek, W., Wiegand, T., and Müller, K. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *CoRR*, abs/1708.08296.

Sharoff, S., Wu, Z., and Markert, K. (2010). The web library of babel: evaluating genre collections. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Shelke, N. M., Deshpande, S., and Thakre, V. (2012). Survey of techniques for opinion mining. *International Journal of Computer Applications*, 57:30–35.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323--332. ISSN 08936080.

Subramaniam, L. V., Roy, S., Faruquie, T. A., and Negi, S. (2009). A survey of types of text noise and techniques to handle noisy text. In *Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data*, AND '09, page 115–122, New York, NY, USA. Association for Computing Machinery.

Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., and Fergus, R. (2015). Training convolutional networks with noisy labels. *ICLR*. 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015.

Sáez, J. A., Luengo, J., and Herrera, F. (2016). Evaluating the classifier behavior with noisy data considering performance and robustness: The equalized loss of accuracy measure. *Neurocomputing*, 176:26 – 35. ISSN 0925-2312. Recent Advancements in Hybrid Artificial Intelligence Systems and its Application to Real-World Problems.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701--1708.

Tang, D., Qin, B., and Liu, T. (2015). Deep learning for sentiment analysis: successful approaches and future challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(6):292--303.

Tang, H., Tan, S., and Cheng, X. (2009). A survey on sentiment detection of reviews. *Expert Systems with Applications*, 36(7):10760 – 10773. ISSN 0957-4174.

Teng, C.-M. (1999). Correcting Noisy Data. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, pages 239--248, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Tenney, I., Das, D., and Pavlick, E. (2019). Bert rediscovers the classical nlp pipeline. In *Association for Computational Linguistics.*

Toledo, R. Y., Mota, Y. C., and Martínez, L. (2015). Correcting noisy ratings in collaborative recommender systems. *Knowledge-Based Systems*, 76:96--108. ISSN 09507051.

Topal, K. and Ozsoyoglu, G. (2016). Movie review analysis: Emotion analysis of imdb movie reviews. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1170–1176, Piscataway, NJ, USA. IEEE Press. ISSN .

Tsapatsoulis, N. and Djouvas, C. (2017). Feature extraction for tweet classification: Do the humans perform better? In *2017 12th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, pages 53–58.

Tsapatsoulis, N. and Djouvas, C. (2019). Opinion mining from social media short texts: Does collective intelligence beat deep learning? *Frontiers in Robotics and AI*, 5:138. ISSN 2296-9144.

Vaibhav, V., Singh, S., Stewart, C., and Neubig, G. (2019). Improving Robustness of Machine Translation with Synthetic Noise. In *Proceedings of the 2019 Conference of the North*, pages 1916--1920, Stroudsburg, PA, USA. Association for Computational Linguistics.

Valdivia, A., Hrabova, E., Chaturvedi, I., Luzón, M. V., Troiano, L., Cambria, E., and Herrera, F. (2019). Inconsistencies on TripAdvisor reviews: A unified index between users and Sentiment Analysis Methods. *Neurocomputing*, 353:3--16. ISSN 09252312.

Van Hulse, J. D., Khoshgoftaar, T. M., and Huang, H. (2007). The pairwise attribute noise detection algorithm. *Knowledge and Information Systems*, 11(2):171--190. ISSN 0219-1377.

Vinciarelli, A. (2005). Noisy text categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1882–1895.

Vinodhini, G. and Chandrasekaran, R. (2012). Sentiment analysis and opinion mining: a survey. *International Journal*, 2(6):282--292.

Wang, X., Jiang, W., and Luo, Z. (2016). Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2428--2437, Osaka, Japan. The COLING 2016 Organizing Committee.

Wankhede, R. and Thakare, A. N. (2017). Design approach for accuracy in movies reviews using sentiment analysis. In *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, volume 1, pages 6–11, Piscataway, NJ, USA. IEEE Press. ISSN .

Wichmann, F. A., Graf, A. B. A., Simoncelli, E. P., Bülthoff, H. H., and Schölkopf, B. (2004). Machine Learning Applied to Perception: Decision-Images for Gender Classification. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, pages 1489--1496, Cambridge, MA, USA. MIT Press.

Xiaojing Shi and Xun Liang (2015). Resolving inconsistent ratings and reviews on commercial webs based on support vector machines. In *2015 12th International*

Conference on Service Systems and Service Management (ICSSSM), pages 1–6, Piscataway, NJ, USA. IEEE Press. ISSN 2161-1890.

Yasen, M. and Tedmori, S. (2019). Movies reviews sentiment analysis and classification. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pages 860–865.

Younes, Z., abdallah, F., and Denœux, T. (2010). Evidential multi-label classification approach to learning from data with imprecise labels. In Hüllermeier, E., Kruse, R., and Hoffmann, F., editors, *Computational Intelligence for Knowledge-Based Systems Design*, pages 119--128, Berlin, Heidelberg. Springer Berlin Heidelberg.

Zhang, L., Wang, S., and Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253.

Zhang, W., Yoshida, T., and Tang, X. (2011). A comparative study of tf* idf, lsi and multi-words for text classification. *Expert Systems with Applications*, 38(3):2758--2765.

Zhang, Y. (2015). Incorporating Phrase-level Sentiment Analysis on Textual Reviews for Personalized Recommendation. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining - WSDM '15*, pages 435--440, New York, New York, USA. ACM Press.

Zhou, C., Sun, C., Liu, Z., and Lau, F. C. M. (2015). A C-LSTM neural network for text classification. *CoRR*, abs/1511.08630.

Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., and Xu, B. (2016). Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3485--3495.

Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 912–919, ACM, NY. AAAI Press.

Zhu, X. and Wu, X. (2004). Class Noise vs. Attribute Noise: A Quantitative Study. *Artificial Intelligence Review*, 22(3):177--210. ISSN 0269-2821.