Why Do Document-Level Polarity Classifiers Fail?

Karen S. Martins
CS Department
UFMG
Belo Horizonte, MG, Brazil
karensm@dcc.ufmg.br

Pedro O.S. Vaz-de-Melo
CS Department
UFMG
Belo Horizonte, MG, Brazil
olmo@dcc.ufmg.br

Rodrygo L. T. Santos
CS Department
UFMG
Belo Horizonte, MG, Brazil
rodrygo@dcc.ufmg.br

Abstract

Machine learning solutions are often criticized for the lack of explanation of their successes and failures. Understanding which instances are misclassified and why is essential to improve the learning process. This work helps to fill this gap by proposing a methodology to characterize, quantify and measure the impact of *hard instances* in the task of polarity classification of movie reviews. We characterize such instances into two categories: *neutrality*, where the text does not convey a clear polarity, and *discrepancy*, where the polarity of the text is the opposite of its true rating. We quantify the number of *hard instances* 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. To the best of our knowledge, this is the first systematic analysis of the impact of *hard instances* in polarity detection from well-formed textual reviews.

1 Introduction

Document-level polarity classification is the task of classifying the polarity of a whole opinionated message (Pozzi et al., 2016). For instance, given a movie review, the system determines whether the review text expresses an overall positive, negative, or neutral opinion about the movie. Although polarity classification naturally suits to analyze consumer opinions about products and services (Gui et al., 2017), 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).

Supervised machine learning is one of the most common and successful approaches for polarity classification, but even state-of-the-art methods fail to correctly classify a substantial portion of the instances, from 10% to 20%, depending on the dataset (Ribeiro et al., 2016). The problem with this approach is that if the data is not representative and reliable, the model is unlikely to perform well. One source of unreliability is data noise, which can be categorized into *class noise* and *attribute noise* (Gupta and Gupta, 2019). Class noise occurs when the training data contains instances that are wrongly labeled. Attribute noise occurs when the training data contains one or more attributes with wrong, incomplete or missing values. In the case of textual data, such noise usually comes in the form of errors in language rules, such as typos, grammatical errors, improper punctuation, and abbreviations (Agarwal et al., 2007; Michel and Neubig, 2018; Lourentzou et al., 2019). Nevertheless, for both cases, the noise can be eliminated from the data by correcting the labels (for class noise) or the problematic text (for attribute noise).

A more problematic source of data unreliability in polarity classification tasks comes from well written text that, for some reason, does not convey its class clearly. Literature calls such instances *hard instances*, which are those that are intrinsically hard to correctly label or classify (Smith and Martinez, 2011; Beigman Klebanov and Beigman, 2014). Differently from noisy instances, *hard instances* cannot be corrected, so the only solution is to identify and remove them from the training data. Also, *hard instances* are not equivalent to outliers, as they do not differ significantly from other observations and may represent a significant portion of the data (Smith et al., 2014). For example, in a polarity classification task, a positive movie review that describes at least as many negative as positive points of the film can be a *hard instance*. To the best of our knowledge, no study exists that characterizes such instances and quantifies their impact on document-level polarity classification tasks.

Thus, we propose a methodology to characterize,

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1782–1794
June 6–11, 2021.

quantify and measure the impact of *hard instances* in polarity classification tasks and demonstrate its usefulness in the task of movie review polarity classification. To this end, we collected $415,867$ *positive* and *negative* movie reviews from Metacritic. One advantage of Metacritic is that the meaning of ratings is clearly stated to the users when a review is being submitted: positive ratings range between 61% and 100%, neutral range between 40% and 60%, and negative between 0% and 39%. Because of that, class noise and biases should be rare, that is, a user who liked (disliked) a movie will very unlikely give a negative (positive) rating to it. Thus, classification errors will mostly be due to *hard instances*, which we assign into two disjoint categories: *neutral* and *discrepant*. A *neutral* review does not have a clear polarity and a *discrepant* review has a human-perceived polarity that is different from its associated rating. This categorization is complete, i.e., every instance that, for a human, does not reveal its class clearly falls into one (and only one) of these two types of *hard instances*.

Neutral and *discrepant* reviews are characterized by a well-defined human classifier that uses human reasoning to infer the class of the example. 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 rating. When the human classifier is not confident about its prediction, we label the review as *neutral*. We labeled $1,200$ reviews and found 198 *neutral* and 64 *discrepant* reviews. We tested state-of-the-art machine classifiers on these reviews and results revealed that *hard instances* can significantly decrease their performances. In short, the main contributions are:

- A simple and reproducible methodology based on a well-defined human classifier to characterize and identify *hard instances* on polarity classification tasks (Section 3);
- A thorough analysis of the impact of *hard instances* in the task of movie review polarity classification (Section 5.2);
- Publicly available datasets of movie reviews describing the expected amounts of five classes of *hard instances* (Section 5.1).

As an additional contribution, we show how far are state-of-the-art machine classifiers from human performance in the task of movie review polarity classification.

2 Related Work

In supervised machine learning, class and attribute noise can increase learning complexity and, consequently, reduce classification accuracy (Zhu and Wu, 2004). 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). Thus, class noise is more often addressed in the literature (Gupta and Gupta, 2019), where several studies analyzed its impact in classification tasks and how to address it (Natarajan et al., 2013; Hendrycks et al., 2018; Liu et al., 2017; Rehbein and Ruppenhofer, 2017; Jindal et al., 2019). In NLP, attribute noise are unintended errors in text, which can come from failures in automatic character recognition processes (Vinciarelli, 2005) or naturally while writing the text in the form of errors in language rules, such as typos, grammatical errors, improper punctuation, irrational capitalization and abbreviations (Agarwal et al., 2007; Contractor et al., 2010; Dey and Haque, 2009; Florian et al., 2010; Michel and Neubig, 2018). In short, noise are unintentional and undesirable errors in the text that can (and should) be eliminated from the data.

Conversely, *hard instances* are noise-free and cannot be corrected, only eliminated from the data (Smith and Martinez, 2011). In addition, they differ from outliers because their feature representation vectors may be similar to others from regular instances (Smith and Martinez, 2011). Nevertheless, *hard instances* are more prone to class noise. In fact, Beigman Klebanov and Beigman (2009) defined *hard instances* in the context of label annotations, under the assumption that items that are easy are reliably annotated, whereas items that are hard display confusion and disagreement among the annotators. Later, Beigman Klebanov and Beigman (2014) showed that the presence of *hard instances* in the training data misleads the machine learner on easy, clear-cut cases. The definition of Smith et al. (2014) is similar to ours: *hard instances* are simply those that "should be misclassified" by machine learning methods. The authors introduced *hardness measures* based on the outputs of an ensemble of classifiers to identify such instances and showed that classifiers are often uncertain about their classes. Following the same idea, Krymolowski (2002) argues that easy instances are correctly classified by all or most classifiers. On the other hand, *hard instances* are missed by most of them.

In this work, we propose a *human classifier* composed by human annotators to identify *hard instances*. Our definition unifies the ones of Beigman Klebanov and Beigman (2009, 2014) and Smith et al. (2014). Similarly to Beigman Klebanov and Beigman (2009, 2014), we define *hard instances* as those in which the *human classifier* is uncertain or wrong about their true labels. However, different from these studies, which quantify the impact *hard instances* have on training, our goal is to provide a methodology to quantify the expected amount of *hard instances* in data and the impact they have on classifiers in production and testing. Also, and similarly to Smith et al. (2014), *hard instances* are divided into "instances that should be misclassified", which we call *discrepant*, and "border points", which we call *neutral*. To the best of our knowledge, we are the first to propose a methodology to characterize and quantify the impact of *hard instances* in unstructured textual data for polarity classification tasks.

Regarding the effect of *hard instances* in sentiment and polarity classification tasks, 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. Also, Valdivia et al. (2019) showed that ratings in TripAdvisor reviews are not strongly correlated with sentiment scores given by sentiment analysis methods and proposed a unified index that aggregates both polarities. Barnes et al. (2019) collected a subset of sentences that an ensemble of state-of-the-art sentiment classifiers misclassified and annotated them for 18 linguistic and paralinguistic phenomena, such as negation, sarcasm, among others. In our work, we analyze manually identified *hard instances* (as opposed to instances misclassified by a machine classifier). As a result, compared to these works, we have a more precise (e.g., a misclassified instance is not necessarily hard) and complete (e.g., not all *hard instances* are misclassified) ground-truth.

3 Methodology

Problem Setting. In this work, we focus on the problem of polarity detection of movie reviews, but all the methods can be applied to any document-level polarity classification task. More formally, in a dataset $\mathcal{D} = (X, Y)$ composed by a set of textual movie reviews X and their corresponding binary ratings Y, each review $x_i \in X$ is associated with a score (or rating) $y_i \in Y$ that can be either 0 (*positive*) or 1 (negative). For the aims of this paper, 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 of 50 on *Metacritic*. By doing this, we isolate *hard instances* from explicit neutral reviews, avoiding class noise and biases.

Our methodology is composed by a human classifier f_H, which identifies *hard instances*, and a machine classifier f_M, which is tested on *hard* and *regular* instances. A classifier is defined as a function $f(x_i)$ that receives a textual movie review x_i as input and returns its polarity $\hat{y}_i \in \{0, 1\}$. We use the human classifier to assign a label l_i to a large sample of movie reviews x_i to indicate whether x_i is a *hard instance* or not. This label can be one (and only one) of a set L of manually defined labels that indicate that the instance is *regular* or a type of *hard instance*. With that, we will be able to quantify the impact of *hard instances* 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 $l \in L$, *regular* included, we will calculate the probabilities $P(l_i = l | y_i \neq \hat{y}_i)$ and $P(y_i = \hat{y}_i | l_i = l)$.

Types of Hard Instances. 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, which are always visible to the reviewers. Thus, every time the polarity of text x_i is inconsistent with its true score y_i, we assume that x_i is a *hard instance*. More specifically, we define two possible hypotheses explaining the hardness of the text x_i, i.e., two disjoint types of *hard instances*: (1) the text does not have a clear polarity, namely *neutrality*, and (2) the text has a clear polarity, but its score y_i is the opposite one, namely *discrepancy*.

A movie review x_i is a *hard instance* of type *neutrality* when its polarity is not clear. We define three labels for *neutral hard instances*: *mixed* (text has mixed opinions), *factual* (text is purely factual) and *contextual* (polarity needs context). The *mixed* label considers reviews that describes both positive and negative points about the movie without having the overall opinion clearly stated. One real exam-

ple is: *"as dumb as the film is, the actors escape relatively unscathed."* The *factual* label defines non-opinionated reviews that describes only facts about the movie, such as: *"it is a movie about the World War II and its consequences on the lives of those who survived."* The label *contextual* characterizes reviews where context is needed to understand its polarity, including those containing irony and sarcasm. One real example is: *"ultimately, Collin's film is one of forgiveness and that's not the usual way great tragedies end."* Finally, the label *hard_undefined* is given to reviews where the reasons for the lack of polarity are not clear.

The second type of *hard instance*, namely *discrepancy*, is given to reviews where the polarity of its text x_i is the opposite of the polarity of its score y_i. For this type, we define a single label: *discrepant* (polarity of text and score are discrepant). As an example, 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. The following textual review has a clear negative polarity although its score y_i is positive: *"Thoroughly predictable from start to finish."* For more examples, see Tables 4 and 5 in the Appendix.

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; Ciresan et al., 2012; Geirhos et al., 2018), which 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 a *hard instance* (Beigman Klebanov and Beigman, 2014). We use this premise to define the two types of *hard instances* discussed in the previous section.

In the task of polarity classification of movie reviews, a human classifier mistake 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 a *hard instance* of type *neutrality*, whereas the second one **(C2)** is associated with a *hard instance* 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 $\hat{y}_i$.

With the aforementioned rationale, we are ready to propose a well-defined human classifier f_H to identify *hard instances* 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 hardness[1]. 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*. 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 is a *hard instance* of type *neutrality*. So, annotator j is asked to choose one label l_i^j that fits best to the *neutrality* of x_i, which can be either *mixed*, *factual* or *contextual*. On the other hand, if $c_i^j = 1$, then l_i^j is set to *regular*. This process is illustrated in Figure 1. Of course, each annotator j is independent and cannot see the others' responses.

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 $l_i^j \in \mathcal{L}_N = \{$ *mixed*, *factual*, *contextual*, *regular* $\}$. 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 1 and 2:

[1] In practice, any number of annotators can be used, including just one.

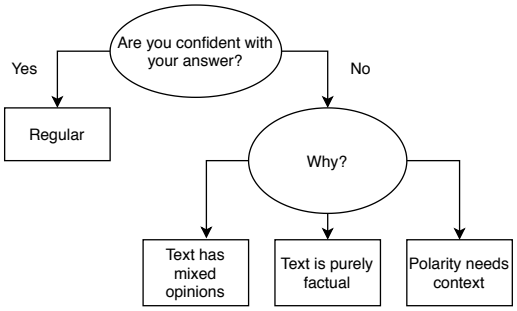

Figure 1: Confidence diagram.

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

$$c_i = \begin{cases} 0 & \textbf{if } \sum_{j=1}^{3} c_i^j \leq 1 \\ 1, & \text{otherwise.} \end{cases} \quad (2)$$

Setting the final *hard instance* 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. $\mathcal{L}_1 = [$*mixed*, *mixed*, *regular* $]$) and $N(l, \mathcal{L}_i)$ the number of elements of $\mathcal{L}_i$ that are equal to label l (e.g. $N(\textit{mixed}, \mathcal{L}_1) = 2$). Then, l_i is the majority vote if at least two annotators (the majority) gave that label to x_i and, if not, l_i is set to *hard_undefined*, indicating no consensus. This process is formally described by Equation 3:

$$l_i = \begin{cases} \arg\max_{l \in \mathcal{L}_N} N(l, \mathcal{L}_i) & \textbf{if } N(l, \mathcal{L}_i) \geq 2 \\ \textit{hard\_undefined}, & \text{otherwise.} \end{cases} \quad (3)$$

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*. It is easy to see that this update step will be executed only if l_i was previously set to *regular*, i.e., it will not overwrite a *neutrality* label. Equation 4 defines the *discrepancy* update step:

$$l_i = \textit{discrepant} \ \textbf{if } \hat{y}_i \neq y_i \ \textbf{and } c_i = 1. \quad (4)$$

4 Experimental Setup

Data Set. We collected movie reviews from Metacritic,[2] which 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 Section 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*. Differently, *regular users* reviews 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*. As previously mentioned, the meaning of each rating is clearly conveyed to users in the Metacritic website. Thus, class noise and biases should be rare in the dataset.

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. 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. 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. Hence, we selected a clean and unambiguous dataset. Reviews from *experts* are usually shorter than 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*. In addition, we observed that *experts* use a more elaborate language. Because of these differences, we will condition our analyses on the type of user (*experts* or *regular users*) and score polarity (*positive* or *negative*).

Machine Classifiers. To evaluate the impact of *hard instances* on machine classifiers, we selected three state-of-the-art models 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). C-LSTM utilizes a Convolutional Neural Network (CNN) to extract a sequence of higher-level phrase representations, which are then fed into a Long Short-Term Memory (LSTM) unit to obtain the sentence representation. CNN-GRU connects a character-aware CNN with a character-aware Gated Recurrent Unit (GRU) to learn long sequence semantics. These two networks are initialized with pre-trained Word2vec vectors from Google News

[2] https://www.metacritic.com/movie

Dataset and have their final representations connected to a dense layer. 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 a dense layer. For all architectures, the output $\hat{y}_i$ is given by a sigmoid function. For implementation and code details, please see the Appendix.

5 Results

5.1 Number of Hard Instances

The first question we need to answer is: how many *hard instances* exist 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 number of *hard instances* in D_H, we use our proposed human classifier f_H described in Section 3 to label every review $x_i \in D_H$. Recall that f_H assigns a polarity $\hat{y}_i \in \{$*positive*, *negative* $\}$ to x_i and, more important to our purpose here, a label l_i, which can be either *regular* (instance is not a *hard instance*), *discrepant* (the polarity of the text is different from the score polarity), or one of the four *neutrality* labels: *mixed* (text has mixed opinions), *factual* (text is purely factual), *contextual* (polarity needs context) and *hard_undefined* (reasons are unclear). Also, let $u_i \in \{$*expert*, *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.

In Table 1, we show the number and proportion of movie reviews that are or are not *hard instances* for *experts*. From the 400 labeled reviews, almost one quarter (92) are *hard instances*. 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* instances are 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 *contextual* (6.8%), followed by *mixed* (4.3%) and *factual* (4.3%). Also, *contextual* instances are more common in *negative* reviews, when *experts* often use figures of speech (e.g. irony) together with external knowledge to create humour. Finally, *factual* instances are more present in *positive* reviews, where the *experts* simply describe some characteristic of the movie that impressed them without explicitly saying that.

label (l_i)	***positive***	***negative***	**total**
experts			
regular	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			
regular	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 1: Number of *hard instances* in reviews.

Also, in Table 1 we show the number and proportion of movie reviews that are or are not *hard instances* for *regular users*. First, note that the number of reviews that are *hard instances* significantly decreased in comparison with the ones written by *experts*. From the 400 labeled reviews, only 36(9%) are *hard instances*, of which 31 are *neutral* and only 5 are *discrepant*. Different from what was verified for *experts*, the most predominant label for *regular users* was *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 *hard instances* are 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 *mixed*.

A note about the human classifier. Because we used three human annotators in f_H and a majority vote function, only two annotators were used initially. The third annotator was called to classify 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 label l_i. 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 labels. Regarding the third annotator, only 1.5% of the instances were not in total agreement with at least one of the other annotators. The Cohen's kappa coefficient for the first two annotators was 0.82 in relation to polarity scores, 0.58 regarding their confidence levels and 0.49 regarding their attribute noise labels.

5.2 Impact of Hard Instances

In this section, we quantify the impact of *hard instances* 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. 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 Section 4 in the labeled dataset D_H (see Section 5.1), which contains 800 reviews. Because this 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 dataset contains $8,796$ reviews, $4,398$ of each polarity. Again, this dataset is solely 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 2. Finally, recall that y_i refers to the author's original polarity score (gold polarity) and $\hat{y}_i$ refers to the polarity predicted by the classifiers, including the human classifier.

Figure 2 shows the classification error (with their respective error bars) for all classifiers in 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) and more *hard instances* (see Section 5.1). 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 *regular* instances, *positive* instances have almost twice more *hard instances*, particularly *discrepant* ones. For *regular user* reviews, *positive* instances also have more *hard instances*, but the difference in terms of *neutral* reviews is more significant. Note that, for both user types, this difference in the instances misclassified by the human classifier is striking.

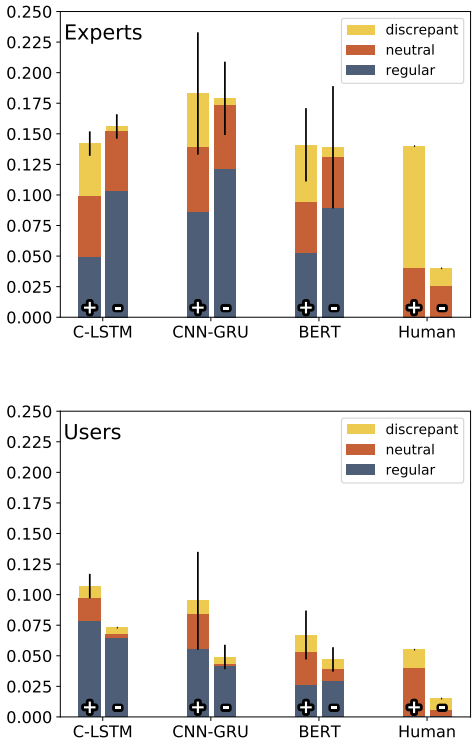

Figure 2: Classification error for all classifiers.

For a more precise assessment of the impact of *hard instances*, we show in Table 2 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 *neutral* instances and get even worse in *discrepant* instances. 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 *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 *hard_undefined*, which is also the label with the least instances, 12 out of 800.

	C-LSTM	CNN-GRU	BERT	Human
regular	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 2: 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 2, 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 $P(l_i = l \mid \hat{y}_i = y_i)$.

label (l_i)	$\hat{\mathbf{y}}_\mathbf{i} = \mathbf{y}_\mathbf{i}$	$\hat{\mathbf{y}}_\mathbf{i} \neq \mathbf{y}_\mathbf{i}$
experts		
regular	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		
regular	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 3: Percentage of labels in correct ($\hat{y}_i = y_i$) and incorrect ($\hat{y}_i \neq y_i$) predictions by BERT.

Table 3 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 *hard instances* in movie reviews is to run BERT and sample from misclassified instances. Our estimates for the probabilities of finding a misclassified *hard instance* is 0.64 for *experts* and 0.56 for *regular users*. In other words, more than 50% of our sampled misclassified instances are *hard instances*. Recall from Table 1 that we found only 23% of *hard instances* 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 *contextual* instances in misclassified *expert* reviews and *mixed* instances in misclassified *regular users* reviews. Therefore, to find *hard instances* with high probability, we propose to train and run BERT in the data (without filtering anything) and, from the misclassified instances, run the human classifier to identify them.

We investigated misclassified *regular* 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. For examples, see Table 6 in the Appendix. 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 *hard instances* is not that simple, where more guided and focused approaches are probably needed, such as the one proposed by Valdivia et al. (2019). They proposed an approach to combine reviews with scores for an aggregated polarity, which can be a good idea to deal with *hard instances*.

Overview of our results. Our first goal was to quantify the expected amount of *hard instances* in misclassifications, which is $\approx 56\%$ for *regular users* and $\approx 64\%$ for *experts*. Note that even though the reviews for these users are intrinsically different, the values are similar. The second goal was to quantitatively show how different the two types of *hard instances* are. Table 1 shows that *neutral* instances are common, and Table 3 shows they might have a significant presence even in correctly classified instances. Contrastingly, *discrepant* instances are rare, particularly among correctly classified instances. Given that our ultimate goal was to quantify and explain the reasons behind misclassifications, from Table 3 we can say that most of the mistakes ($\approx 60\%$) occur because of *neutral* ($\approx 38\%$) and *discrepant* ($\approx 22\%$) instances.

6 Conclusion

In this work, we propose a methodology to characterize, quantify and measure the impact of *hard instances* in the task of polarity classification of movie reviews. We characterized such instances into two disjoint categories: *neutrality* and *discrepancy*. We provided empirical evidence about the need to pay attention to such instances, as they are much harder to be classified, for both machine and human classifiers.

The main hypothesis of this work is that *hard instances* can make polarity classifiers fail. To demonstrate this hypothesis, we provided two well defined types of *hard instances*, which are based on human reasoning, and a methodology to find them in labeled data. With that, one can quantify how many instances of those types there are in their data, which can shed light on why and when classifiers fail. We collected a noise-free (no *class noise*) and well separated (no *neutral* polarity) dataset and showed that even in such a dataset most of the mistakes made by a state of the art classifier, namely BERT, are in our defined *hard instances*. Observe in Table 3 that more than 50% of our sampled misclassified instances are *hard instances* (*discrepant* or *neutral*).

Our methodology works for every type of supervised classification task. Because our proposed labels are defined from the perspective of a classifier fully capable of human-reasoning, they are easy to interpret and can be generalized to every classification task (e.g. polarity, image, song genre, topic) that humans are able to do. After employing our methodology, it will be possible to differentiate mistakes that come from *hard instances*, which are those even humans cannot classify with confidence (or at all), and mistakes that could be solved by improving the classifier architecture. In short, our proposed methodology can help quantify and explain why classifiers are making mistakes.

We made the dataset containing the labels publicly available[3] so it can be used as a standard benchmark for robustness to *hard instances* in polarity classification tasks, and to potentially foster research on models, datasets and evaluation metrics tailored for this problem.

Acknowledgements

This work is supported by the authors' individual grants from FAPEMIG, CAPES and CNPq. We also thank all the reviewers for their thoughtful comments which helped to improve this work.

References

S. Agarwal, S. Godbole, D. Punjani, and S. Roy. 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.

Ravi Arunachalam and Sandipan Sarkar. 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.

Jeremy Barnes, Lilja Øvrelid, and Erik Velldal. 2019. Sentiment Analysis Is Not Solved! Assessing and

[3] https://github.com/karenstemartins/NAACL2021

Probing Sentiment Classification. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 12–23, Stroudsburg, PA, USA. Association for Computational Linguistics.

Beata Beigman Klebanov and Eyal Beigman. 2009. From Annotator Agreement to Noise Models. *Computational Linguistics*, 35(4):495–503.

Beata Beigman Klebanov and Eyal Beigman. 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.

Adam Bermingham and Alan F. Smeaton. 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.

Marc Cheong and Vincent C S Lee. 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.

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

Danish Contractor, Tanveer A. Faruquie, and L. Venkata Subramaniam. 2010. Unsupervised cleansing of noisy text. In *Coling 2010: Posters*, pages 189–196. Coling 2010 Organizing Committee.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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. Association for Computational Linguistics.

Lipika Dey and S. K. Mirajul Haque. 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.

Mohamed Elaraby and Muhammad Abdul-Mageed. 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. Association for Computational Linguistics.

Radu Florian, John Pitrelli, Salim Roukos, and Imed Zitouni. 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.

Benoit Frenay and Michel Verleysen. 2014. Classification in the Presence of Label Noise: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869.

Robert Geirhos, Carlos R. M. Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. 2018. Generalisation in humans and deep neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7538–7550. Curran Associates, Inc.

Andrew B Goldberg, Xiaojin Zhu, and Stephen Wright. 2007. Dissimilarity in graph-based semi-supervised classification. In *Artificial Intelligence and Statistics*, pages 155–162.

Lin Gui, Yu Zhou, Ruifeng Xu, Yulan He, and Qin Lu. 2017. Learning representations from heterogeneous network for sentiment classification of product reviews. *Knowledge-Based Systems*, 124:34–45.

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

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. 2018. Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in neural information processing systems*, pages 10456–10465.

Ishan Jindal, Daniel Pressel, Brian Lester, and Matthew Nokleby. 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. Association for Computational Linguistics.

Yuval Krymolowski. 2002. Distinguishing easy and hard instances. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Tianyu Liu, Kexiang Wang, Baobao Chang, and Zhifang Sui. 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. Association for Computational Linguistics.

Ismini Lourentzou, Kabir Manghnani, and ChengXiang Zhai. 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.

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

Paul Michel and Graham Neubig. 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.

Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. 2013. Learning with noisy labels. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, page 1196–1204. Curran Associates Inc.

Federico Alberto Pozzi, Elisabetta Fersini, Enza Messina, and Bing Liu. 2016. *Sentiment analysis in social networks*. Morgan Kaufmann.

Ines Rehbein and Josef Ruppenhofer. 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. Association for Computational Linguistics.

Filipe N Ribeiro, Matheus Araújo, Pollyanna Gonçalves, Marcos André Gonçalves, and Fabrício Benevenuto. 2016. SentiBench - a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1):23.

Michael R. Smith and Tony Martinez. 2011. Improving classification accuracy by identifying and removing instances that should be misclassified. In *The 2011 International Joint Conference on Neural Networks*, pages 2690–2697. IEEE.

Michael R. Smith, Tony Martinez, and Christophe Giraud-Carrier. 2014. An instance level analysis of data complexity. *Machine Learning*, 95(2):225–256.

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

Ana Valdivia, Emiliya Hrabova, Iti Chaturvedi, M. Victoria Luzón, Luigi Troiano, Erik Cambria, and Francisco Herrera. 2019. Inconsistencies on TripAdvisor reviews: A unified index between users and Sentiment Analysis Methods. *Neurocomputing*, 353:3–16.

Jason D. Van Hulse, Taghi M. Khoshgoftaar, and Haiying Huang. 2007. The pairwise attribute noise detection algorithm. *Knowledge and Information Systems*, 11(2):171–190.

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

Xingyou Wang, Weijie Jiang, and Zhiyong Luo. 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. The COLING 2016 Organizing Committee.

Yongfeng Zhang. 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. ACM Press.

Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis C. M. Lau. 2015. A C-LSTM neural network for text classification. *CoRR*, abs/1511.08630.

Xingquan Zhu and Xindong Wu. 2004. Class Noise vs. Attribute Noise: A Quantitative Study. *Artificial Intelligence Review*, 22(3):177–210.

A Appendix

A.1 Examples of Reviews

Tables 4 and 5 show some examples of *hard instances* labeled by our human classifier. All the code and data is publicly available at `https://github.com/karenstemartins/NAACL2021`.

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 4: Examples of *hard instances* from *experts*.

A.2 Regular Reviews

Table 6 shows real examples of misclassified *regular* reviews with their original polarities given by their authors. The first and last review contain

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: Examples of *hard instances* from *regular users*.

some "irony" that is clear to humans, but they are created using words with their opposite polarity of the final score. The second review contains positive and negative points of the movie, but humans can easily identify what side has the most weight.

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)
"From the first moment of Superman till the very end scene Lex luthor this is a true comic book movie adaption. True there are few CGI errors, but "Nothing is perfect in this world" and this is just a movie." (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 6: Examples of misclassified *regular* reviews.

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 3 shows the result for the last review in Table 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 to the classifier misclassify the review.

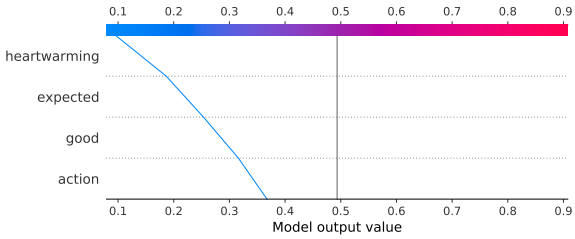

Figure 3: SHAP plot for the last review in Table 6.

A.3 Machine Classifier

A.3.1 Third Part Material

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

A.3.2 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 (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 the paper 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.