Quality of Sentiment Analysis Tools: The Reasons of Inconsistency

Wissam Mammar Kouadri
LIPADE,Université de Paris&
IMBA Consulting
wissam.maamar_kouadri@etu.parisdescartes.fr

Mourad Ouziri
LIPADE,Université de Paris
mourad.ouziri@u-paris.fr

Salima Benbernou
LIPADE,Université de Paris
salima.benbernou@u-paris.fr

Karima Echihabi
Mohammed VI Polytechnic University
karima.echihabi@um6p.ma

Themis Palpanas
LIPADE, Université de Paris
French University Institute IUF
themis@mi.parisdescartes.fr

Iheb Ben Amor
IMBA Consulting
iheb.benamor@imbaconsulting.com

ABSTRACT

In this paper, we present a comprehensive study that evaluates six state-of-the-art sentiment analysis tools on five public datasets, based on the quality of predictive results in the presence of semantically equivalent documents, i.e., how consistent existing tools are in predicting the polarity of documents based on paraphrased text. We observe that sentiment analysis tools exhibit *intra-tool inconsistency*, which is the prediction of different polarity for semantically equivalent documents by the same tool, and *inter-tool inconsistency*, which is the prediction of different polarity for semantically equivalent documents across different tools. We introduce a heuristic to assess the data quality of an augmented dataset and a new set of metrics to evaluate tool inconsistencies. Our results indicate that tool inconsistencies is still an open problem, and they point towards promising research directions and accuracy improvements that can be obtained if such inconsistencies are resolved.

PVLDB Reference Format:
Wissam Mammar Kouadri, Mourad Ouziri, Salima Benbernou, Karima Echihabi, Themis Palpanas, and Iheb Ben Amor. Quality of Sentiment Analysis Tools: The Reasons of Inconsistency. PVLDB, 14(4): 668 - 681, 2021.
doi:10.14778/3436905.3436924

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/AbCd256/Quality-of-sentiment-analysis-tools.git.

1 INTRODUCTION

With the growing popularity of social media, the internet is replete with sentiment-rich data like reviews, comments, and ratings. A sentiment analysis tool automates the process of extracting sentiments from a massive volume of data by identifying an opinion and deriving its polarity, i.e., Positive, Negative, or Neutral. In the last decade, the topic of sentiment analysis has also flourished in the research community [15, 22, 26, 29, 31, 39, 69, 76, 80, 82], and has also attracted the attention of the data management community, which has studied problems related to polarity aggregation, opinion mining, sentiment correlation, fact-checking, and others [2, 3, 23, 49, 56, 64, 67, 68, 70, 71, 74, 75, 82]. Therefore, organizations are showing great interest in adopting sentiment analysis tools to exploit this data for decision making. For instance, it is used in politics for predicting election results, in marketing to measure user's satisfaction [68], in crowdsourcing to measure workers' relevance [63], and in the health care domain [36]. Nevertheless, sentiment analysis of social media data is still a challenging task due to the complexity and variety of natural language, through which the same idea can be expressed and interpreted using different words. Let us illustrate this issue by considering two texts (documents) expressed differently, but having the same meaning: *(1) China urges the US to stop its unjustifiable crackdown on Huawei.*; and (2) *China urges the United States to stop its unreasonable crackdown on Huawei.* We notice that although the two documents are structured differently, they are, in fact, semantically equivalent paraphrases, because they convey the same meaning.

The sentiment analysis research community has adopted the consensus that semantically equivalent text should have the same sentiment polarity[8, 19, 28, 62, 72, 77]. For example, [19] proposed an approach that learns the polarity of affective events in a narrative text based on weakly-supervised labels, where the semantically equivalent pairs (event/effect) got the same polarity, and opposite pairs (event /effect) got opposite polarities. Authors in [26] have extended their dataset with opinion paraphrases by labeling the generated paraphrases with polarity labels of the original text. However, recent works show that sentiment analysis systems assign different polarity labels to semantically equivalent documents, i.e., those systems are predicting different outputs for semantically equivalent inputs [1, 9, 35, 44, 48, 61, 73]. Such documents are called *adversarial examples*. The latest works propose new tools that generate adversarial examples to extend datasets, which lead to more general and robust models [1, 35, 61]. The work in [35] proposes a syntactically controlled paraphrase generation model, and the one in [1] generates semantically equivalent documents using population genetic optimization algorithms. A rule-based framework that generates semantically equivalent adversarial examples is also proposed in [61].

Proceedings of the VLDB Endowment, Vol. 14, No. 4 ISSN 2150-8097.
doi:10.14778/3436905.3436924

In this work, we conduct an empirical study that quantifies and explains the non-robustness of sentiment analysis tools in the presence of adversarial examples. Our evaluation uncovers two different anomalies that occur in sentiment analysis tools: *intra-tool inconsistency*, which is the prediction of different polarity for semantically equivalent documents by the same tool, and *inter-tool inconsistency*, which is the prediction of different polarity for semantically equivalent documents across different tools. In our in-depth analysis, we offer a detailed explanation behind such inconsistencies. Existing works on [1, 35, 53, 73] have typically focused on generating data for adversarial training and for debugging models [61]. In our work, we focus on studying the problem from a data quality point of view, and understanding the reasons behind the aforementioned inconsistencies. We also demonstrate that sentiment analysis tools are not robust in the presence of adversarial examples, regardless of whether they are based on machine learning models, lexicons, or rules.

In summary, we make the following contributions.

• We address the problem of adversarial examples in sentiment analysis tools as a data quality issue (Sections 3 and 5), evaluate the prediction quality of tools on two levels (intra-tool and inter-tool), and review five datasets in terms of quality and consistency (Section 5.2). Our experiments reveal that inconsistencies are frequent and present in different configurations, so we define a non-exhaustive list of causes that may influence the inconsistency.

• We build a benchmark for paraphrases with polarity labels to evaluate the consistency and the accuracy of sentiment analysis tools and propose a method to reduce the cost of the benchmark quality Section 5.1.

• We survey six state-of-the-art sentiment analysis algorithms (Section 4) and provide an in-depth analysis of them in the presence of adversarial examples across different domains (Sections 5.3 and 5.4).

• We propose new metrics that allow the evaluation of intra-tool and inter-tool inconsistencies (Section 5.1).

• We present a rich discussion of our experimental results and show the effects of reducing inconsistencies on the accuracy of sentiment analysis tools (Section 5.4).

• We propose recommendations for choosing sentiment analysis tools under different scenarios (Section 6), which helps the data managers select the relevant tool for their frameworks. For the sake of reproducibility, we make our code and datasets publicly available[1].

2 RELATED WORK

The problem of inconsistency in sentiment analysis has attracted the interest of researchers in the data management community [23, 56, 70]. Studies performed on this problem can be classified into two categories: time-level inconsistency (or sentiment diversity), and polarity inconsistency represented by both intra-tool and inter-tool inconsistencies. For instance, authors in [70] have studied the inconsistency of opinion polarities (on a specific subject) in a given time window, as well as its evolution over time, and propose a tree-based method that aggregates opinion polarities by taking into consideration their disagreements at large scale. Authors in [56] proposed a set of hybrid machine learning methods that rank tweet events based on their controversial score (inconsistency score) in a time window. As an example of polarity inconsistency work, authors in [23] have studied the polarity consistency on sentiment analysis dictionaries on two levels (intra- and inter-dictionaries). Hence, their study has focused only on one type of sentiment analysis tools (lexicon-based methods with word dictionary), while our study is more thorough and includes a wider range of algorithms (rule-based, machine learning-based and concept lexicon methods) and data. Besides, our study is performed on the document level, and analyzes the tool inconsistency from different points of view: statistical, structural, and semantic. Other studies about intra-tool inconsistency have been limited on earlier research that studied sentiment preservation in translated text using lexicon-based methods [4, 10]. The results of our study do not contradict previous findings, but confirm, deepen and generalize them.

To the best of our knowledge, this is the first study that carries out a comprehensive structural and semantic analysis on large scale datasets for sentiment analysis tools for different domains and varieties of algorithms.

3 PROBLEM STATEMENT

3.1 Motivating Example

Our research work is motivated by a real observation from social media. We present here a data sample collected from twitter about Trump's restrictions on Chinese technology.

Let $D = \{d_1, \ldots, d_8\}$ be a sample set of documents/tweets such that:

• d_1 : Chinese technological investment is the next target in Trump's crackdown. d_2 : Chinese technological investment in the US is the next target in Trump's crackdown.

• d_3 : China urges end to United States crackdown on Huawei. d_4 : China slams United States over unreasonable crackdown on Huawei. d_5 : China urges the US to stop its unjustifiable crackdown on Huawei.

• d_6 : Trump softens stance on China technology crackdown. d_7 : Trump drops new restrictions on China investment. d_8 : Donald Trump softens tone on Chinese investments.

Note that D is constructed with subsets of semantically equivalent documents. For instance, d_1 and d_2 are semantically equivalent as they both express the idea that the US is restricting Chinese technological investments. We denote this set by A_1 and we write: $A_1 = \{d_1, d_2\}$ and $A_2 = \{d_3, d_4, d_5\}$, which express that the Chinese government demands the US to stop the crackdown on Huawei, and $A_3 = \{d_6, \ldots, d_8\}$ which convey the idea that Trump reduces restrictions on Chinese investments. A_i are partitions of D, that is: $D = \bigcup_{i=1}^{n} A_i$ and $\bigcap_{i=1}^{n} A_i = \emptyset$. We analyse D using four sentiment analysis tools: The Stanford Sentiment Treebank [66], Senticnet5 [8], Sentiwordnet [25] and Vader [29]. We associate to those sentiment analysis tools the following polarity functions respectively: P_{rec_nn}, $P_{senticnet}$, $P_{sentiwordnet}$, P_{vader}, and P_h is associated to human annotations as a ground truth. Table 1 summarizes the results of the analysis. Note that different tools can attribute different sentiment labels for the same document (for e.g., only P_{rec_nn} attributes the correct label to d_3 in A_2) and the same tool can attribute different

[1] https://github.com/AbCd256/Quality-of-sentiment-analysis-tools.git

Table 1: Predicted polarity on dataset D by different tools

A_i	Id	P_{rec_nn}	$P_{senticnet}$	$P_{sentiwordnet}$	P_{vader}	P_h
A_1	d_1	Neutral	Negative	Negative	Neutral	Negative
	d_2	Negative	Negative	Negative	Neutral	Negative
A_2	d_3	Negative	Positive	Positive	Neutral	Negative
	d_4	Negative	Positive	Negative	Neutral	Negative
	d_5	Negative	Positive	Negative	Neutral	Negative
A_3	d_6	Neutral	Positive	Positive	Neutral	Positive
	d_7	Negative	Negative	Positive	Neutral	Positive
	d_8	Neutral	Positive	Negative	Neutral	Positive

labels for semantically equivalent documents (for e.g., $P_{senticnet}$ considers d_6 as Positive and d_7 as Negative).

3.2 Definitions and Terminology

3.2.1 Sentiment Analysis. **Sentiment Analysis** is the process of extracting the quintuplet $< E_i, F_{ij}, H_z, O_{ijz}, t >$ from a document d, such that: E_i is an entity i mentioned in document d, F_{ij} is the j^{th} feature of the entity i, H_z is the holder or the person who gives the opinion O_{ijz} on the feature ij at the time t [45]. Notice that the same document can contain several entities and that each entity can have a number of features.

In our work, we consider sentiment analysis as the assignment $< d_i, O_i >$ where a polarity label O_i is extracted from document d_i.

3.2.2 Polarity. **Polarity.** We call polarity the semantic orientation of the sentiment in the text. Polarity can be Negative, Neutral or Positive and is denoted by $p \in \Pi$ with $\Pi = \{Negative, Neutral, Positive\}$.

Polarity function. For each tool t_k, we associate a polarity function P_{t_k} defined as: $\forall P_{t_k} \in \Gamma,\ P_{t_k} : D \rightarrow \Pi$, with Γ the set of all polarity functions and D a set of documents. Otherwise, this function assigns to each document d_i a polarity label.

Ground truth. We denote by P_h the polarity given by human annotation that we consider as the ground truth.

Polar Word. We define a polar word as a word that has a polarity different from Neutral ($P_h(w_j) \neq Neutral$).

Opinionated Document. Let's consider a document d_i as a set of words such that $d_i = \{w_1, \ldots, w_m\}$. An opinionated document is a document that contains a polar word, formally: $\exists\ 1 \leq j \leq m, such\ that\ P_h(w_j) \neq Neutral$.

Polar Fact/Objective Document. A polar fact, a.k.a an objective document, is a document that does not contain a polar word, i.e., d_i is a polar fact iff: $\forall w_j \in d_i, such\ that\ 1 \leq j \leq m,\ P_h(w_j) = Neutral$.

3.2.3 Analogical Set. **Analogical Set**. is a set of semantically equivalent documents: $A_l = \{d_1, \ldots, d_n\}$ s.t: $\forall d_i, d_j \in A_l\ , d_i \overset{s}{\Longleftrightarrow} d_j$ where $\overset{s}{\Longleftrightarrow}$ denotes semantic equivalence, and d_i and d_j are called analogical documents.

3.2.4 Sentiment Consistency. For each dataset D and polarity functions set Γ, we define sentiment consistency as the two rules 1) and 2):

1) Intra-tool consistency. Intra-tool consistency assesses the contradiction of tools when considered individually. In other words, there is an Intra-tool inconsistency regarding a tool P_{t_k} if it assigns different polarities to two analogical documents d_i and d_j. The intra-tool consistency is defined as:

$$\forall A_l\ \forall d_i, d_j \in A_l\ \forall P_{t_k} \in \Gamma,\ P_{t_k}(d_i) = P_{t_k}(d_j) \tag{1}$$

We call **adversarial examples** the documents that belong to the same analogical set and that violate intra-tool consistency. $d_i\ d_j$ are adversarial for P_* if:

$$P_*(d_i) \neq P_*(d_j)\ s.t:\ P_* \in \Gamma \tag{2}$$

For example, documents d_6 and d_7 in Table 1 violate intra-tool consistency so they are adversarial examples for the Senticnet sentiment analysis tool.

2) Inter-tool consistency. Inter-tool consistency assess the contradiction between the tools. In other words, there is an Inter-tool inconsistency between the tools P_{t_k} and $P_{t'_k}$ if they assigns different polarities to a given document d_i. The inter-tool consistency is defined as:

$$\forall A_l\ \forall d_i \in D\ \forall P_{t_k}, P_{t'_k} \in \Gamma \quad P_{t_k}(d_i) = P_{t'_k}(d_i) \tag{3}$$

In the example 3.1, the inter-tool consistency is not respected for the document d_3, since we have: $P_{rec\_nn}(d_3) \neq P_{senticnet}(d_3)$. We call this case inter-tool inconsistency.

Inconsistency classes. We distinguish three classes of inconsistencies based on the predicted polarity of analogical documents: (1) Inconsistency of type Positive/Neutral, (2) Inconsistency of type Negative/Neutral and (3) Inconsistency of type Negative/Positive.

4 APPROACHES

In this section, we describe the six popular sentiment analysis tools that we cover in our study. They are representative tools from the state-of-the-art, based on their accuracy, polarity classes they predict, and methods they are based on (lexicon-based, rule-based and machine learning-based).

4.1 Lexicon-Based Approaches

This family of approaches is considered a trivial solution to sentiment analysis. It encompasses unsupervised methods that use a word matching scheme to find the sentiment of a document based on a predefined dictionary of polar words, concepts and events/effects [12, 13, 18, 19]. We propose a description of the general approach adopted by lexicon-based methods (template), displayed in Algorithm 1. To represent this class of methods, we select Sentiwordnet [25] because it is a widely used lexicon for sentiment analysis [17, 19, 32, 41, 54]. It is based on Wordnet [52] and is annotated automatically using a semi-supervised classification method. It uses a ternary classifier (a classifier for each polarity label) on Wordnet that is trained by a dataset of labeled synsets (set of word's synonyms). The dataset is created automatically by clustering the Wordnet synsets around small samples of manually labeled synsets.

We run Algorithm 1 by using Sentiwordnet as a lexicon on document d_8 from Example 3.1. First, we represent the document in a bag-of-words model to tokenize it, **(step 1)**: Donald:1, Trump:1, soften:1, tone:1, on:1, Chinese:1, investments:1. where the numbers represent the frequency of tokens in the sentence. Then we search the sentiment score of each token in the dictionary, **(step 2)** we find: $list = \{0, 0, 0, -0.125, 0, 0, 0\}$ (we multiply the polarity by the effective of the token). We aggregate tokens' polarities, by calculating the mean of the polarity scores **(step 3)**, and get the polarity label associated with these scores **(step 4)**, we find that: $P_{sentiwordnet}(d_8) = Negative$.

Algorithm 1 Lexicon-Based Sentiment Analysis

Input : d_i : *Document*
Output : Polarity label $l_i \in \Pi$

procedure LexiconBased
  //**step1:***Tokenize the document*
    $W = tokenize(d_i)$
  //**step2:** *Calculate the polarity of each token in W*
    for each $w_i \in W$:
      $list.add(P_*(w_i))$
  //**step3:***Aggregate words polarities (using mean*
  // in most cases)
    $score = mean(list)$
  //**step 4:***Get the polarity label based on*
  //the polarity score
    $l = getLabel(score)$
  return l

4.2 Rule-Based Approaches

Rule-based methods are similar to lexicon-based methods with an additional inference layer that allows deducing the correct polarity of documents based on linguistic rules. First, we create a lexicon of words, concepts, or event/effects. Then, a layer of syntactic and part-of-speech rules is added to allow the inference of the correct polarity label. For example, the Vader tool [29] uses the following syntactic rules: 1) If all characters of a polar word are in upper case (e.g. GOOD), the polarity intensity increases by 0.733; and 2) The polarity intensity decreases in the text preceding the conjunction "but" by 50% and increases by 50% in the text that follows. Senticnet [8] is a semantic network of concepts (lexicon with relations between concepts) that has a layer for polarity inference. It uses linguistic rules named "sentic patterns" to infer the right polarity of the sentence based on its structure. As an example: if the concept is in a polarity switchers scoop, such as negation, its polarity will be inverted from Positive to Negative. We propose Algorithm 2 which is the template adopted by rule-based methods.

Algorithm 2 Rule-Based Sentiment Analysis

Input: d_i: Document
Output: Polarity label $l_i \in \Pi$

procedure RuleBased
  // **Step1**: Split the sentence into units(words,
  //concepts, event/effect)
    $W = tokenize(d_i)$
  // **Step2**: Get the polarity of each token in W from
  //the semantic network/lexicon
    for each $w_i \in W$:
      $list.add(P_*(w_i))$
  // **Step3**: Infer the final polarity of tokens using
  //the set of defined syntactic rules s
    $list = infer\_polarity(list, s)$
  // **Step4**: Aggregate words polarities
    $score = aggregate(list)$
  // **Step5** : Get the polarity label based on the
  //polarity score
    $l = getLabel(score)$
  return l

Consider the document "China does not want to supplant the US, but it will keep growing" that we analyze using Senticnet:**(step 1)**, Senticnet divides this sentence into concepts using the $tokenize$ function, which produces the list W of tokens (a.k.a. concepts): $W = \{china, supplant\_US, keep\_grow\}$ **(step 2)**, Senticnet determines the polarity of each concept as a list of couples $< Concept : polarity >$:
$list = \{china : Neutral, supplant\_US : Positive, keep\_grow : Positive\}$ **(step 3)**, the function $infer\_polarity$ activates Sentic patterns to infer each concept's polarity. The concept $< Supplant\_US >$ is in the scoop of a polarity switching operator, which inverts the polarity of this concept to "Negative" **(step 4)**, when aggregating the concepts polarities, Senticnet encounters the conjunction $< but >$; therefore, it infers the polarity as being Positive because it applies the following adversary rule defined in Sentic patterns: the polarity of the document is equivalent to the polarity of the text following the conjunction $< but >$.

4.3 Learning-Based Approaches

Recently, researchers have widely explored deep learning models [21, 34, 46, 57, 65, 66] in sentiment analysis to exploit their ability to extract the relevant features for sentiment analysis from the text. For instance, [66] has proposed a Recursive Neural Network model (RecNN) for sentiment classification. These methods learn a vector representation of sentences (sentence embedding) through a recursive compositional computation, then feed it to a Softmax sentiment classifier.

On the embedding level, a standard Convolutional Neural Network (Text_CNN) was proposed in [37] for text classification tasks, including sentiment analysis. It uses a word embedding model pre-trained on the Google News corpus [51], which guarantees a good coverage of the language space and catches the semantics of the sentence. Authors in [21] have refined the sentence embedding level and proposed a character to sentence CNN for short text classification (Char_CNN) that has two convolutional layers that allow handling sentiment features at both the character and sentence levels (word order, character characteristics). Since on social media, the way the word is written may embed a sentiment.

We propose the descriptions of the methods that use CNNs and follow the general template displayed in Algorithms 3 for training and 4 for classification. CNNs perform document compositionality and transform the matrix document into a vector of features used by a classifier. Thus, in this case, we first split the document d_i to a sequence of n words such that $d_i = \{w_{i1}, \ldots, w_{in}\}$. Then, we represent each word as a vector to obtain a matrix representation of the document. Many research works [21, 37, 43, 65] have used a word embedding model for this task. Other methods [51, 55] use a pre-trained model to initialize the word embedding vectors while others initialize the vectors randomly. In this case, our document d_i will be represented as: $d_i = \{U_{i1}, \ldots, U_{in}\}$ Where $U_{ij} \in \mathbb{R}^k$ k is the size of the embedding vector, $U_{ij} = g(w_{ij})$, and g is a word embedding function. We concatenate vectors to obtain the matrix representation of the document as follows: $M_i = U_{i1} \oplus U_{i2} \oplus \cdots \oplus U_{in} M_i \in (\mathbb{R})^{k \times |d_i|}$

This matrix is the input of the convolution layer, which extracts features from the matrix M_i by applying convolutions between

Algorithm 3 (CNN for Sentiment Analysis (training))

Input :- training dataset $D = \{d_1, \ldots, d_n\}$
-L: Hyper parameters list//batch size,
window size,number of filters …
Output: Trained model

procedure TRAINING
  //Step1: initialize weights
  //Step2: Document representation
  //(word embedding)
  Foreach $d_i \in D$:
    $M_i = \{U_{ij} | \forall w_{ij} \in d_i, u_{ij} = g(w_i)\}$
    $list.add(M_i)$
  // **Step4**: Training network
  For nb_epoche **do** :
    Convolution: Equation 4
    Poolling
    Classifier
    Optimization (backpropagation) Equation 5

filters W and the input matrix on a window of size l. The convolution operation is defined by:

$$z_j = h(W.M_i[j, j + l - 1] + b) \tag{4}$$

Algorithm 4 CNN for Sentiment Analysis (Classification)

Input : d_i : Document, $model$
Output : Polarity label $l \in \Pi$

procedure LEARNINGBASEDPREDICTION
  $load(model)$
  Convolution: Equation 4
  Pooling
  $v = Classifier$// returns a value v
  return get_polarity(v)

where h is the activation function that could be $tanh$, $ReLU$,or $Softmax$, z_j is the features map (resulted features), $W \in \mathbb{R}^{l \times o}$ where l and o are hyperparameters. Typically, $o = |d_i|$. W is the filter to learn and b is the bias. Pooling is applied to extract only the significant features. After that, the extracted vector of features is fed into a normal classifier (Neural Network, SVM, Softmax, etc.). The filters are the parameters to learn. The training process consists of using forward-propagation then updating weights using stochastic back-propagation with the objective of maximizing the log likelihood[2]:

$$max \sum_{i=1}^{n} log\ pr(P_*(d_i)|d_i, \theta) \tag{5}$$

5 EXPERIMENTS

In this section, we conduct a thorough experimental evaluation of the robustness of existing sentiment analysis tools in the presence of adversarial examples. First, we introduce our experimental setup; then, we evaluate the different techniques based on a statistical analysis followed by an in-depth structural and semantic analysis. For the sake of reproducibility, we make publicly available our datasets and codebases. All our experiments are implemented in python and java, and were conducted on a server with 75GB of RAM and two Intel Xeon E5-2650 v4 2.2GHz CPUs.

[2]Equivalently, some methods [21] minimize the Negative likelihood.

Table 2: Comparison of sentiment analysis tools

Technique	Method	Mean Accuracy	Test Dataset
Lexicon based method	Sentiwordnet [25]	Binary: 51% [17]	MPQA dataset [78]
Rule based methods	SenticNet [8]	Binary: 93.7%	Blitzer [7] Movie reviews [47]
	Vader [29]	Fine grained:79%	Amazon Reviews Movie Reviews Tweets NY times Editorials
Machine learning (RNN)	RecNN[66]	Fine grained : 45% Binary: 85.4%	Sentiment tree bank[66]
Machine learning (CNN)	CNN[37]	Fine grained: 48% binary: 84.5%	Movie Reviews[47] sentiment tree bank [66] Costumer reviews [50]
	CharCNN[21]	Fine graind: 48.7% Binary : 85.8%	Sentiment treebank [66] stanford twitter Sentiment [30]

5.1 Experimental Setup

Tools. In our experiments, we use the sentiment analysis tools summarized in Table 2. We use SentiWordnet [25] as a representative method for the lexicon-based category because of its popularity and coverage. We also use two representative methods for rule-based approaches: Vader [29], whose high quality has been verified by human experts, and SenticNet [8], which is a lexicon of concepts. Using these different lexicons allows us to evaluate inconsistencies on two different levels: concepts and word level. In the learning-based methods, we use three machine learning tools: RecNN [66] that learns word embeddings, Text_ CNN [37] that uses a pre-trained word embedding method, and Char_CNN [21], that uses two levels of embedding. All these methods are powerful and widely used, and use different embedding types, which allows us to study the relationship between the embedding type and inconsistencies. We note that we implemented the two methods [21] and [37], and retrained the networks on the used datasets. In the rest of the paper, we refer to sentiment analysis tools by their polarity functions, i.e., SenticNet as $P_{senticnet}$, Sentiwordnet as, $P_{sentiwordnet}$, Vader as P_{vader}, Text_CNN as P_{text_cnn}, Char_CNN as P_{text_cnn}, and RecNN as P_{rec_nn}.

Benchmark. To evaluate the tools' inconsistencies and accuracy, a dataset of analogical sets with polarity labels is needed. To the best of our knowledge, this dataset does not exist publicly. Therefore, we built a benchmark of analogical sets labeled with polarities based on publicly available sentiment datasets that we augmented with paraphrases. This method allows generating analogical datasets with a size up to ten documents, instead of the straight-forward method which consists of labeling paraphrases datasets manually. For this, we extended five publicly available datasets using the syntactically controlled paraphrase networks (SCPNs) [35].

Base Datasets. The five datasets were chosen based on their popularity [11, 16, 27, 34, 37, 38, 76, 79], size and having at least three polarity labels. (1) The *News Headlines* [14] dataset contains short sentences related to financial news headlines collected from tweets and news RSS feeds and annotated by financial news experts with polarity values $p \in [-1, 1]$. (2) The *Stanford Sentiment Treebank (SST)* [66] is built based on movie reviews and annotated using

Algorithm 5 Clean_dataset

Input: $D = \{A_1 = [S_{11}, \ldots, S_{1m}], \ldots, A_n = [S_{n1}, \ldots, S_{nm'}]\}$
Golden dataset B
Output: Clean dataset

```
//Step1:Get the distance threshold
t = Calculate_threshold(B)
//Step2:Verify if generated elements
respect the max threshold t
for each A_i ∈ B :
    S_o = A_i[0]
        for each S_ik ∈ A_i :
            if WMD(S_ij, S_o) < t :
                NewA.add(S_ij)
    list.add(NewA)
return list
```

Mechanical Turk crowdsourcing platform [3] with value $o \in [0, 1]$. (3) The *Amazon Reviews Dataset [33]* contains product reviews from the Amazon websites, including text reviews and the product's five-level rating as a polarity score. (4) The US airlines tweets dataset[4] that contains tweets about the US airline companies and annotated based on the emojis present in the tweets. (5) The first GOP debate dataset[5] that contains tweets related to the debate between Trump and Clinton. We refer to these datasets as *News*, *SST*, *Amazon*, *US airlines*, *FGD* datasets, respectively, in the rest of the paper. We also consider another dataset, *Microsoft research paraphrases corpus* [59], that contains valid paraphrases.

Augmented Datasets. We extend the previously mentioned sentiment labeled datasets using the method described in [35], that automatically generates syntactically controlled paraphrases for each document (between 2 and 10 documents) that should have the same polarity.

Refined Datasets. The problem with the tool in [35] is that it produces wrong predictions for 20% of the cases, which may affect the quality of the analysis. To resolve this problem and to improve the quality of the benchmark by reducing the produced error rate, we propose a protocol that minimizes the human effort and the cost of data quality verification. The two-steps protocol is described in Algorithm 5 to obtain a clean datasets and Algorithm 6 to calculate the similarity threshold needed in step 2 of the protocol.

We consider two analogical datasets B, and D, where B is the valid set of paraphrases from the Microsoft research paraphrases corpus [59] and D is generated automatically. We calculate the Word Mover Distance (WMD) [42] between each pair of paraphrases in B. For that, we consider the word2vec embedding matrix $X \in R^{m \times n}$ such that m is the size of the word embedding vector [51]. The distance between two words is the euclidean distance called distance transportation cost (c). We define also the transformation matrix $T \in R^{n \times n}$ which represents the proportion of each word in the transformation of the document d to the document d'. The weights of this matrix must verify the condition: $\sum_{i=0}^{n} T_{ij} = d_i$ and $\sum_{j=0}^{n} T_{ij} = d'_j$. Then, WMD is the solution of the objective function: $Min \sum_{i=0}^{n} T_{ij} c(i, j)$ such that $\sum_{i=0}^{n} T_{ij} = d_i$ and $\sum_{j=0}^{n} T_{ij} = d'_j$. Once the WMD has been calculated between all couples of paraphrases

[3]https://www.mturk.com/
[4]https://www.kaggle.com/crowdflower/twitter-airline-sentiment
[5]https://www.kaggle.com/crowdflower/first-gop-debate-twitter-sentiment

Algorithm 6 Calculate_threshold

Input: Golden dataset $B = \{A_1 = [S_{11}, S_{12}], \ldots, A_n = [S_{n1}, S_{n2}]\}$
Output: Similarity threshold t

```
for each A_i ∈ B :
//Step1:calculate WMD distance
    dist = WMD (A_i[0], A_i[1])
    list.add(dist)
//Step2: Calculate the threshold distance
t= median(list)
return t
```

Table 3: Statistics of used datasets.

Statistics	# elements	# Clusters	# Positive	# Neutral	# Negative
Amazon	21483	10704	16944	2028	2511
News	1580	831	890	36	654
SST	3504	1033	1492	628	1384
FGD	28192	9298	4319	6296	1384
US airlines	38236	12894	5107	6816	26313
Total	92995	34760	28752	15804	48439

in B, we calculate the population's median value that we consider as the threshold distance. The main steps of this part of the method are described briefly in Algorithm 6.

After defining the max distance threshold t, we calculate the WMD distance between each two document S_{ij} and S_{ik} in analogical sets A_i of D. Then, we only keep the generated documents that have a distance lower than t. The output after this step is a refined dataset of analogical sets containing semantically close documents. In the rest of the paper, we only use the refined datasets, while for training machine learning models, we use the base datasets. For simplicity, we refer to generated datasets using the same name as the base datasets. For example, *SST* refers to the extended, then refined version of the dataset *SST*. The statistics about our benchmark are displayed in Table 3.

Metrics. In this section, we present different metrics used in our experiments besides WMD described previously:

- We evaluate *Accuracy* by calculating the rate of correctly predicted polarities compared to the golden polarity (human annotation) as follows: $Accuracy = \frac{\sum_{i=0}^{n} \mathbb{1}_{(P_h(d_i)=P_*(d_i))}}{n}$.
- *Cos similarity* between two vectors V and U: $Cos\_sim(V, U) = \frac{U.V}{\|V\|.\|U\|}$.
- We evaluate the intra-tool inconsistency rate for each document d_j in the analogical set A and sentiment analysis tool t_k as the proportion of documents d_z that have a different polarity than $P_{t_k}(d_i)$ with regards to the tool t_k. We write

$$\forall d_i \in A, P_{t_k} \in \Gamma, inc_{in}(d_i, P_{t_k}) = \frac{card(S)}{n-1} \quad \text{s.t } S = \{d_j \in A | P_{t_k}(d_i) \neq P_{t_k}(d_j)\} \tag{6}$$

The intra-tool inconsistency rate of an analogical set A is the mean of different intra-tool inconsistency rates of its documents:

$$inc_{in}(A, P_{t_k}) = \frac{\sum_{j=1}^{n} inc_{in}(d_j, P_{t_k})}{n}, n = card(A) \tag{7}$$

- We measure the *inter-tool inconsistency rate* for each tool t_k and document d_j as the rate of tools $t_{k'}$ that give different polarities

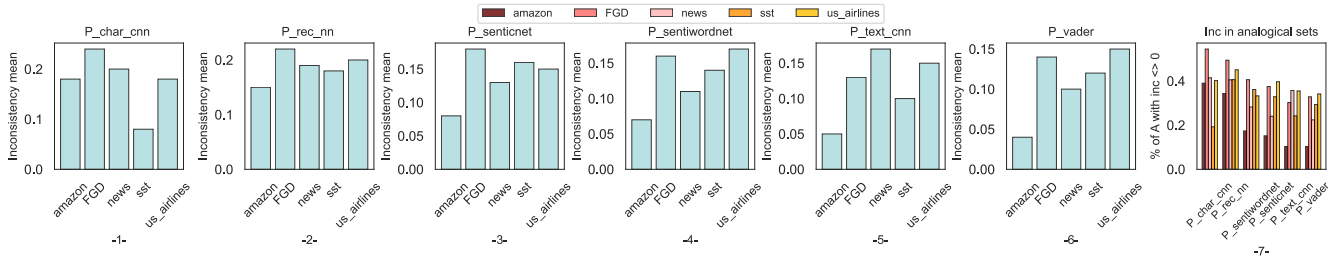

Figure 1: Intra-tool inconsistency distribution

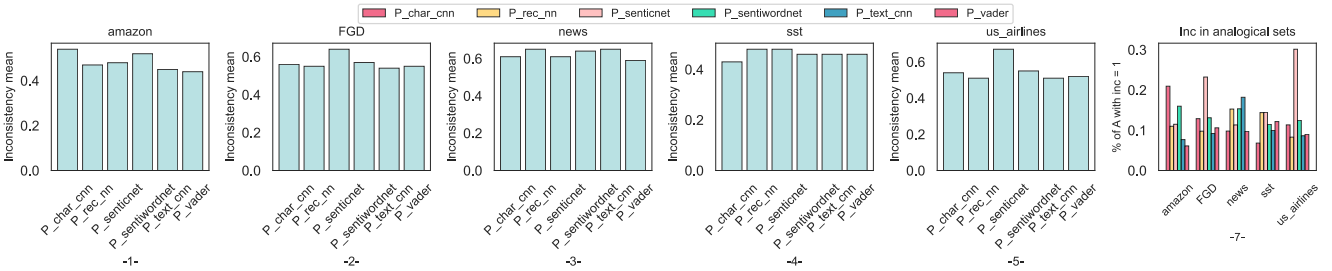

Figure 2: Inter-tool inconsistency rate

to the document d_j than P_{t_k}. We write:

$$\forall P_{t_k} \in \Gamma, \forall d_i \in A,\ inc_{inter}(d_i, P_{t_k}) = \frac{card(S')}{m-1}$$
$$\text{s.t } S' = \{P_{t'_k} \in \Gamma | P_{t'_k}(d_j) \neq P_{t_k}(d_j)\} \quad (8)$$

The inter-tool inconsistency rate in the set Γ is the mean of inconsistency rates of the different tools:

$$inc_{inter}(d_j, \Gamma) = \frac{\sum_{k=1}^{m} inc_{inter}(d_j, P_{t_k})}{m},\ m = card(\Gamma) \quad (9)$$

5.2 Statistical Analysis

Our goal is to answer the following questions: (1) Is intra-tool inconsistency a rare event? (2) What types of inconsistencies exist? (3) Are there domains that are more prone to inconsistencies?

5.2.1 Intra-tool Inconsistency Rate. The purpose of this experiment is to evaluate the intra-tool inconsistencies that occur in different sentiment analysis tools. For this, we calculate the inconsistencies in each analogical set using Equation (7), then we calculate the mean inconsistency of tools in each dataset. We display the results in Figure 1. Sub-figures 1-6 represent the tools' mean inconsistency on datasets, while sub-figure 7 shows the proportion of analogical set that have an intra-tool inconsistency different from 0. We notice that 44% of analogical sets have an intra-tool inconsistency different from 0 on P_{char_cnn}, 40% on P_{text_cnn}, 26.6% on $P_{sentiwordnet}$, 33.6% on $P_{senticnet}$, 44.8% on P_{rec_nn} and 8.5% on P_{vader}. We notice a high intra-tool inconsistency degrees on the machine learning-based methods P_{char_cnn} and P_{rec_nn} (an average of 0.17 and 0.188 respectively), while we notice lower inconsistency degrees on lexicon and rule-based method ($P_{sentiwordnet}$ 0.104, and P_{vader} 0.11). We notice also a low inconsistency degree of 0.11 on P_{text_cnn}. So we can deduce that lexicon-based methods are more consistent than machine learning methods. We also notice that tools are more inconsistent on the *FGD* and *US airlines* datasets. We notice also high inconsistency on *SST* even for the P_{rec_nn} that was trained on *SST* dataset.

Summary: intra-tool inconsistencies are frequent on different sentiment analysis tools, which motivate a more in-depth study to cover the causes of such anomalies in the next sections (Sections 5.3 and 5.4).

5.2.2 Inter-tool Inconsistency Rate. In this experiment, we calculate the inter-tool inconsistency degree according to Equation (8), then we calculate the main inter-tool inconsistency of tools on the datasets. Sub-figures 1-6 of Figure 2 report the mean inter-tool inconsistency between tools on our datasets, and Sub-figure 7 shows the percentage of documents where tools have an inconsistency degree of 1. We notice that there is a large degree of inter-tool inconsistency between different sentiment analysis tools. For example, the mean inter-tool inconsistency is 0.535 on P_{char_cnn}, 0.52 on P_{rec_nn}, 0.575 on $P_{senticnet}$, 0.547 on $P_{sentiwordnet}$, and 0.53 on P_{text_cnn}. We notice also that P_{char_cnn} contradict all tools (inconsistency degree =1) on 12.35% of cases, $P_{sentiwordnet}$ on 11.7%, P_{text_cnn} on 18.2% and P_{vader} on 13.7%.

The highest inter-tool inconsistencies are on the *News* dataset with a mean inconsistency degree of 0.62, which can be explained by the challenging nature of the documents present in this dataset which are short in length, factual, and meaningful. Therefore it is difficult to catch the polarity features from the sentences.

Summary: These experiments show that inter-tool inconsistency is a frequent event that occurs with a high degree on the Financial *News* dataset that represents relevant data for critical decision-making processes. This motivates us to refine our analysis to unveil and explain the causes behind these inconsistencies in the next sections (Sections 5.2.4 and 5.2.5).

5.2.3 Intra-tool Inconsistency Type. In this experiment, we calculate the mean of polarity inconsistencies by classes (see definition 3.2.4). We consider the document consistency in this case as a

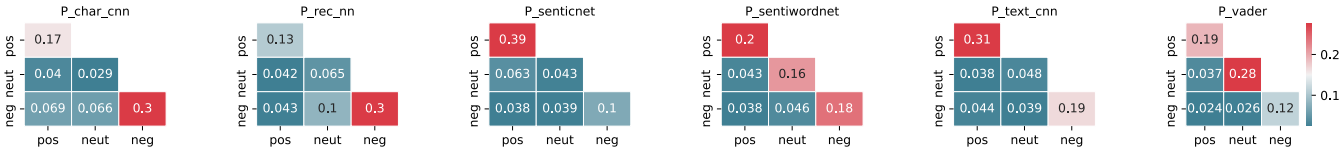

Figure 3: intra-tool inconsistency type

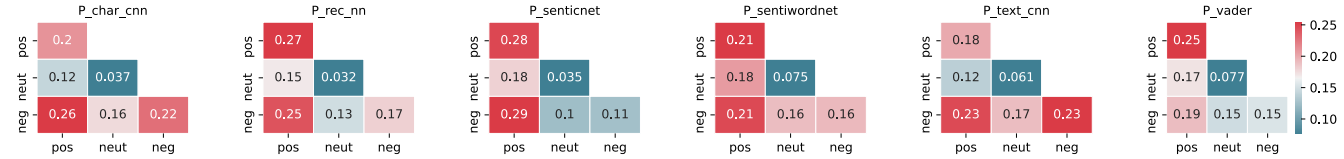

Figure 4: Inter-tool inconsistency type

$M \in R^{3\times3}$ matrix where M_{ij} is the proportion of equivalent documents that got respectively the polarities i and j by the sentiment analysis tool k formally: $M_{ij} = \frac{card(S)}{n}$ such that for a document d_z, $S = \{d_{z'} \in A | P_k(d_z) = i \wedge P_k(d_{z'}) = j\}$. The consistency matrix of the analogical set A_i is the average of documents consistency matrices in A_i, and the inconsistency matrix of the dataset D is the average of inconsistency matrices of all analogical sets in D.

In Figure 3, we display the intra-tool inconsistency matrices for each tool and dataset (inconsistencies are the no diagonal values). This experiment demonstrates that the inconsistencies are not only ones of type (Neutral/Positive)[6], or (Neutral/Negative). In fact, an important proportion of inconsistencies are of type (Positive/Negative). For instance, most inconsistencies on P_{char_cnn} and P_{char_cnn} are of type Positive/Negative, while on $P_{senticnet}$ the inconsistency degree of Positive/Negative inconsistencies is 0.038, 0.043 on P_{rec_nn}, and 0.038 on $P_{sentiwordnet}$. We observe that the inconsistencies of type (Negative/Positive) are frequent in learning-based tools compared to lexicon-based tools. We also note that lexicon-based methods predict the polarity of an important proportion of documents as Neutral: 0.28 in P_{vader} and 0.16 in $P_{sentiwordnet}$ compared to other tools because those methods require the presence of a polar word in the document to classify it as Positive or negative, which is not the case of real data.

Summary: This experiment shows that we have several types of intra-tool inconsistencies including Negative/Positive.

5.2.4 Inter-tool Inconsistency Type. In this experiment, we categorize the prediction results of the sentiment tools on documents according to polarity. The inter-tool consistency is represented with a matrix $M \in R^{3\times3}$ where M_{ij} is the proportion of tools that attribute respectively the polarities i and j to the document d formally: $M_{ij} = \frac{card(S)}{m}$ with m the size of Γ such that for a document $d_z \in A$, a polarity function $P_{t_k} \in \Gamma$, $S = \{P_{t_{k'}} \in \Gamma | P_{t_k}(d_z) = i \wedge P_{t'_k}(d_z) = j\}$.

The experiment results are displayed in Figure 4. We observe that most inconsistencies are of (Positive/Negative) between all tools.

5.2.5 Discussion. In the above experiments, we have demonstrated that intra-tool inconsistency is a frequent anomaly in sentiment analysis tools, in particular for machine learning-based methods. After refining our analysis by categorizing inconsistencies by type, we found less intra-tool inconsistencies on methods that use a word dictionary. These methods require the presence of a polar word in the document to classify it as positive or negative. Otherwise, they consider the review as neutral, contrary to machine learning and lexicon-based methods with a concept dictionary. We notice most intra-tool inconsistencies on the *FGD*, *US airlines*, and *SST* datasets, and the most inter-tool inconsistencies on the *News* datasets. We explain this by the presence of abbreviations and syntax errors on the *FGD* and *US airlines* datasets. We analyzed the datasets *News* and *SST* to unveil the causes of inconsistencies, and we found that in the *News* dataset, the documents are meaningful and short in length. Hence, it is difficult for tools to extract the sentiment features present in the review. While on *SST*, we notice the presence of semantically equivalent items with different polarities. Table 4 represents a snapshot of inconsistencies on *SST*. We see that the documents with identifiers 43060 and 5 express the same information. However, there are missing punctuation (. and ') that do not affect the document's polarity, and the movies' title (The Cockettes) is written without any uppercase letters, which generally does not affect the polarity. Certainly, uppercase words may embed sentiment as it is mentioned in [29], but it impacts the strength of the sentiment and does not switch the polarity from Positive to Negative.

[6]The notation "(Neutral/Positive)" refers to two analogical documents with polarities Neutral and Positive, respectively.

Table 4: Example of inconsistencies on the *SST* dataset

Id	Document	Score	Polarity
43060	' (The Cockettes) provides a window into a subculture hell-bent on expressing itself in every way imaginable. '	0.625	Positive
5	' (the cockettes) provides a window into a subculture hell-bent on expressing itself in every way imaginable	0.375	Negative
179516	's a conundrum not worth solving	0.38889	Negative
179517	's a conundrum not worth solving	0.5	Neutral

5.3 Structural Analysis

In this section, we study the impact of analogical set structures on the intra-tool inconsistency by checking if inconsistencies are due to the semantic or syntactic distance between documents. In other words, checking whether the inconsistencies depend on the *Cos_Sim* as a measure to capture the syntax difference between sentences or on the *WMD_Sim* as a measure to capture their semantic. We calculate the Cos similarity between the bag-of-words representations of the documents. The Cos similarity measures the

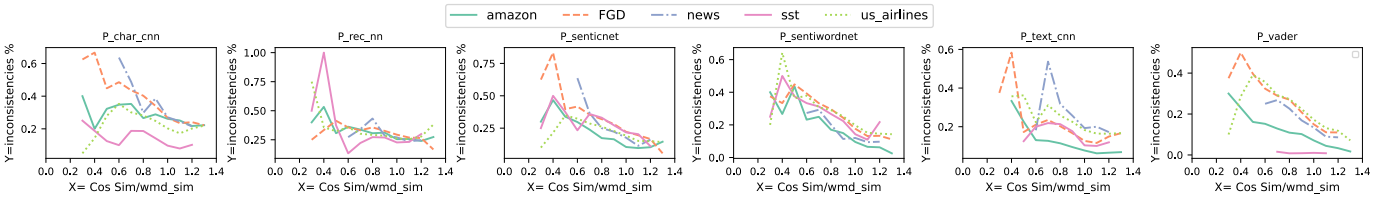

Figure 5: Inconsistencies and similarity

Table 5: Accuracy and inconsistency

Tools	*Amazon*			*News*			*SST*			*FGD*			*Usairlines*			Mean		
	Acc	Intra-tool inc	Inter-tool inc	Acc	Intra-tool inc	inter-tool inc	Acc	intra-tool inc	Inter-tool inc	Acc	Intra-tool inc	Inter-tool inc	Acc	Intra-tool inc	Inter-tool inc	Mean Acc	Mean Intra-tool inc	Mean Inter-tool inc
P_{char_cnn}	46.6%	0.18	0.54	47.9%	0.197	0.608	76.14%	0.078	0.43	47.4%	0.24	0.56	62.4%	0.175	0.538	56.1 %	0.17	0.535
P_{rec_nn}	52.5%	0.155	0.45	40.52%	0.188	0.649	65.08%	0.175	0.46	54.6%	0.22	0.538	64.4%	0.2	0.513	55.42%	0.188	0.522
$P_{senticnet}$	59.5%	0.077	0.48	52.67%	0.132	0.607	45.84%	0.164	0.48	31.15%	0.18	0.64	24.9%	0.15	0.67	42.8%	0.14	0.575
$P_{sentiwordnet}$	41.98%	0.07	0.52	33.01%	0.102	0.636	48.64%	0.137	0.46	41.69%	0.159	0.57	45.85%	0.17	0.55	42.23%	0.104	0.547
P_{text_cnn}	84.02%	0.03	0.47	49.04%	0.167	0.647	46.3%	0.09	0.48	70.01%	0.13	0.547	64.11%	0.15	0.51	62.696%	0.113	0.53
P_{vader}	56.5%	0.04	0.44	45.5%	0.1	0.6	51.4%	0.124	0.46	43.8%	0.14	0.55	49.9%	0.15	0.52	49.42%	0.109	0.514
MV	67.22%			**56.3%**			61.4%			53.8%			60%			N/A		

syntax similarity between the documents, since we considered the synonyms and abbreviations as different words. The WMD based similarity is calculated between the Word2Vec representations of the words in the documents, which allows to handle semantics. In this experiment, we verify the relation between the syntactic difference measured by Cos_Sim, the semantic difference measured by WMD_Sim, and the inconsistencies between every two analogical documents $d_i, d_j \in A$.

For this, we first represent the documents as a bag-of-words such that $d_i = [v_1, \ldots, v_n]$, with n being the vocabulary size and v_w the occurrence of the word w in the document d_i. Then, we calculate $Cos\_sim = (d_i, d_j)$ between each two documents d_i and d_j of A. We calculate also the WMD between the word2vec representation of the two documents d_i, $d_j \in A$ and we convert it to a similarity score by using $WMD\_Sim = \frac{1}{1+WMD(d_i,d_j)}$. After that, we calculate the ratio $\delta = \frac{Cos\_Sim(d_i,d_j)}{WMD\_Sim(d_i,d_j}$ which is high for large values of Cos_Sim and low values of WMD_Sim, and low for large values of WMD_Sim and low values of Cos_Sim. The percentage of inconsistent documents is calculated as: $\forall d_i, d_j \in A, P_{t_k} \in \Gamma \; \mathbb{1}_{(P_{t_k}(d_i) \neq Pt_k(d_j))}$. We categorize the results by δ values (X-axis) and calculate the proportion of inconsistencies (adversarial examples) to the total number of elements that have $delta$ (Y-axis). The results are displayed in Figure 5. We notice that we have more inconsistencies for a low δ: a high WMD_sim and low Cos_sim on all tools for all datasets comparing to the inconsistencies that we have for a high δ i.e, a low WMD_sim and high Cos_sim, which indicates that sentiment analysis tools are much influenced by the syntactic variation of the sentence.

Summary: We observed more inconsistencies for high values of WMD_sim and low values of Cos_Sim on all tools.

5.3.1 Discussion. In the above experiment, we have evaluated the relationship between analogical set structure, document structure, and inconsistency. The results show that most tools have a large proportion of inconsistencies between documents that have a high semantic similarity degree and a low syntactic similarity degree. We learn from these experiments that the structure of the sentence affects the tool vulnerability for inconsistencies. According to [31], the document structure may embed a sentiment, but not each modification in the structure implies a sentiment variation as shown in the examples of Table 4. It is crucial to consider this point when developing tools since most sentiment analysis applications are in uncontrolled environments like social media and review websites. In an uncontrolled environment, there is a lot of unintentional documents restructuring that does not imply a polarity switching.

5.4 Semantic Analysis

In this section, we answer the following questions: (1) Are inconsistencies frequent between polar facts or opinionated documents? (2) Do fewer inconsistencies imply higher accuracy? In other words, we evaluate the efficiency of tools in terms of accuracy and inconsistency.

5.4.1 Inconsistencies and Polar Facts. In this experiment, we study whether inconsistencies occur only between polar facts (i.e., the documents that do not include a polar word, such as: "this product does its job"), or even between opinionated documents (i.e., documents that contain polar words such as: "yeah! this product does its job"). For this, we first extract two sub-datasets from the original ones by filtering polar facts from opinionated documents using the word-lexicon-based method $P_{\text{sentiwordnet}}$. We first eliminate objective documents based on the ground truth, i.e. documents with Neutral polarity, then we consider the documents classified as Positive/Negative opinionated, and the misclassified Positive and Negative documents as Neutral polar facts. Then, we calculate the proportion of intra-tool inconsistencies for different tools on the opinionated/fact sub-datasets (Y-axis). The results are displayed in Figure 6, where sub_sub represents the proportion of inconsistencies between opinionated documents, $fact\_fact$ represents the proportion of inconsistencies between polar facts, and sub_fact the proportion of inconsistencies between polar facts and opinionated documents. We notice that the large proportion of inconsistencies is between polar fact and opinionated documents in all tools/datasets. However, we have no inconsistencies between polar facts in lexicon-based methods because those tools classify polar facts as Neutral most of the time.

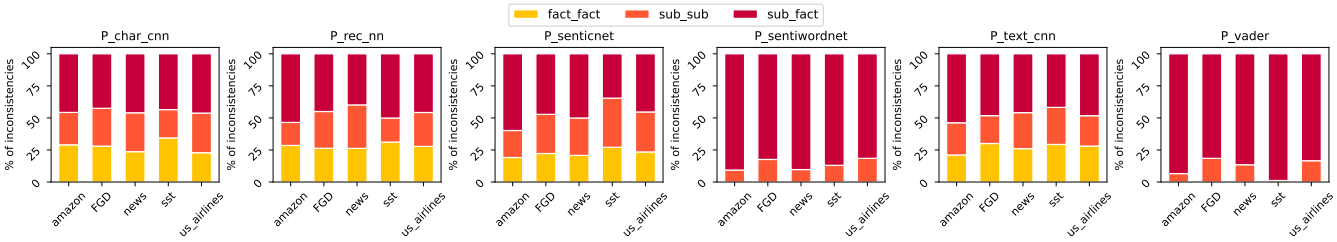

Figure 6: Inconsistencies and polar fact

Table 6: Accuracy after resolving inconsistencie using MV

	P_{text_cnn}	P_{char_cnn}	$P_{senticnet}$	$P_{sentiwordnet}$	P_{rec_nn}	P_{vader}	MV
Amazon	**84.4%**	**49.26%**	**60.7%**	**42.8%**	**54.5%**	**57.3%**	66.7%
News	**50.6%**	**51.4%**	**53.67%**	**33.48%**	**47.4%**	**47.2%**	**57.4%**
SST	45.9%	**77.3%**	**47.9%**	**49%**	**66%**	**51.5%**	63.4%
FGD	49.2%	**57.5%**	**30.9%**	**42.07%**	**71.05%**	**44.1%**	58.7%
US airlines	64.6%	**68.1%**	24.5%	**46.11**	**65.4%**	**50.3%**	65.5%

For other methods, we notice a difference in the proportion of inconsistencies. For instance, on P_{char_cnn}, $recn\_nn$, and $text\_cnn$, we have more inconsistencies between polar facts than between opinionated documents with proportions of 29.8%, 29.6%, and 32.2%, respectively. On $P_{senticnet}$, we have more inconsistencies between opinionated documents with a proportion of 27.3%.

Summary: This experiment shows that most inconsistencies occur between polar facts and opinionated documents, which motivates us to do more experiments and verify which of them is likewise closer to the ground truth.

5.4.2 Inconsistencies and Accuracy. In this section, we show the relationship between inconsistency and accuracy in different tools. For this, we calculate on each dataset the accuracy, and the mean intra-tool and inter-tool inconsistency rate following Equation 7. The results are summarised in Table 5. We observe a high accuracy on the machine learning-based tools P_{char_cnn}, P_{text_cnn} and P_{rec_nn} (mean accuracy of 56.1%, 62.7696% and 55.42% respectively) compared to the lexicon based methods where we notice a mean accuracy of (42.23% on $P_{sentiwordnet}$, 49.42% on P_{vader}, and 42.8% $P_{senticnet}$. We notice that alow intra-tool inconsistency does not imply good accuracy since we have a low intra-tool inconsistency on lexicon based methods and a low accuracy. We observe an inverse relation between inter-tool inconsistency and accuracy which means that a high inter-tool inconsistency is a sign of low accuracy.

5.4.3 Hyper-parameters and Inconsistency. In this section, we focus on studying the impact of the learning hyper-parameters on the intra-tool inconsistency and the accuracy of the CNN. For that, we consider the basic CNN described in [37], that we train each time by varying one of the hyper-parameters: the training dataset, the embedding type, the batch size, the cross validation folds, the learning rate, the dropout probability and the number of training epochs.

We answer the following questions: (1) Is intra-tool inconsistency present for all CNN configurations? (2) What is the best configuration that respects both accuracy and consistency? (3) Which training dataset guarantees a general and accurate model?

Inconsistency and training dataset. In this experiment, we evaluate the robustness of the learned model based on the training dataset. We trained the model each time on the original, not extended, datasets (*Amazon*, *News* , *SST*, *FGD*, and *US airlines*) and tested the model on our benchmark. This experiment allows us to verify the generalization of the model on the extended dataset and the other datasets. The results are presented in Figure 7 (Inc & training dataset) and Figure 8 (Acc & training dataset). We observe a mean accuracy of 50% on the models trained with the Amazon dataset (92.6% on *Amazon*, 56% on *News*, 47% on *SSt*, 18% on *FGD* and 20% *US airlines* datasets), 55.4% on the model trained with the *FGD* dataset, 52.8% on the models trained with the *News* dataset, 58.2% on the model trained with *SST* and 54.4% on the model trained with *US airlines*. For the inconsistency rate, the *Amazon* dataset has a mean inconsistency of 0.066, while The *FGD*, *News*, *SST* and *US airlines* -trained model achieve 0.124, 0.186, 0.15, and 0.1, respectively. We notice that the model trained with amazon is more general, since it leads to good accuracy and inconsistency when applied on other datasets (i.e., news and reviews), but not on the tweets datasets, which contain many abbreviations and errors, and need models specifically trained for them.

Inconsistency and embedding type. In this experiment, we check the robustness of the model with different embedding types: a pre-trained version of BERT as an embedding layer of the CNN, a word embedding using google news vectors, a word embedding using glove and an embedding combining both words and characters. The results of this experiment are presented in Figure 7 (Inc & embedding type) and 8(Acc & embedding type). We notice a mean accuracy of 58.7% on the model with the BERT embedding, a mean accuracy of 66.6% on the model with both word and character embeddings, a mean accuracy of 53.4% on glove and a mean accuracy of 64.1%% on the model with google news vectors word embedding. We notice a high accuracy on social media data (*FGD* and *US airlines* datasets) on models that use both character embedding and word2vec embedding. We notice also a mean inconsistency degree of 0.117 on BERT, 0.158 on models with both character and word embeddings. On the models with a word embedding layer that uses glove, we notice a mean inconsistency of 0.104.

We notice that the model using the pre-trained bert as embedding is not adapted to twitter data. Unsurprisingly, the use of the pre-trained BERT for embedding has improved the model's generalization on the *News* and *Amazon* datasets. We can explain this generalization by the context-dependent nature of BERT that allows us to generate different embeddings of the word depending on the context. The best inconsistency score (the lowest) was obtained on the model that uses the pre-trained Glove embeddings.

Inconsistency and batch size: In this experiment, we train the model described in [37] on the *News* dataset with a batch size of

: 50, 100, 150, 200, 300, and 500. On the *News* dataset, we observe that the accuracy decreases when the batch size increases, (from an accuracy of 0.756 for a batch size of size 50 to an accuracy of 0.724 for a batch size of 500). On the *Amazon* dataset, the accuracy increases with the batch size (accuracy of 0.56 on the model with batch size of 50 to an accuracy of 0.68 for a batch size of 500). On the *SST* dataset, the accuracy slightly drops from 0.455 to 0.433. We also observe that, contrary to the other datasets, inconsistency increases on the *News* dataset.

The optimal batch size is 300. A smaller batch size does not help the model generalize well as good inconsistency and accuracy results are only observed for the training dataset.

Inconsistency and cross validation. In this experiment, we evaluate the impact of the n-cross validation on both the inconsistency and accuracy. We trained the model [37] on the *News* dataset and we varied n using the values: 2, 5, 10 and 20. We observe that the accuracy increases with the cross validation size (n) on all datasets, while the inconsistency decreases with n on the *News* and *Amazon* datasets.

Cross validation increases the accuracy, consistency and generalization of the model. We recommend using 10-cross validation to optimize the consistency/accuracy trade-off.

Inconsistency and dropout probability. In this experiment, we verify the impact of the dropout probability on the accuracy, inconsistency and generalization of the model. To achieve this, we train our model and vary the dropout probability following the values: 0.2, 0.3 and 0.5. The results are presented in Figures 7 (Inc & dropout p) and 8 (Acc & dropout p). We notice that the inconsistency on the datasets decreases when the dropout probability increases. The accuracy increases with the dropout probability on all datasets except the *Amazon* dataset.

Inconsistency and learning rate. To evaluate the impact of the learning rate on inconsistency and accuracy, we train our model on the *News* dataset, and we vary the learning rate values following the values: $1e^{-4}$, $1.5e^{-4}$, $2e^{-4}$, and $3e^{-4}$. The results are presented in Figures 7 (Inc & learning rate) and 8 (Acc & learning rate). We observe that when the learning rate increases, the inconsistency decreases on the *News* dataset and increases on the others datasets. We also observe that the accuracy increases with learning rate on the datasets *News*, *FGD*, and *SST* and decreases on the datasets *Amazon* and *US airlines* datasets. The learning rate affects both accuracy and inconsistency since models with a low learning rate are more general and consistent.

The number of epochs. In this experiment, we verify the influence of the number of epochs on the inconsistency and the accuracy. We train the model on numbers of epochs following the values: 50, 100, 200, 300 and 3000, then test the model using our benchmark. The results in Figure 8 (Acc & nb_epochs) show that the accuracy of the model on the datasets increases then decreases as the number of epochs increase, which means that the model lost its generalization when trained with a high number of epochs (overfitting). As for the inconsistency, Figure 7 (Inc & nb_epochs) shows that it first decreases then increases which confirms the overfitting explanation. Even if training the model on a high number of epochs may improve the training accuracy of the model, this makes it lose in terms of intra-tool consistency and generalization.

5.4.4 Discussion. From the findings of the previous experiments, we observe that the inconsistencies are present on different *CNN* configurations. We also notice that inconsistencies depend on the subjectivity of the document and most inconsistencies occur between polar facts and opinionated documents. To verify which of the two types is more close to the ground truth, we calculate the accuracy in the sub-datasets of opinionated documents and facts. We found that polar facts are more accurate on P_{char_cnn} with an accuracy of 62% for polar facts and 57.2% for opinionated documents, while in other tools we notice a higher accuracy on opinionated data (74.5%, 68.4%, 73.5% for opinionated data and 55.7%, 56.1% and 64.4% for polar facts on P_{rec_nn}, $P_{senticnet}$ and P_{text_cnn}, respectively). In this case, we can use the document nature as a feature when resolving inconsistency. We also learned that there is an inverse correlation between inter-tool inconsistency and accuracy. Therefore, the most accurate tools are consistent.

These findings motivate us to study the impact of resolving inconsistency on accuracy. We display in Table 6 the accuracy of tools after resolving intra-tool inconsistency on different tools using majority voting (MV) and both intra-tool and inter-tool inconsistency (the Column MV). We observe an important improvement of the accuracy when resolving intra-tool inconsistency. This may be explained by the fact that the tools prediction may be erroneous on some documents and correct on others. Hence, resolving intra-tool inconsistency by unifying the polarity in the analogical set may increase the accuracy in most cases.

We observe an accuracy degradation when resolving inconsistencies on the *SST*, *FGD*, and *Amazon* datasets, because the tools are not compatible in terms of accuracy on these datasets. To overcome this problem and guarantee the accuracy improvement, we recommend to apply majority voting between tools compatible in terms of accuracy. Few works have exploited the inconsistency to improve the accuracy of classification, such as the work in [60], where the authors minimize inter-tool inconsistency of various labeling functions based on the accuracy, correlation and label absence. These solutions outperform majority voting only for a low setting of redundancy (i.e., when the number of tools is low), which confirms that the inconsistency problem is not yet resolved. However, these works do not consider both intra-tool and inter-tool inconsistencies, which would lead to better results. Since the inconsistency problem has been wildly studied in data management [5, 6, 20, 58] and the fact inference field [24, 40, 81], where has been made tangible progress, we suggest to refer to methods from fact inference by considering workers as tools in the formalization to resolve inconsistency and improve accuracy [24, 40].

5.4.5 Scalability experiments. In this experiment, we evaluate the scalability of tools with respect to dataset and document size.

Figures 9(a-b) show execution time when we vary the data size: 25%, 50%, 75%, and 100% of the full benchmark. (Figure 9(b) focuses on the performance of P_{char_cnn} and P_{text_cnn} only.) We observe, that for lexicon-based methods and P_{rec-nn}, the time scales (almost) linearly with the size. Machine learning methods P_{text_cnn} and P_{char_cnn} are faster than the other tools. This is explained by the fact that these tools perform prediction in parallel (prediction by batch), which accelerates the prediction process.

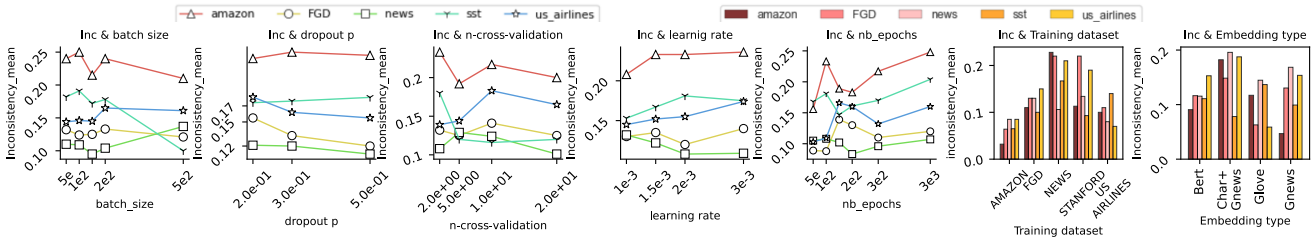

Figure 7: Inconsistency and learning hyper-parameters

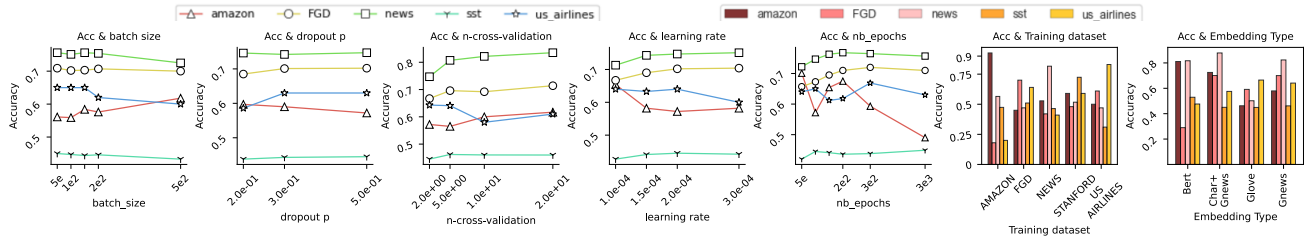

Figure 8: Accuracy and learning hyper parameters

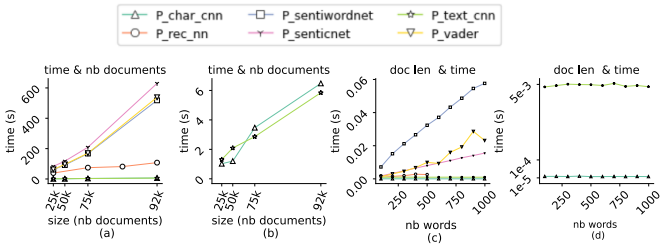

Figure 9: Scalability

We also evaluate the scalability of tools in terms of response time when varying the document size (number of words), using a synthetic dataset sampled from the word corpus of nltk.

The results are presented in Figures 9(c-d), where the x-axis is document size expressed as number of words and y-axis is execution time. (Figure 9(b) focuses on the performance of P_{char_cnn} and P_{text_cnn} only.) We observe that execution time scales linearly with the document size for all methods except for the machine learning ones: these methods use a fixed-size embedding, and they then crop or pad the text (if it is too long or short, respectively), which makes them inaccurate for long text.

6 CONCLUSIONS

This paper presents the first large study on data quality for sentiment analysis tools regarding the quality of polarity extraction from documents across different domains. Our study offers several findings that are of interest to the machine learning, NLP , and data management communities.

We have built a benchmark, which we make available, along with our code, for reproducibility and future research. We provide statistical, structural, and semantic analysis for the inconsistency problem, show that the inconsistency problem is not yet resolved, and argue for the potential for improvement that can be obtained on accuracy when resolving intra-tool and inter-tool inconsistencies.

Based on our analysis and findings, we present the following recommendations, which tool to use for which data type:

- For short text, we recommend CNN based methods. Short texts have few sentiment features, which can not be effectively handled by word-lexicon or concept based methods.
- From the explainability point of view, we recommend lexicon based methods, especially those based on concept lexicons, thanks to their performance in terms of accuracy.
- For streaming data with long text and a large window, we recommend an ensemble of lexicon-based methods compatible in terms of accuracy for inter-tool inconsistency. For small window sizes (i.e., number of documents < 1000 per δt), we recommend CNN.
- For social media data, both character and word embedding are recommended, since we can train the model to be fault tolerant. For reviews, word embedding is enough and for news and factual data, we recommend BERT embedding.
- Inconsistencies are influenced by the analogical set structure and the document nature (fact or opinionated).

Our insights have implications to various use cases, such as crowdsourced fact checking, where we can accurately classify workers, remove inconsistent labels, or alleviate biased labeling.

- The scalability experiments point to interesting research opportunities for the data management community in order to improve the scalability properties of existing solutions to better serve the machine learning and NLP communities.

To the best of our knowledge, no previous research work has explored the problem of resolving both intra-tool and inter-tool inconsistencies. The results of this study could be used to develop more accurate frameworks in this area.

ACKNOWLEDGMENTS

We thank Khodor Hamoud for his support, and the reviewers for the numerous constructive comments. The work is supported partially by IMBA Consulting and ANRT French program through the grant nr.2018/0576 e-reputation.

REFERENCES

[1] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*, 2018.

[2] S. Amer-Yahia, T. Palpanas, M. Tsytsarau, S. Kleisarchaki, A. Douzal, and V. Christophides. Temporal analytics in social media. In *Encyclopedia of Database Systems, Second Edition*. Springer, 2018.

[3] M. Balduini, E. D. Valle, D. Dell'Aglio, M. Tsytsarau, T. Palpanas, and C. Confalonieri. Social listening of city scale events using the streaming linked data framework. In *ISWC*, 2013.

[4] M. Bautin, L. Vijayarenu, and S. Skiena. International sentiment analysis for news and blogs. In *ICWSM*, 2008.

[5] S. Benbernou and M. Ouziri. Enhancing data quality by cleaning inconsistent big RDF data. In *2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017*, pages 74–79, 2017.

[6] L. E. Bertossi. Inconsistent databases. In *Encyclopedia of Database Systems, Second Edition*. Springer, 2018.

[7] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447, 2007.

[8] E. Cambria, S. Poria, D. Hazarika, and K. Kwok. Senticnet 5: discovering conceptual primitives for sentiment analysis by means of context embeddings. In *Proceedings of AAAI*, 2018.

[9] N. Carlini and D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM, 2017.

[10] B. Chen and X. Zhu. Bilingual sentiment consistency for statistical machine translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 607–615, 2014.

[11] T. Chen, R. Xu, Y. He, and X. Wang. Improving sentiment analysis via sentence type classification using bilstm-crf and cnn. *Expert Systems with Applications*, 72:221–230, 2017.

[12] Y. Choi and J. Wiebe. +/-effectwordnet: Sense-level lexicon acquisition for opinion inference. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1181–1191, 2014.

[13] Y. Choi, J. Wiebe, and R. Mihalcea. Coarse-grained+/-effect word sense disambiguation for implicit sentiment analysis. *IEEE Transactions on Affective Computing*, 8(4):471–479, 2017.

[14] K. Cortis, A. Freitas, T. Daudert, M. Huerlimann, M. Zarrouk, S. Handschuh, and B. Davis. Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 519–535, 2017.

[15] N. N. Dalvi, A. Machanavajjhala, and B. Pang. An analysis of structured data on the web. *Proc. VLDB Endow.*, 5(7):680–691, 2012.

[16] G. Demartini and S. Siersdorfer. Dear search engine: what's your opinion about...?: sentiment analysis for semantic enrichment of web search results. In *Proceedings of the 3rd International Semantic Search Workshop*, page 4. ACM, 2010.

[17] K. Denecke. Using sentiwordnet for multilingual sentiment analysis. In *2008 IEEE 24th International Conference on Data Engineering Workshop*, pages 507–512. IEEE, 2008.

[18] H. Ding and E. Riloff. Acquiring knowledge of affective events from blogs using label propagation. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[19] H. Ding and E. Riloff. Weakly supervised induction of affective events by optimizing semantic consistency. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[20] X. L. Dong and D. Srivastava. Entity resolution. In *Encyclopedia of Database Systems, Second Edition*. Springer, 2018.

[21] C. Dos Santos and M. Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, 2014.

[22] M. Dragoni and G. Petrucci. A fuzzy-based strategy for multi-domain sentiment analysis. *International Journal of Approximate Reasoning*, 93:59–73, 2018.

[23] E. C. Dragut, H. Wang, P. Sistla, C. Yu, and W. Meng. Polarity consistency checking for domain independent sentiment dictionaries. *IEEE Transactions on knowledge and data engineering*, 27(3):838–851, 2015.

[24] A. Drutsa, V. Fedorova, D. Ustalov, O. Megorskaya, E. Zerminova, and D. Baidakova. Crowdsourcing practice for efficient data labeling: Aggregation, incremental relabeling, and pricing. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 2623–2627, 2020.

[25] A. Esuli and F. Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC*, volume 6, pages 417–422. Citeseer, 2006.

[26] R. Feldman. Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89, 2013.

[27] X. Feng, Y. Zeng, and Y. Xu. Recommendation algorithm for federated user reviews and item reviews. In *Proceedings of the 2018 International Conference on Artificial Intelligence and Virtual Reality*, pages 97–103. ACM, 2018.

[28] G. Fu, Y. He, J. Song, and C. Wang. Improving chinese sentence polarity classification via opinion paraphrasing. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 35–42, 2014.

[29] C. H. E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*, 2014.

[30] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

[31] S. Greene and P. Resnik. More than words: Syntactic packaging and implicit sentiment. In *Proceedings of human language technologies: The 2009 annual conference of the north american chapter of the association for computational linguistics*, pages 503–511. Association for Computational Linguistics, 2009.

[32] H. Hamdan, F. Béchet, and P. Bellot. Experiments with dbpedia, wordnet and sentiwordnet as resources for sentiment analysis in micro-blogging. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 455–459, 2013.

[33] R. He and J. McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517. International World Wide Web Conferences Steering Committee, 2016.

[34] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1681–1691, 2015.

[35] M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, 2018.

[36] X. Ji. Social data integration and analytics for health intelligence. In *Proceedings VLDB PhD Workshop*, 2014.

[37] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[38] F. Kokkinos and A. Potamianos. Structural attention neural networks for improved sentiment analysis. *arXiv preprint arXiv:1701.01811*, 2017.

[39] E. Kouloumpis, T. Wilson, and J. D. Moore. Twitter sentiment analysis: The good the bad and the omg! *Icwsm*, 11(538-541):164, 2011.

[40] E. Krivosheev, S. Bykau, F. Casati, and S. Prabhakar. Detecting and preventing confused labels in crowdsourced data. *Proceedings of the VLDB Endowment*, 13(12):2522–2535, 2020.

[41] F. M. Kundi, S. Ahmad, A. Khan, and M. Z. Asghar. Detection and scoring of internet slangs for sentiment analysis using sentiwordnet. *Life Science Journal*, 11(9):66–72, 2014.

[42] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966, 2015.

[43] S. Lai, L. Xu, K. Liu, and J. Zhao. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[44] B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*, 2017.

[45] B. Liu and L. Zhang. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer, 2012.

[46] L. Luo, X. Ao, F. Pan, J. Wang, T. Zhao, N. Yu, and Q. He. Beyond polarity: Interpretable financial sentiment analysis with hierarchical query-driven attention. In *IJCAI*, pages 4244–4250, 2018.

[47] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

[48] T. Mahler, W. Cheung, M. Elsner, D. King, M.-C. de Marneffe, C. Shain, S. Stevens-Guille, and M. White. Breaking nlp: Using morphosyntax, semantics, pragmatics and world knowledge to fool sentiment analysis systems. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 33–39, 2017.

[49] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller. Tweets as data: demonstration of tweeql and twitinfo. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 1259–1262, 2011.

[50] J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM, 2013.

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

[52] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[53] T. Miyato, A. M. Dai, and I. Goodfellow. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*, 2016.

[54] B. Ohana and B. Tierney. Sentiment classification of reviews using sentiwordnet. In *9th. it & t conference*, volume 13, pages 18–30, 2009.

[55] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[56] A.-M. Popescu and M. Pennacchiotti. Detecting controversial events from twitter. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1873–1876, 2010.

[57] S. Poria, E. Cambria, and A. Gelbukh. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2539–2544, 2015.

[58] N. Prokoshyna, J. Szlichta, F. Chiang, R. J. Miller, and D. Srivastava. Combining quantitative and logical data cleaning. *Proc. VLDB Endow.*, 9(4):300–311, 2015.

[59] C. Quirk, C. Brockett, and W. B. Dolan. Monolingual machine translation for paraphrase generation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 142–149, 2004.

[60] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré. Snorkel: Rapid training data creation with weak supervision. *The VLDB Journal*, 29(2):709–730, 2020.

[61] M. T. Ribeiro, S. Singh, and C. Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, 2018.

[62] J. Risch and R. Krestel. Aggression identification using deep learning and data augmentation. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 150–158, 2018.

[63] H. Rong, V. S. Sheng, T. Ma, Y. Zhou, and M. A. Al-Rodhaan. A self-play and sentiment-emphasized comment integration framework based on deep q-learning in a crowdsourcing scenario. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[64] K. Schouten and F. Frasincar. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830, 2015.

[65] A. Severyn and A. Moschitti. Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 959–962. ACM, 2015.

[66] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[67] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou. Sentiment embeddings with applications to sentiment analysis. *IEEE transactions on knowledge and data Engineering*, 28(2):496–509, 2015.

[68] M. Tsytsarau, S. Amer-Yahia, and T. Palpanas. Efficient sentiment correlation for large-scale demographics. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 253–264, 2013.

[69] M. Tsytsarau and T. Palpanas. Survey on mining subjective data on the web. *Data Min. Knowl. Discov.*, 24(3):478–514, 2012.

[70] M. Tsytsarau and T. Palpanas. Managing diverse sentiments at large scale. *IEEE Transactions on Knowledge and Data Engineering*, 28(11):3028–3040, 2016.

[71] M. Tsytsarau, T. Palpanas, and M. Castellanos. Dynamics of news events and social media reaction. In *KDD*, 2014.

[72] S. Vosoughi, P. Vijayaraghavan, and D. Roy. Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1041–1044. ACM, 2016.

[73] K. Wang and X. Wan. Sentigan: Generating sentimental texts via mixture adversarial networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 4446–4452, 2018.

[74] W. Wang, J. Gao, M. Zhang, S. Wang, G. Chen, T. K. Ng, B. C. Ooi, J. Shao, and M. Reyad. Rafiki: Machine learning as an analytics service system. *Proc. VLDB Endow.*, 12(2):128–140, 2018.

[75] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang. Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1031–1040, 2011.

[76] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu. Sentiment analysis by capsules. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1165–1174. International World Wide Web Conferences Steering Committee, 2018.

[77] J. W. Wei and K. Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

[78] J. Wiebe, T. Wilson, and C. Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210, 2005.

[79] B. Yang and C. Cardie. Context-aware learning for sentence-level sentiment analysis with posterior regularization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 325–335, 2014.

[80] K. Zhao, Y. Liu, Q. Yuan, L. Chen, Z. Chen, and G. Cong. Towards personalized maps: Mining user preferences from geo-textual data. *Proc. VLDB Endow.*, 9(13):1545–1548, 2016.

[81] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment*, 10(5):541–552, 2017.

[82] L. Zhu, A. Galstyan, J. Cheng, and K. Lerman. Tripartite graph clustering for dynamic sentiment analysis on social media. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1531–1542, 2014.