Tampere University

Nannan Zou

TEXT ANALYTICS METHODS FOR SENTENCE-LEVEL SENTIMENT ANALYSIS

Faculty of Information Technology and Communication Sciences
Master of Science thesis
May 2019

ABSTRACT

Nannan Zou: Text Analytics Methods for Sentence-level Sentiment Analysis
Master of Science thesis
Tampere University
Degree Programme in Electrical Engineering
May 2019

Opinions have important effects on the process of decision making. With the explosion of text information on networks, sentiment analysis, which aims at predicting the opinions of people about specific entities, has become a popular tool to make sense of countless text information. There are multiple approaches for sentence-level sentiment analysis, including machine-learning methods and lexicon-based methods. In this MSc thesis we studied two typical sentiment analysis techniques – AFINN and RNTN, which are also the representation of lexicon-based and machine-learning methods, respectively.

The assumption of a lexicon-based method is that the sum of sentiment orientation of each word or phrase predicts the contextual sentiment polarity. AFINN is a word list with sentiment strength ranging from -5 to $+5$, which is constructed with the inclusion of Internet slang and obscene words. With AFINN, we extract sentiment words from sentences and sentiment scores are then assigned to these words. The sentiment of a sentence is aggregated as the sum of scores from all its words.

The Stanford Sentiment Treebank is a corpus with labeled parse trees, which provides the community with the possibility to train compositional models based on supervised machine learning techniques. The labels of Stanford Sentiment Treebank involve 5 categories: negative, somewhat negative, neutral, somewhat positive and positive. Compared to the standard recursive neural network (RNN) and Matrix-Vector RNN, Recursive Neural Tensor Network (RNTN) is a more powerful composition model to compute compositional vector representations for input sentences. Dependent on the Stanford Sentiment Treebank, RNTN can predict the sentiment of input sentences by its computed vector representations.

With the benchmark datasets that cover diverse data sources, we carry out a thorough comparison between AFINN and RNTN. Our results highlight that although RNTN is much more complicated than AFINN, the performance of RNTN is not better than that of AFINN. To some extent, AFINN is more simple, more generic and takes less computation resources than RNTN in sentiment analysis.

Keywords: text mining, sentiment analysis, machine-learning methods, lexicon-based methods, AFINN, RNTN, performance assessment, hypothesis test, bootstrap, cross-validation

PREFACE

First, lots of thanks to my supervisor, Prof. Frank Emmert-Streib. During the thesis process, he created the topic, provided assistance and feedback to my research and writing. From him, I learned how to start the research and a serious attitude to my work. He is so patient and he is also one of the best professors in my life.

Second, thanks to the authors of the LaTeX community and R packages. With their work, I can carry out my experiments successfully and present a beautiful thesis format.

Third, thanks to the authors who provided the original datasets that are used in our experiments and the computational resources provided by Tampere University.

Fourth, I would like to thank my family for their encouragements to take on this study and putting up with me for the past four years.

And last but not least, I am grateful to my friend, Han Feng. During my thesis writing, I got into trouble with my life and sometimes I almost gave up the thesis. It was him who was always encouraging me to believe that the life would become better and I could make it. I also would like to thank my friend Shubo Yan for his encouragements, dumplings and help for printing. A big thank you to my favorite singer Chenyu Hua whose songs accompanied me every day and night, and all friends who supported and comforted me during this period.

Tampere, 9th May 2019

Nannan Zou

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

A	accuracy
AFINN	a lexicon-based method for sentiment analysis
AI	artificial intelligence
AMT	Amazon Mechanical Turk
ANEW	Affective Norms for English Words
CK	Cohen's Kappa
Dt	data set
D	data set
E	error function
$F1$	the harmonic average of precision and recall
f	NN function
FNeR	false neutral ratio
FNR	false negative ratio
FPR	false positive ratio
h	the output of a tensor product
IVR	Interactive Voice Response
kNN	k-nearest neighbor classification
λ	a weight vector in Maximum Entropy
LIWC	a dictionary-based analysis tool
LOOCV	leave-one-out cross validation
Macro-$F1$	average $F1$ scores over all classes
MaxEnt	maximum entropy
MV-RNN	matrix-vector recursive neural network
Neg	negative class
Neu	neutral class
NLP	natural nanguage processing
NYT	New York Times
p	the parent vector
$P(c)$	the prior probability of class
P_{ME}	the conditional probability in Maximum Entropy
$P(X\|c)$	the probability of predictor given class
$P(X)$	the prior probability of predictor
P	precision of a class
POS	part-of-speech
Pos	positive class
R	recall of a class
RNN	recursive neural network
RNTN	Stanford Recursive Neural Tensor Network
SE	standard error
SVM	support vector machine
TCSC	Tampere Center for Scientific Computing
TED	a media organization
TNeR	true neutral ratio

TNR	true negative ratio
TPR	true positive ratio
U	a uniform distribution
V	matrix of each tensor slice
W_s	the sentiment classification matrix
W	weight of neuron
WSD	word sense disambiguation
YTB	YouTube

1 INTRODUCTION

Opinions from ordinary people and experts always have significant influences on the process of decision making. Finding out "What others think" forms an important part for most of us to gather information. Even before the World Wide Web, we always asked friends to explain their opinions about political events, requested colleagues for recommendation letters, or consulted a shopping-guide to determine what dishwasher to purchase [47]. As the Internet and the Web became more and more widespread, it is now possible for people to search for various views. For instance, in two surveys of more than 2000 American adults, 32% have posted ratings regarding a product or service, and more than 73% report that online comments had important effects on their purchase [47]. Similarly, according to another survey of over 2500 American adults, approximate 30% have the need for political information [47]. However, the problem of overwhelming and confusing online information leads to a rapidly increasing demand for better information-understanding system.

Data science is a novel discipline that emphasizes on extracting implicit, nontrivial and potentially meaningful information from data [14]. The basic motivation behind data science is that valuable information is contained in these large databases but concealed within the mass of uninteresting data. Data science combines techniques and theories drawn from many fields like statistics, mathematics, and computer science. Primarily, predictive causal analytics, prescriptive analytics and machine learning are used to make decisions and predictions in data science [25].

Mining information from large databases has been recognized as an important topic in research, and it also provides a great opportunity of revenues for many industrial companies. The applications of extracted information consist of decision making, information management, process control, query processing and etc. Moreover, some emerging applications in information delivery services, such as online services, also use a variety of information mining approaches to improve their work [8].

Text mining is known as information extraction from textual database [27]. It is the extraction of novel and interesting knowledge from diverse written resources. However, text mining is different from what users usually do in web search [24]. In web search, the purpose is to look for something that is already aware of and has been written by someone else. The only difficulty is to select what you need and push aside all the materials that are not relevant. In text mining, user are trying to discover something that is non-trivial and so could not have been yet written down [24].

Due to such a fact, sentiment analysis, which aims at predicting the opinions of authors about specific entities, is a currently hot research area in text mining. Explosive social media sites, such as Twitter, Facebook, blogs, user forums and message boards, provide an extremely convenient way for individuals and organizations to monitor their reputation and get real-time feedback [19]. Nevertheless, since privately and publicly available information is constantly growing over Internet, it is tricky for a human reader to identify relevant sites and accurately summarize the opinions within them [36]. Furthermore, it is difficult for people to produce consistent results with a large amount of information, due to their physical and mental limitations [36]. Thus, sentiment analysis has become a popular tool to make sense of countless text information.

Nowadays, there is a considerable number of applications of sentiment analysis. Over 7,000 articles have been written about such a technique, and various startups are developing solutions and packages to analyze and monitor sentiments on social networks [19]. Many online merchants enable their customers to review the products they have purchased, and keep track customer opinions with the analysis of their reviews [28]. In finance, financial investors make use of sentiment analysis to discover public moods on the market and indicate analytical perspective [5, 43]. Another successful application is in politics, where the analysis of political sentiment closely relates to the candidates' political positions [66]. In some ways, it can reflect the election result.

Sentiment analysis can be categorized into three specific groups: document-level sentiment analysis, sentence-level sentiment analysis, and aspect-based sentiment analysis [19]. Document-level sentiment analysis is the simplest form of sentiment analysis and its assumption is that the author expresses an opinion on one major object in this document. A document can be segmented into multiple sentences; therefore, sentence-level sentiment analysis means obtaining the sentiment of an individual sentence. Nevertheless, not all the sentences contain opinions and present sentiment polarity [69]. Because objective sentences have no help to infer the polarity, only subjective sentences deserves further analyzing. In other words, sentence-level sentiment analysis is also known as polarity classification. One entity usually have numerous attributes and people often have different opinions about each of these attributes. Thus, aspect-based sentiment analysis provides the possibility to detect all the sentiments within the given entity. Due to the particular short sentences that people prefer to use on social networks, in this thesis we mainly focus on the sentence-level sentiment analysis [19].

In recent years, a huge amount of approaches have been proposed for sentence-level sentiment analysis. To achieve state-of-the-art performance, recent methods mostly adopt machine learning algorithms or lexical-based algorithms. With advances in computer technology, machine learning is an application of artificial intelligence (AI) which is undergoing intense development. It denotes methods which provide computers with the ability to learn and improve automatically without human intervention or assistance [1]. Machine learning techniques usually include two categories: supervised learning and unsupervised learning [1].

Supervised learning, where the desired outputs have been labeled, aims to learn a function that best approximates the relationship between inputs and outputs. In contrast, the goal of unsupervised learning, without labeled outputs, is to automatically identify the natural structure in data. Machine learning has been applied successfully on many fields, for instance, it has demonstrated outstanding performance in bioinformatics [4], natural language processing [10], computer vision [55], and data mining [67]. Moreover, machine learning methods are currently active and showing significant performances on sentiment analysis tasks. These techniques are completely prior-knowledge-free, and they attempt to learn an end-to-end mapping between sentences and sentiment polarities. The most common machine learning methods in sentiment analysis include Neural Networks, Naive Bayes classification, Support Vector Machines and Maximum Entropy classification [46].

However, lexicon-based methods need strong prior information. The lexicon-based methods use a predefined dictionary in which each word corresponds to a specific sentiment. The sentiment dictionary plays a key role in sentiment analysis tasks. Mostly, subjective sentences can be separated into sentiment terms (words or phrases) which convey positive or negative sentiment polarities [50]. The identification of polarities for such terms would help in better inferring sentiment of the whole sentence. Nevertheless, these approaches require diverse predefined dictionaries to adapt to varying contexts. For example, PANAS-t was proposed to analyze sentiments based on a well-established psychometric scale [23], whereas Linguistic Inquiry and Word Count (LIWC) was proposed to measure more formal psychological words [63].

Since the state-of-the-art performance has not been clearly confirmed, any popular machine learning or lexicon-based approaches are acceptable by the research community to measure sentiments. Nevertheless, we are aware of little about relative efficiency and performance of the two different approaches. In other words, many recently proposed techniques are widely deployed for developing applications without deeper comparing their efficiency in distinct contexts with each other [19]. Therefore, it is necessary to conduct a thorough comparison of machine learning and lexicon-based sentiment analysis approaches across multiple datasets.

In this thesis, we introduce various approaches for sentence-level sentiment analysis, including machine-learning methods and lexicon-based methods. Additionally, we study two typical sentiment analysis techniques: AFINN and RNTN. With the benchmark datasets that cover diverse data sources, we also carry out a thorough comparison between AFINN and RNTN.

The remainder of this thesis is organized as follows. In Chapter 2, we briefly review theoretical background. In Chapter 3, we describe the software and hardware resources, sentiment analysis techniques that we compared, datasets and performance assessment approaches. Chapter 4 summarizes our results and analysis. Finally, Chapter 5 concludes the thesis and presents direction for future work.

2 THEORETICAL BACKGROUND

In this chapter, we shall first discuss natural language processing, text mining and text encoding that forms the basic components of sentiment analysis. Furthermore, background and existing related work on sentence-level sentiment analysis are briefly reviewed.

2.1 Natural language processing

Natural Language Processing (NLP) focuses on the field that aids computers to understand and manipulate the human's natural language. NLP researchers are interested in exploring how humans understand and communicate with natural language, and then the computer systems can use these techniques to manipulate natural languages to solve target problems. NLP combines the fields of linguistics, electrical and electronic engineering, robotics, statistics, psychology, artificial intelligence, and computer sciences [9]. NLP can perform outstanding in a number of common applications, such as language translation (e.g. Google Translate), Interactive Voice Response (IVR), personal assistant (e.g. Siri), and speech recognition.

The core idea behind NLP is natural language understanding. For computers, there are three principal problems in understanding humans' natural languages: thought procedures, the representation and meaning of the linguistic input, and the world knowledge [9]. When a text has been provided, the NLP program will utilize algorithms to decide the morphological structure and nature at the word level. Then, it will try to extract meaning associated with the whole sentence and collect the essential information. Finally, the program will consider the context or the overall domain of the given text. Sometimes, the NLP program may fail to correctly understand the meaning of a sentence, since the given context has a significant effect on the connotation of words or sentences.

A program that actually "understands" natural language is difficult to be determined in the NLP research. All we can actually test is whether a program appears to understand humans' language by successfully completing its task. The Turing test ($q.v.$), proposed by Turing, has been the classical model [9]. In this test, the NLP program has to be undistinguishable from a human when both answer arbitrary interrogation by a human over a terminal. A growing concern in NLP is developing more sensitive models of evaluation that can measure progress. The common method is to carry out evaluation tests within restricted domains to examine specific capabilities. For instance, statistical measures

can be computed relied on the set of human-generated questions collected in protocols ($q.v.$) that use another human to simulate the program.

Actually, the pervasive ambiguity is the major problem in processing natural language. Several common ambiguity are introduced in the following:

- **Simple lexical ambiguity**. For example, "bank" can be a noun (the financial institution) or a verb (to tip laterally).
- **Structural or syntactic ambiguity**. For example, in the sentence "Jack helped the woman with a wheelchair", the wheelchair might be utilized for the help or might be used by the woman being helped.
- **Semantic ambiguity**. For example, the word "make" has more than 10 different meanings in any dictionary.
- **Pragmatic ambiguity**. For example, "Could you pass the water to me?" may be a request to pass the water or a yes/no question.
- **Referential ambiguity**. For example, "Jorge talked with Frank in Starbucks. He looked bad . . . ," it is not clear who looks bad, even the remainder of the sentence might suggest a correct answer.

The history of NLP could be divided into four phases [30] with different concerns and styles. The first phase emerged from the need of Machine Translation in the 1940s [30]. Initially, machine translation mainly focused on English and Russian. Gradually, other languages such as Chinese also became popular in the 1960s. However, machine translation had little development during 1966 as the research of this field almost died at that time [56].

Since the need of Artificial Intelligence (AI) emerged, NLP acquired a new life in the 1980s [30]. This phase focused on meaning representation and gave more attention to world knowledge. LUNAR, developed by W.A woods in 1978, is a pioneering work of the question-answering systems influenced by AI [56]. The Yale group early recognized the need to explore the humans' goals if the NLP techniques wanted to completely understand the natural language. Thus, this phase also emphasized on both surface and underlying meanings of the language.

In the period of 1990s, NLP started to grow quickly. This trend was stimulated by the development of grammars, tools and practical resources [30]. For example, word sense disambiguation and statistically colored NLP had become the most striking feature of this decade. Additionally, NLP techniques of this phase involved other essential topics, such as semantic classification, information extraction, statistical language processing, and automatic summarizing [56].

Currently, everyone expects the machine to think and talk like humans, and the Natural Language Processing is the only method which can help us to achieve this goal. Many talking machines named Chatbot, such as Alexa developed by Amazon, can manage complicated interactions with human beings and process the streamlined business [56].

Particularly, the integration with Machine Learning and Deep Learning greatly expands the capabilities of NLP technology. Nowadays NLP techniques can be used to handle many different areas, such as health care, sentiment analysis, cognitive analytics, spam detection, human resources and conversational framework.

Text mining is the process to discover and extract useful and nontrivial information from unstructured text. This process involves the disciplines of information retrieval, text classification and clustering, and event extraction. NLP is the technique that attempts to explore the patterns to represent a full meaning of the unstructured text. NLP typically utilizes syntax techniques such as lemmatization, morphological segmentation, word segmentation, part-of-speech tagging, parsing, sentence breaking and stemming; semantics such as named entity recognition, word sense disambiguation and natural language generation; grammatical structure such as noun phrase, prepositional phrase, and dependency relations [31].

Modern NLP includes machine learning, machine translation, speech recognition, and machine text reading. Combining these branches together means that AI has the ability to acquire real knowledge from the world, not just learning the experience from humans. In the future, computers will have the capability of gaining information online and learn from it. With the continuous research, NLP will reach at a human level of understanding and awareness [35].

2.2 Text mining

Text mining is a young subject which focuses on detecting useful patterns from large database [3]. It combines the fields of information retrieval, data mining, machine learning, statistics and computational linguistics together, and extracts new pieces of knowledge from textual data. Although the sources of new knowledge are diverse, unstructured texts, such as webpages, e-mail messages, and many business documents etc., are still the largest readily accessible source of discovery [27]. Thus, the obvious distinction between text mining and regular data mining is that the patterns of data mining are extracted from structured database of facts [26].

With electronic files becoming the main method of sorting, storing and accessing written information, managing electronic information is changing the world greatly. Many other fields, like manufacturing, education, business, health care and government, can benefit from text-mining techniques. As a result, text mining technology and solutions are considered to have a high potential worth.

The main problem of text mining is machine intelligence [17]. It is easy for humans to identify and employ linguistic patterns to text and overcome obstacles like spelling variations, a slang and contextual meaning, while computers cannot manage them easily. On the other hand, humans are not able to handle texts in large volumes or at high speeds [17], though they are capable of comprehending unstructured data. Therefore,

creating a technique with both a computer's speed and accuracy and a human's linguistic capabilities is the key to text mining.

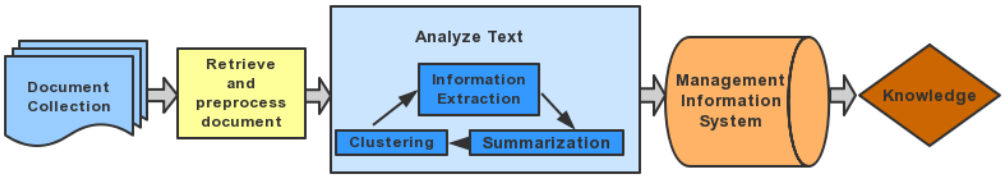

Figure 2.1. *A simple procedure of Text Mining [17].*

Figure 2.1 exhibits a simple example of the text mining procedure. It starts with a group of documents, and then checks the format and word lists of the retrieved particular document. After that, it implements text analysis, repeating a combinations of techniques, such as information extraction, clustering and summarization, to extract the targeted information. Finally, the management information system can process the resulting information to generate knowledge for its users.

In order to narrow the gap between humans and computers, various technologies that teach computers to analyze and understand natural languages have been produced. They consist of topic tracking, categorization, information extraction, information visualization, summarization, question answering, clustering, and concept linkage [17]. With these techniques, text mining has been demonstrated to be helpful in telecommunications, biomedical engineering, climate data, and geospatial data sets.

2.3 Text encoding

Preprocessing the text files and storing the contents in a data structure is a necessary step for text mining, which is convenient for further processing. Although exploring the syntactic structure and semantics is a key point in some methods, many text mining techniques rely on the concept that a set of words can describe a text document (bag-of-words representation). Due to the importance of bag-of-words representation, we will briefly describe how it can be obtained in the following.

2.3.1 Tokenization

The tokenization step is necessary to obtain all contained words for a given sentence. A token is a unit of text that is meaningful for analysis, such as a phrase or a word. Tokenization removes all punctuation marks and replaces other characters with white spaces, and then splits a text document into tokens. For text mining, it is easier and more

effective to use the specific tidy text format which is defined as a table with one-token-per-row [60]. In this one-token-per-row structure, the token can be a single word, an n-gram, a sentence or a paragraph. This tidy-text structure is then used for further processing.

For example, we have a sentence "$Because\ I\ could\ not\ stop\ for\ Death$". The process of tokenization needs to break the text into individual tokens "$because$", "i", "$could$", "not", "$stop$", "for", "$death$". For another example, the phrases "$new\ york\ university$", $right\ direction$, and $green\ gas$ can also be considered as tokens.

2.3.2 Filtering

Filtering approaches can filter words from the tidy text structure and thereby from the text document. A well known filtering approach is the removal of stop words [57]. Stop words, such as articles, conjunctions, prepositions etc., represent the items bearing little or no information of the contents. Moreover, words occurring extremely often are likely to bearing little knowledge, and also words occurring very seldom can be said to be no statistical relevance. Thus, all of these words belong to stop words. Stop-word filtering relies on the concept that removing non-discriminative words decreases the feature space of the classifier and assists them to generate more correct results [61]. For this reason, a stop dictionary includes the words that should be removed in the bag-of-words representation procedure. Although a set of general words, like and and or, can be seen as stop words in almost all cases, words of the stop dictionary are dependent in languages and tasks.

2.3.3 Lemmatization

For many applications of text mining, lemmatization is a significant preprocessing step. It is also extensively applied in NLP and other domains related to linguistics [49]. Lemmatization attempts to replace nouns with the singular form and verb forms with the infinite tense. In other words, lemmatization always looks for a transformation which can be applied to a word to obtain its normalized form. The lemmatization methods are similar to word stemming, except that lemmatization only requires to find the normalized form of a word but not to generate the word stem [49].

For example, the normalized form of the words $working$, $works$, $worked$ is $work$; and the word stem of them is also $work$. In this context, lemmatization is equal to word stemming. However, sometimes the normalized form is different with the stem of the word [49]. For instance, the words $computes$, $computing$, $computed$ would change to the normalized form $compute$ standing for the infinitive, but their stem word is $comput$.

2.3.4 Stemming

Generally, the morphological variants of words express the similar semantic meaning, and it consumes much time if each word form is considered different. Hence, it is essential to distinguish every word form with its basic form [29]. Stemming is also a preprocessing step in text-mining applications. It attempts to retrieve the base forms of words, i.e. remove the 's' from nouns, the 'ed' from verbs, or other affixes. In a word, stemming means a process to normalize all words that have the same stem to a basic form [37].
Stemming:
$introduction, introducing, introduces - introduc$
$walked, walking, walks - walk$

As explained in Lemmatization part, lemmatization requires to find the normalized form of a word (examples are showed in Lemmatization part). But stemming works by cutting off the common prefixes and suffixes of the word. This indiscriminate cutting offers limitations in some occasions. For example, the stem word of $studies$ would be $studi$ which is not so meaningful for our analysis. However, a lemma word is always the base form of all its inflectional forms. For the same example $studies$, its lemma word is $study$.

2.3.5 Linguistic processing

In addition to basic text-mining preprocessing, usually linguistic processing methods are also utilized to explore more information about text. For this reason, some frequently applied approaches are introduced in the following.

Part of speech (POS) is the conventional term that classifies words in a language [58]. The grammatical property of a word is the primary criteria for part-of-speech classification. In text mining, POS tagging is usually used to determine the part-of-speech tag, such as verb, noun, and adjective, for each item.

The textual unit of adjacent tokens is named as chunk [18], and text chunking means grouping an unstructured sequence of adjacent text units in a piece of text. Each chunk includes a set of adjacent words which are mutually linked through dependency chains of some specifiable kinds [18].

Human language is ambiguous and many words bear different interpretation based on the context. Word sense disambiguation (WSD) is a technique that attempts to recognize the interpretation of single word or phrase in the context [41]. For example, the word $bank$ clearly denotes meanings: a financial institution that accepts deposits or the slope beside a body of water. Although WSD causes a more complex dictionary, the core idea of considering many semantics of a term is close to human comprehension.

Parsing, also known as syntactic analysis, refers to the procedure that analyzes sentence structure and generates a full parse tree of a sentence [38]. With parsing, the relation

between every word and all the others can be easily explored.

2.4 Machine learning classifiers

In this section, we briefly review six machine-learning classifiers [11, 21, 34, 51, 54, 59] which are extensively applied in the field of sentiment analysis.

2.4.1 Naive Bayes classifier

The Naive Bayes classifier is dependent on the Bayes theorem with an assumption of independent predictors [54]. Despite its simplicity, Naive Bayes classifier works well on sentiment analysis. With Bayesian theorem, we can calculate posterior probability $P(c|X)$ from $P(c)$, $P(X)$ and $P(X|c)$ [54]:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{2.1}$$

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c) \tag{2.2}$$

where $P(X)$ is the prior probability of predictor, $P(X|c)$ is the probability of predictor given class, and $P(c)$ is the prior probability of class [54].

Naive Bayes classifier is simple and fast to classify the target data and it also performs outstanding in multiclass classification. Especially, Naive Bayes classifier needs less training data and performs better compared to other classifiers when holding independent assumption. However, its assumption also gives an obvious limitation, where independent predictors are almost impossible in real life.

2.4.2 Nearest Neighbor classifier

In text mining, the sentiment of the given sentence may be predicted from the sentiments of other similar sentences. Thus, this method is called as nearest neighbor classifier, where k-nearest neighbor classification (kNN) is the most frequently useful one [11]. kNN includes a training dataset of both positive and negative classes, and a target sentence is classified by computing the distance to the k nearest points and assigning the label of the majority [11]. To find out the best k, a cross validation method is often carried out. We generally create a list of k varying among some range and then the testing accuracy is given based on the validation set. Finally, a graph k vs $accuracy$ is plotted, and we can determine the best k among the range using the plot.

Nearest Neighbor classifier is quite simple and easy to implement. Its drawback is the

computational effort during classification. In another term, it is not practical enough and takes much time to test the data, where we tend to have fast testing to have real-time results.

2.4.3 Maximum Entropy

Maximum Entropy (MaxEnt) principle arises in statistical mechanics. Unlike Naive Bayes, MaxEnt has no assumptions of independence for its attributes. To model a given data set, it indicates that the most appropriate distribution is the one with highest entropy among all those satisfying the constrains of prior knowledge [21]. Because maximizing entropy minimizes the amount of prior information built into the distribution. The MaxEnt can be represented as the following:

$$P_{ME}(c|d,\lambda) = \frac{exp[\sum_i \lambda_i f_i(c,d)]}{\sum_c exp[\sum_i \lambda_i f_i(c,d)]} \tag{2.3}$$

Here c means the class, d describes the sentence, and λ indicates a weight vector. The weight vectors are optimized by numerical optimization of λ to maximize the conditional probability [21].

In text mining practice, the features to model MaxEnt are linguistically simple, but yet outperform many of learning algorithms under similar circumstances. The major disadvantage is that the exact maximum entropy solution does not exist in some cases, where the probability distribution may lead to poor testing accuracy [52].

2.4.4 Decision trees

Decision tree classifier, which involves decision nodes and leaf nodes, yields a final decision by repetitively dividing a dataset into gradually smaller subsets [51]. A decision node includes two or more branches, while a leaf node states a decision. Once the decision tree has been constructed, classifying a sentence is straightforward. Hence, constructing an optimal decision tree is the key aspect in the decision tree classifier. Let Dt be the training dataset and the attribute t_i is selected. Then Dt is split into two subsets. The subset Dt_i^+ contains the sentences including t_i, and the subset Dt_i^- includes the sentences without t_i. This process is recursively applied to Dt_i^+ and Dt_i^- until all sentences in a subset are classified as the same class.

Decision trees can easily handle irrelevant attributes and missing data, and they are also quite fast at testing time [51]. Nevertheless, the disadvantage is that the final decision is determined only on relatively few features in sentiment analysis. And also, sometimes they may not find the best tree in real world.

2.4.5 Support Vector Machines

The Support Vector Machine (SVM) means a discriminative classifier which attempts to find a separating hyperplane that distinctly classifies the data points in an n-dimensional space [59]. In other terms, SVMs output an optimal decision surface that categorizes new samples dependent on labeled training data. In sentiment analysis, the SVM classifier decides a hyperplane that is settled between the positive and negative categories of the dataset, where the margin is particularly maximized.

The learning of SVMs is almost regardless of the dimensionality of its feature space. Because feature selection is rarely required in SVM, SVM classifier is particularly suitable for text classification which usually involves a large amount of features. In addition, the kernel function does not have an important effect on the performance of text classification, since kernels are subject to overfitting [27].

2.4.6 Deep learning

In representation learning, the raw data is fed into a computer and the computer would automatically explore the needed representations for classification or detection [34]. Deep learning classifiers belong to representation learning approaches, which involve multiple levels of representation. As the non-linear modules of deep learning transform the representation between different levels, the classifier will learn quite complex functions during this process. For sentiment classification in text, the features that are significant for identification are amplified and the irrelevant parameters are suppressed by deeper layers of representation. Particularly, deep learning can use their purposed learning process to learn these layers of features, instead of designed by human engineers [34].

In recent years, deep learning has demonstrated outstanding performance in solving problems. It shows great capability in discovering intricate patterns for high-dimensional data, and it has been widely used in science, business and government [34]. However, there are a few challenges that have to be tackled to develop it. First, large amounts of data are required to train deep learning algorithms – as they learn progressively. Moreover, data availability for some sectors may be sparse and thus hamper deep learning in practice. Additionally, the high performing graphics processing units of deep learning require and consume a lot of power and are thereby a costly affair.

2.5 Lexicon-based methods

In this part, we briefly introduce six Lexicon-based approaches [7, 12, 16, 22, 63, 65] that are widely applied in the field of sentiment analysis.

2.5.1 Emoticons

Recently, Emoticons have become increasingly popular. Detecting the emoticons contained in the sentences is the simplest way for sentiment classification [22]. Because emoticons can represent happy or sad feelings, a set of common emoticons are utilized to extract polarity, such as exhibited in Table 2.1. This table consists of most popular emoticons and also their variations, which express positive, negative, and neutral polarities. If some sentences have more than one emoticon, then the polarity is determined with the first appeared emoticon in the text [22]. However, compared to total subjective sentences, the rate of online-social-network texts which contained at least one emoticon is less than 10% [22]. Hence, emoticons are usually used in combination with other methods for sentiment analysis.

Table 2.1. *Sample emoticons and their variations [22].*

Polarity	Positive	Negative	Neutral

Symbols	:) :] :} :o) :o] :o}
	:-] :-) :-} =) =] =}
	=^] =^) =^} :B :-D :-B
	:^D :^B =B =^B =^D :')
	:'] :'} =') ='] ='} <3
	^.^^-^^_^^^:* =*
	:-* ;) ;] ;} :-p :-P
	:-b :^p :^P :^b =P
	=p \o\/o/ :P :p :b =b
	=^p =^P =^b \o/
		D: D= D-: D^: D^= :(:[
		:{ :o(:o[:^(:^[:^{
		=^(=^{ >=(>=[>={ >=(
		>:-{ >:-[>:-(>=^[>:-(
		:-[:-(=(=[={ =^[
		>:-=(>=[>=^(:'(:'[
		:'{ ='{ ='(='[=\:\
		=/ :/ o.O O_o Oo =$
		:$:-{ >:-{ >=^{ :o{
			:| =| :-| >.<><>_<:o
			:0 =O :@ =@ :^o :^@ -.-
			-.-' -_- -_-' :x =X :#
			=# :-x :-@ :-# :^x :^#

2.5.2 LIWC

LIWC is a dictionary-based analysis tool, evaluating emotional, cognitive, and structural items of a given sentence [63]. Besides negative and positive categories, LIWC also detects other classes of sentiment. For instance, the term "agree" can be categorized into six word groups: affective, assent, positive feeling, positive emotion, and cognitive process. The LIWC2007 version contains labels for 100 word categories and more than 4500 English words, and Table 2.2 lists a simple example of LIWC2007 lexicon [63].

Currently, LIWC provides a commercial software and optimization option which allows users to create customized dictionaries rather than the standard ones. The LIWC software is accessible at `http://www.liwc.net/`.

Table 2.2. *A simple example of LIWC2007 lexicon [63].*

Category	Affective process	Cognitive process	Perceptual process
Examples	happy, cried, abandon, love, nice, sweet, hurt, ugly, nasty, worried, fearful, nervous, hate, kill, annoyed, crying, grief, sad	cause, know, ought, think, know, consider, because, effect, hence	observing, heard, feeling, view, saw, seen, listen, hearing, feels, touch

2.5.3 SentiStrength

The core idea of SentiStrength depends on the list of words from the LIWC lexicon. The authors added some new features, including a set of positive and negative words, the words to weaken (e.g., "a bit") or strengthen (e.g., "too") sentiments, emoticons, and repeated punctuation (e.g., "Good!!!!") to strengthen sentiments, to expand the baseline for the online-social-network context [65]. In the experiments, the authors evaluated six different datasets from Web 2.0: MySpace, BBC Forum, Twitter, Digg, YouTube Comments, and Runners World Forum [65]. Now, the authors also provides a useful tool to produce almost state-of-the-art results. Table 2.3 illustrates some sample words from the lexicon of SentiStrength.

Table 2.3. *Sample words from the lexicon of SentiStrength [65].*

List Name	Sentiment word list	Booster word list	Idiom list	Negation word list	Emoticon word list
Sample Word	Awful (-4)	Slightly (-1)	Shocker horror (-2)	Cant (-)	:'((-1)
and Score	Blissful (+5)	Extremely (+2)	Whats good (+2)	Never (-)	:-D (+1)

2.5.4 SentiWordNet

SentiWordNet relies on an English lexicon named WordNet. WordNet classifies nouns, verbs, adjectives and other grammatical groups into synsets [16]. The SentiWordNet using three values with synsets to state the polarity of the sentence: negative, positive, and neutral. The values are in the range of [0, 1] and the total sum is 1 [16]. For example, we have a given synset $s = [bad, wicked, terrible]$, and then SentiWordNet will score negative sentiment with 0.850, positive sentiment with 0.0, and neutral sentiment with 0.150, respectively.

Generally, scores from neutral sentiment have no effect on final sentiment decision. If the average positive score of all associated synsets of a target sentence is higher than that of the negative score, the polarity would be considered to be positive. Table 2.4 lists the 5 top-ranked positive and negative synsets in SentiWordNet 3.0.

Table 2.4. *The 5 top-ranked positive and negative synsets in SentiWordNet 3.0 [16].*

Rank	1	2	3	4	5
Positive	good#n#2 goodness#n#2	better_off#a#1	divine#a#6 elysian#a#2 inspired#a#1	good_enough#a#1	solid#a#1
Negative	abject#a#2	deplorable#a#1 distressing#a#2 lamentable#a#1 pitiful#a#2 sad#a#3 sorry#a#2	bad#a#10 unfit#a#3 unsound#a#5	scrimy#a#1	cheapjack#a#1 shoddy#a#1 tawdry#a#2

2.5.5 SenticNet

SenticNet is a tool extensively utilized in opinion mining, and its goal is to infer the polarity at a semantic level instead of the syntactic level [7]. SenticNet utilizes NLP approaches to create a polarity for almost 14000 concepts [7] which are defined as common sense – obvious things we normally know and usually leave unstated . For example, suppose that a given sentence "Great, it is Friday evening" is ready for sentiment classification, SenticNet first identifies concepts, which are "great" and "Friday evening" in this task. Then it produces sentiment score, between the values of -1 and 1, to every concept. In this task, "great" gets +0.383 and "Friday evening" gets +0.228, thereby the final sentiment score +0.3055 which is the average of the total values.

The authors of SenticNet evaluated it with the data of patients' opinions about the National Health Service in England and posts with over 130 moods from LiveJournal blogs [7]. Table 2.5 exhibits some sample concepts with sentiment scores in SenticNet.

Table 2.5. *Sample concepts with sentiment scores in SenticNet [7].*

Concept	Sentiment Score	Concept	Sentiment Score
Want degree	0.020	A lot	+0.970
Child play	0.023	A way of	+0.303
Grow up	0.290	Abandon	-0.858
Birthday cake	0.292	Abash	-0.130
Enough food	0.580	Abhor	-0.376
Wood spoon	-0.023	Able use	+0.941
Death row	-0.290	Abhorrent	-0.396
Break arm	-0.580	Mess up	-0.581

2.5.6 Happiness Index

The method of Happiness Index bases on the Affective Norms for English Words (ANEW) [12]. ANEW involves a list of 1034 unique words which are widely associated with their valence (ranging from pleasant to unpleasant), arousal (ranging from calm to excited)), and dominance [6]. Osgood's [44] seminal work indicates that these three dimensions can account for the variance in emotional assessments. Happiness Index indicating the quantity of happiness present in the message by giving scores for given terms between the values of 1 and 9. To increase the accuracy, the authors computed the appearing frequency of ANEW words in the sentences and calculated a weighted valence. When the weighted valence is evaluated on the song titles, song lyrics, and micro-blogs, the authors explored that the happiness value had decreased from 1961 to 2007 for song lyrics, while the value for blogs had increased during the same time [12].

To adapt this method for polarity classification, sentence that is categorized with Happiness Index in the spanning of [1, 5) is considered to be negative and in the spanning of [5, 9] is considered to be positive [12]. Table 2.6 presents some sample words from the lexicon of Happiness Index.

Table 2.6. *Sample words from the lexicon of Happiness Index [6].*

Words	Valence	Arousal	Dominance
Abduction	2.76	5.53	3.49
Bench	4.61	3.59	4.68
Carcass	3.34	4.83	4.90
Dawn	6.16	4.39	5.16
Ecstasy	7.98	7.38	6.68
False	3.27	3.43	4.10
Game	6.98	5.89	5.70
Happy	8.21	6.49	6.63
Illness	2.48	4.71	3.21

3 RESEARCH METHODOLOGY AND MATERIALS

The software and hardware resources, sentiment analysis techniques, datasets and performance assessment approaches are introduced in this section.

3.1 Software and hardware

All calculations were carried out in R. The R library $tidytext$ (version 0.2.0) [64] provided the text mining methods, i.e. word processing and sentiment analysis.

To accelerate parallel computing, we used the local grid computing resources (TUTGrid) provided by Tampere Center for Scientific Computing (TCSC), since the bootstrap and cross-validation parts could be evaluated independently at the same time.

3.2 A Lexicon-based method for sentiment analysis (AFINN)

The assumption of a lexicon-based method is that the sum of sentiment orientation of each word or phrase predicts the contextual sentiment orientation [68]. This approach creates a sentiment lexicon and rates the sentence according to the function that evaluates how the words and phrases of the sentence matches the lexicon. Sentiment analysis on web messages is challenging, which needs to process emoticons, informal words, word shortening, and spelling variation.

In different lexicon-based approaches, e.g., ANEW, SentiWordNet, and SentiStrength, the word dictionaries differ by the words they contain. Some word lists do not contain Internet slang acronyms and strong obscene words, such as "ROFL" and "WTF" [42]. However, such terms could play an important role in reaching outstanding performance while working with short informal texts from social networks. Another significant difference between word lists is that some word lists are scored with sentiment strength (e.g. specific scores) and the others are rated with positive/negative polarity (e.g. negative, positive or very positive) [42].

3.2.1 Construction of AFINN

Many lexicons have been developed for sentiment analysis, such as ANEW, SenticNet, and SentiWordNet. To analyze the sentiment of microblogs (e.g. Twitter), the need for novel word lists has obtained lots of attention. A New ANEW, which is termed as AFINN, is one of the simplest sentiment analysis methods [42]. AFINN is a word list with sentiment strength, which was constructed with the inclusion of Internet slang and obscene words. Initially, it was built up for sentiment analysis of tweets in relation to the United Nation Climate Conference in 2009 [42]. Since then AFINN has been gradually developed for various activities. The version named AFINN-96 is consist of 1468 different words, including a few phrases (i.e. $right\ direction$ and $not\ working$). The improved version named AFINN-111 includes 2477 unique words, including 15 phrases (i.e. $does\ not\ work$, $dont\ like$, $green\ washing$, and $not\ good$). Currently, the size of AFINN has been extended into 3383 English words [42].

AFINN initiates from a set of obscene words as well as a few positive words. Gradually, it was expanded with tweets collected for the United Nation Climate Conference, the public dictionary *Original Balanced Affective Word List* by Greg Siegle, Internet slang acronyms such as WTF, LOL and ROFL, and the word list *The Compass DeRose Guide to Emotion Words* by Steven J. DeRose [42]. The author determined in which contexts the word appeared by using Twitter, and he also discovered relevant words by using the Microsoft Web n-gram similarity Web service. To avoid ambiguities, the author excluded words like power, firm, patient, mean and frank. And also the words with high arousal but with variable sentiment, such as "surprise", were excluded from AFINN.

Like SentiStrength [65] rating range from -5 to $+5$, the author also scored AFINN from -5 to $+5$ for ease of labeling. Figure 3.1 exhibits the histogram of sentiment scores for AFINN. As illustrated in Figure 3.1, the majority of the negative words were labeled by -2 and most of the positive words by $+2$ in AFINN. But some strong obscene words were scored with either -4 or -5, such as asshole (-4), bastard (-5), bitch (-5), and bullshit (-4). Compared to the number of positive words (878), AFINN has a bias towards negative words (1598), which also occurs similarly in the OpinionFinder sentiment lexicon (2718 positive and 4911 negative words) [42].

Table 3.1 shows a simple example of AFINN lexicon. Definitely, AFINN is one of the most popular word lists that could be utilized widely for sentiment analysis. Although AFINN was initially constructed for sentiment analysis on Twitter, we can get a good idea of general sentiment statistics across different text categories with it. The author has also created nice wrapper libraries in both *R* and *Python*, which could be used directly for analysis. All versions of this lexicon can be found at the author's official GitHub repository.

Table 3.1. *An example of AFINN lexicon [42].*

Scores	-5	-4	-3	-2	-1	1	2	3	4	5
Words	bastard	bullshit	abhor	abandon	absentee	aboard	ability	admire	amazing	breathtaking
	bitch	catastrophic	abuse	abduction	admit	absorbed	absolve	adorable	awesome	hurrah
	cock	damn	acrimonious	accident	affected	accept	accomplish	affection	brilliant	outstanding
	cunt	dick	agonize	accusation	afflicted	achievable	acquit	amuse	ecstatic	
	prick	fraud	anger	accuse	affronted	active	advantage	astound	overjoyed	
		fuck	angry	ache	alas	adequate	adventure	audacious		
		jackass	anguish	admonish	alert	adopt	agog	award		
		motherfucker	bad	afraid	ambivalent	advanced	agreeable	beautiful		
		nigger	betray	aggravate	apology	agree	amaze	best		
			disastrous	aggression	empty	backs	ambitious			
			disgusted	aghast	envy		defender			
			distrust	alarm	escape					
			douche	degrade	eviction					
			dreadful	distort						
				embarrass						
				emergency						

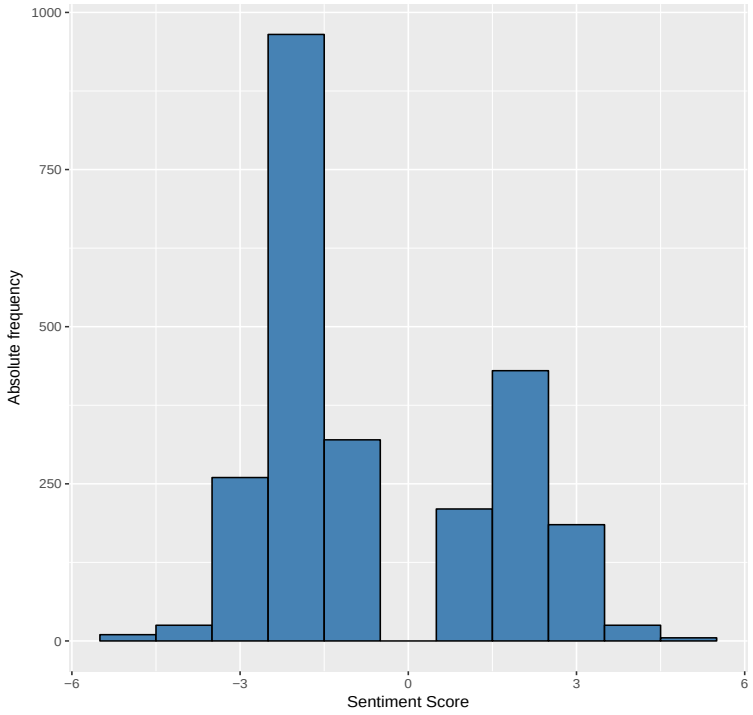

Figure 3.1. *Histogram of sentiment scores for AFINN [42].*

3.2.2 Sentiment prediction

A typical sentence includes word variations, emoticons, hashtags etc. Therefore, we need the preprocessing steps to normalize the sentence before sentiment prediction.

- **POS Tagging:** POS tagging indicates the procedure of identifying a particular part of speech of a word, given both its definition and context. The process is complicated because a single word possibly has an unique part of speech tag in different sentences given different contexts. POS Tagger could give part-of-speech tag associated with words.

- **Stemming:** As explained in previous part, stemming means a process to normalize all words that have the same stem to a basic form. And this process could help the computer to match the word in the sentence to the lexicon.
- **Exaggerated word shortening:** AFINN lexicon contains normal English words only. Thus, if words have same letter more than twice but not existing in AFINN, the words will be simplified to the word with the repeating letter just once [68]. For instance, the exaggerated word "Yesssss" is reduced to "Yes".
- **Hashtag detection:** Hashtag is a phrase which starts with # with no space between them. If a sentence contains a hashtag, the hashtag will be a topic or a keyword of the sentence. Hashtags could provide important information for sentiment analysis.

Sentiment prediction indicates the aggregation of the sentiment bearing words of the sentence. We extract sentiment words from sentences and sentiment scores are then assigned to these words. The final polarity of a sentence bases on the sum of scores from all its words. Algorithm 1 shows the sentiment prediction algorithm.

Algorithm 1 Sentiment Prediction

Require: Preprocessed sentences
Ensure: Results: Positive, Negative, Neutral
  Build up the table of sentiment words $SentiWords$;
  $SentiScore$ = 0;
  for $each\ word\ in\ the\ SentiWords$ **do**
    $SentiScore$ = $SentiScore$ + sentiment of $word$;
  end for
  if $Hashtag\ is\ existing$ **then**
    Extract all the sentiment words in hashtag and add them to $SentiWords$
  end if
  $SentiClass$ = "Neutral";
  if $SentiScore > 0$ **then**
    $SentiClass$ = "Positive";
  end if
  if $SentiScore < 0$ **then**
    $SentiClass$ = "Negative";
  end if
  return $SentiClass$

3.3 Stanford Recursive Neural Tensor Network (RNTN)

Although semantic vector spaces have been used widely to represent single words, they cannot represent longer phrases appropriately. The problem is that capturing such pattern in the sentences requires powerful models and large training resources. To remedy this, the Stanford Sentiment Treebank and the Recursive Neural Tensor Network (RNTN) [62] are used to predict the compositional semantic effects. Figure 3.2 shows one example of RNTN with clear compositional structure. In Figure 3.2, RNTN can capture the negation and its scope in the sentence and predict 5 sentiment classes ($--$, $-$, 0, $+$,

++) at each node of a parse tree.

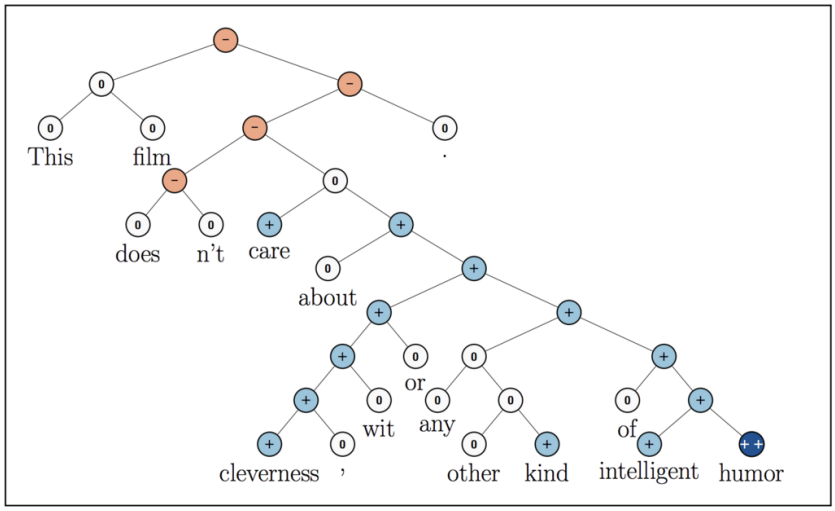

Figure 3.2. *An example of RNTN [62].*

3.3.1 Stanford Sentiment Treebank

The Stanford Sentiment Treebank is a corpus with labeled parse trees, which allowing us to analyze completely the sentiment in a language [62]. The original dataset of movie reviews, including 10,662 single sentences, was initially collected by Pang and Lee [45]. Moreover, half sentences of this dataset were considered negative and the other half positive. All these sentences were parsed with the Stanford Parser [32], thereby resulting 215,154 unique phrases from these parse trees [62]. These phrases are labeled with Amazon Mechanical Turk, forming the corpus of Stanford Sentiment Treebank. This new corpus provides the community with the possibility to train compositional models by machine learning techniques [62].

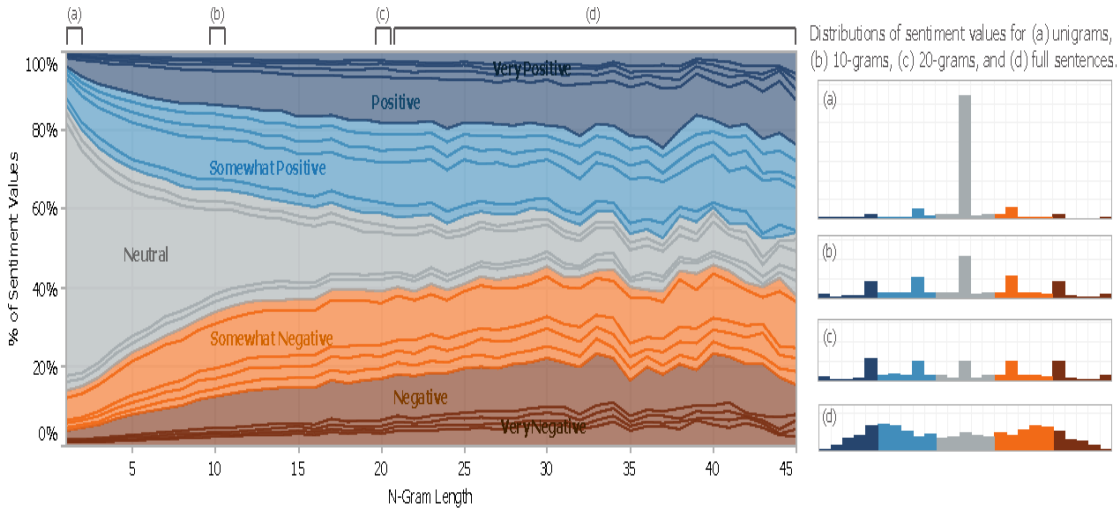

Figure 3.3. *The normalized histogram of sentiment labels at each n-gram length [62].*

Figure 3.3 illustrates the normalized histogram of sentiment labels at each n-gram length. The authors noticed that longer phrases often represented stronger sentiment and the majority of the shorter phrases were neutral [62]. Since the extreme values were quite rare, the labels of the Stanford Sentiment Treebank only covered 5 classes: negative, somewhat negative, neutral, somewhat positive and positive [62].

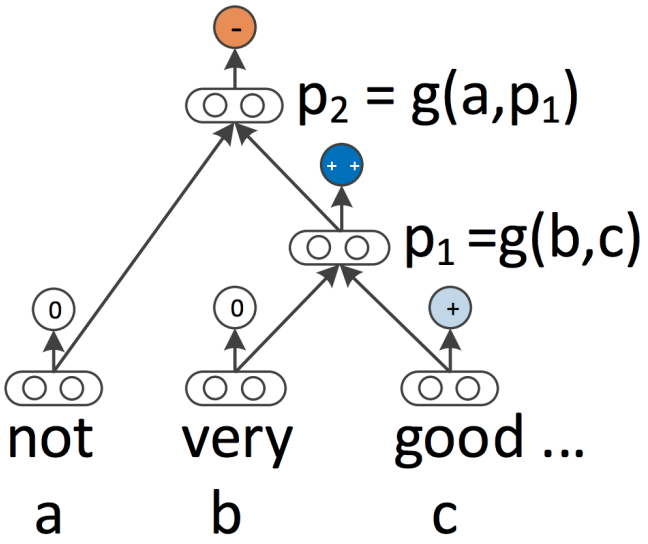

Figure 3.4. *Approach of computing compositional vector representations for phrases [62].*

3.3.2 Recursive Neural Models

Figure 3.4 displays the approach of computing compositional vector representations for phrases. When an n-gram is given as an input, a binary tree is built up by parsing the input into separate words which are corresponding to each leaf node. The recursive neutral model will then use various functions g to compute parent vectors in a bottom up fashion. And the parent vectors will work as features for classification.

A d-dimensional vector is utilized to represent each word. And the word vector is initialized by randomly sampling from a uniform distribution: $U(-r, r)$, where $r = 0.0001$ [62]. The embedding matrix $L \in \Re^{d \times |V|}$ stacks all the word vectors, and $|V|$ indicates the size of the vocabulary. Moreover, the word vectors can work as features and also parameters to optimize a $softmax$ classifier [62]. For five-class classification, the posterior probability with labels is computed based on the word vector:

$$y^a = softmax\left(W_s a\right) \tag{3.1}$$

where $W_s \in \Re^{5 \times d}$ is the sentiment classification matrix. Similarly, we can get the posterior probability for the vectors b and c in Figure 3.4 using the same step.

The standard recursive neural network (RNN) is one of the most popular techniques for

text mining. The Equations 3.2 represent that how RNNs compute the parent vectors:

$$p_1 = f\left(W\begin{bmatrix} b \\ c \end{bmatrix}\right), p_2 = f\left(W\begin{bmatrix} a \\ p_1 \end{bmatrix}\right) \tag{3.2}$$

where $f = tanh$ denotes a standard element-wise nonlinearity, $W \in \Re^{d\times 2d}$ indicates the main parameters, and the bias is omitted for simplicity. The parent vectors must have the same dimensionality and each parent vector p_i uses the same $softmax$ function of Equation 3.1 to compute its label probability.

Most parameters of Matrix-Vector RNN (MV-RNN) are associated with words. The MV-RNN uses both a vector and a matrix to represent each word or phrase in a parse tree [62]. The matrix of every word is initialized as a $d \times d$ identity matrix with a minor number of Gaussian noise. For the parse tree containing vector and matrix nodes, the MV-RNN computes the first parent vector and its matrix:

$$p_1 = f\left(W\begin{bmatrix} Cb \\ Bc \end{bmatrix}\right), P_1 = f\left(W_M\begin{bmatrix} B \\ C \end{bmatrix}\right) \tag{3.3}$$

where $W_M \in \Re^{d\times 2d}$ is a $d \times d$ matrix. Similarly, we can compute the second parent node using the previous (vector, matrix) pair (p_1, P_1).

3.3.3 Recursive Neural Tensor Network

However, both RNN and MV-RNN have their own problems. The problem of RNN is that the input vectors only implicitly interact through the non-linearity function, and the number of parameters in MV-RNN is too large for processing. In order to address these problems, the authors proposed a more powerful single composition model: the Recursive Neural Tensor Network (RNTN). For all nodes in the parse tree, RNTN attempts to use the same and tensor-based composition function, which is also the core idea of RNTN [62].

For each slice $V^{[i]} \in \Re^{d\times d}$, the output of a tensor product $h \in \Re^d$ is defined as:

$$h = \begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} b \\ c \end{bmatrix}; h_i = \begin{bmatrix} b \\ c \end{bmatrix}^T V^{[i]} \begin{bmatrix} b \\ c \end{bmatrix} \tag{3.4}$$

where $V^{[1:d]} \in \Re^{2d\times 2d\times d}$ is the tensor that defines multiple bi-linear forms.

The RNTN uses this definition to compute p_1:

$$p_1 = f\left(\begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} b \\ c \end{bmatrix} + W \begin{bmatrix} b \\ c \end{bmatrix}\right) \tag{3.5}$$

where W is the same with that defined in the previous models. The same weights can be used to compute the next parent vector p_2 in Figure 3.4:

$$p_2 = f\left(\begin{bmatrix} a \\ p_1 \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} a \\ p_1 \end{bmatrix} + W \begin{bmatrix} a \\ p_1 \end{bmatrix}\right) \tag{3.6}$$

The following part describes the training step for the RNTN model. As explained above, each node predicts the target vector t via a $softmax$ classifier which is trained on its vector representation. It is assumed that the target distribution vector at each node has a 0-1 encoding [62]. If there are C classes, the assumption is that the target vector has length C and all other entries are 0 except a 1 at the correct label.

The error function of the RNTN parameters $\theta = (V, W, W_s, L)$ for a sentence is:

$$E(\theta) = \sum_i \sum_j t_j^i log y_j^i + \lambda ||\theta||^2 \tag{3.7}$$

x^i denotes the vector at node i. Each node back-propagates its error to the recursively used weights V, W. Then $\delta^{i,s} \in \Re^{d \times 1}$ will be the $softmax$ error vector at node i:

$$\delta^{i,s} = \left(W_s^T \left(y^i - t^i\right)\right) \otimes f'\left(x^i\right) \tag{3.8}$$

where $\otimes$ is the Hadamard product between the two vectors. f' is the element-wise derivative of $f = tanh$.

For the derivative of each slice $k = 1, \ldots, d$:

$$\frac{\partial E^{p_2}}{\partial V^{[k]}} = \delta_k^{p_2,com} \begin{bmatrix} a \\ p_1 \end{bmatrix} \begin{bmatrix} a \\ p_1 \end{bmatrix}^T \tag{3.9}$$

where $\delta_k^{p_2,com}$ is the k'th element. Next, the error message for the two children of p_2 can be computed:

$$\delta^{p_2,down} = \left(W^T \delta^{p_2,com} + S\right) \otimes f'\left(\begin{bmatrix} a \\ p_1 \end{bmatrix}\right) \tag{3.10}$$

where

$$S = \sum_{k=1}^{d} \delta_k^{p2,com} \left(V^{[k]} + \left(v^{[k]} \right)^T \right) \begin{bmatrix} a \\ p_1 \end{bmatrix} \tag{3.11}$$

The complete δ comes from two parts. One is that the children of p_2 each take half of this vector and the other is their own $softmax$ error message [62]. Then we have:

$$\delta^{p_1,com} = \delta^{p_1,s} + \delta^{p_2,down}[d+1:2d] \tag{3.12}$$

where p_1 is the right child of p_2 and hence takes the 2^{nd} half of the error of p_2.

For the tri-gram tree in Figure 3.4, the full derivative for slice $V^{[k]}$ is calculated:

$$\frac{\partial E}{\partial V^{[k]}} = \frac{E^{p_2}}{\partial V^{[k]}} + \delta_k^{p_1,com} \begin{bmatrix} b \\ c \end{bmatrix} \begin{bmatrix} b \\ c \end{bmatrix}^T \tag{3.13}$$

and similarly for W.

3.4 Data

Using gold standard labeled datasets is a key aspect in comparing sentiment analysis methods. Table 3.2 presents the main characteristics of seven datasets covering a wide range of sources. For example, the number of messages, the number of messages in each sentiment class, and the average number of words per message are summarized in Table 3.2. "#Pos" indicates the positive class, "#Neg" means the negative class, and "#Neu" states the neutral class. Additionally, Table 3.2 also states the methodology applied in the sentiment classification. Labeling with Amazon Mechanical Turk (AMT) was carried out in three out of the seven datasets, while the left datasets use volunteers and other strategies which contain non-expert annotators. Generally, an agreement strategy, such as majority voting, ensures that each sentence has an agreed-upon polarity in each dataset. Table 3.2 also shows the number of evaluators who labeled the datasets.

The New York Times (NYT) is an American newspaper which has significant worldwide influence and readership. The NYT provides its readers with different sections on various topics, such as sports, arts, science, home, and travel. Comments_NYT includes 5190 sentence-level snippets from 500 New York Times opinion editorials [20]. This dataset was labeled with AMT, and there are 2204 positive messages, 2742 negative messages and 244 neutral messages.

TED (`www.ted.com`) is a popular media organization which posts public talks and user-contributed material (favorites, comments) [48]. It is an online repository that includes

talks on many political, scientific, academic, and cultural topics under a Creative Commons license. The authors of Comments_TED crawled the TED website in September 2012, and they collected sentences from 74,760 users and 1,203 talks, with 134,533 favorites and 209,566 comments [48]. Comments_TED, including 839 comments, is a subset of original dataset with 839 positive sentences, 318 negative sentences and 409 neutral sentences.

YouTube is a popular American video-sharing website, allowing its users to watch, share, and comment on videos. YouTube provides a wide range of contents, such as movie trailers, music videos, documentary films, and TV shows. Comments_YTB includes 3407 text comments posted to videos on the YouTube website [65]. In Comments_YTB, there are 1665 positive comments, 767 negative comments and 975 neutral comments.

Myspace is a social networking website, offering an interactive network of friends, blogs, photos, personal profiles, groups, videos, and music. Myspace was the most common social networking from 2005 to 2009 in the world [65]. In this thesis, Myspace is a corpus of 1041 comments from the social network site MySpace, including 702 positive comments, 132 negative comments and 207 neutral comments [65].

Twitter is a microblogging site in the Web. Furthermore, we usually gather tweets which express sentiment on popular topics from Twitter. Tweets_Semeval is the dataset used in SemEval-2013 task 2, and the authors first used a Twitter-tuned NER system to extract name entities from millions of tweets, which they gathered over a one-year period ranging from January 2012 to January 2013 [39]. Here, Tweets_Semeval is just a subset of the original dataset with 6087 tweets, including 2223 positive tweets, 837 negative tweets and 3027 neutral tweets.

Tweets_RND_I comes from Thewall's study that attempted to explore the representative patterns of sentiment changes in an event. In addition, the data was used to determine whether sentiment changes could indicate the amount of interest in an event during the early stages of its evolution [65]. The raw dataset is consist of 35 million tweets spanning from February 9, 2010 to March 9, 2010 [65]. Here, Tweets_RND_I only includes 4242 tweets, where 1340 tweets are positive, 949 tweets are negative and 1953 tweets are neutral.

Tweets_RND_III is a training dataset of 3771 tweets. Tweets were collected from timelines of randomly selected Twitter users [40]. In Tweets_RND_III, 739 tweets are positive, 488 tweets are negative and 2536 tweets are neutral.

Cohen's Kappa is a common metric to compute inter-annotator agreement [53]. In Table 3.2, this approach is used to calculate column CK, exhibiting the level of agreement of each dataset. The given sentences with mixed polarity could be the possible reasons for the disagreement with the evaluations. Indeed, some of them are quite tricky to annotate because they are strongly related to original context. Landis and Koch suggest that Kappa values mean moderate agreement if they amid 0.4 and 0.6, and values between 0.6 and 0.8 indicate substantial agreements [33].

Table 3.2. *Overview of our datasets for sentiment analysis.*

Dataset	# Msgs	# Pos	# Neg	#Neu	Average # of phrases	Average # of words	Annotators expertise	# of annotators	CK
Comments_NYT	5190	2204	2742	244	1.01	17.76	AMT	20	0.628
Comments_TED	839	318	409	112	1	16.95	Non expert	6	0.617
Comments_YTB	3407	1665	767	975	1.78	17.68	Non expert	3	0.724
Myspace	1041	702	132	207	2.22	21.12	Non expert	3	0.647
Tweets_RND_I	4242	1340	949	1953	1.77	15.81	Non expert	3	0.683
Tweets_RND_III	3771	739	488	2536	1.54	14.32	AMT	3	0.824
Tweets_Semeval	6087	2223	837	3027	1.86	20.05	AMT	5	0.617

3.5 Performance assessment

To provide a more thorough comparison among AFINN and Stanford RNTN, two tests are performed in this thesis. In the first test, we consider 3-class (positive, negative and neutral) identification of these two methods. In the second test, we remove the neural messages firstly and then only consider two classes: positive and negative class. All these experiments were carried out with the datasets described in Table 3.2.

Table 3.3 exhibits a 3 $\times$ 3 confusion matrix for the possible classification outcomes.

Table 3.3. *A 3 $\times$ 3 confusion matrix for 3-class classification [53].*

		Predicted class		
		Positive	Neutral	Negative
True class	Positive	a	b	c
	Neutral	d	e	f
	Negative	g	h	i

In Table 3.3, each letter describes the number of samples. The recall R of a class is the fraction of the known elements that are correctly classified in the classification. Precision P of a class is the ratio of the number of instances classified correctly to the total class with same label in prediction [53]. For instance, the precision of the negative class is:

$$P(neg) = \frac{i}{c + f + i} \tag{3.14}$$

its recall:

$$R(neg) = \frac{i}{g + h + i} \tag{3.15}$$

$F1$ score [53] is the harmonic average of precision P and recall R:

$$F1(neg) = \frac{2P(neg) \times R(neg)}{P(neg) + R(neg)} \tag{3.16}$$

The overall accuracy A is computed as:

$$A = \frac{a+e+i}{a+b+c+d+e+f+g+h+i} \tag{3.17}$$

The correct prediction of each sentence is considered equally important in the overall accuracy A. And it also generally measures the ability of the approach to produce the correct result. $Macro{-}F1$ scores are calculated by first computing $F1$ scores for each class independently, and then averaging over all classes [53].

$$Macro{-}F1 = \frac{F1(pos) + F1(neg) + F1(neu)}{3} \tag{3.18}$$

Accuracy A and $Macro{-}F1$ can offer complementary evaluation of the classification effectiveness. $Macro{-}F1$ is especially significant to verify the ability of the approach to perform well in very skewed classes [53].

Table 3.4 presents a 2 $\times$ 2 confusion matrix for the possible classification outputs. For example, the precision P of positive class is calculated as:

$$P(pos) = \frac{a}{a+c} \tag{3.19}$$

its recall:

$$R(pos) = \frac{a}{a+b} \tag{3.20}$$

while its $F1$ score is:

$$F1(pos) = \frac{2P(pos) \times R(pos)}{P(pos) + R(pos)} \tag{3.21}$$

Table 3.4. *A 2 $\times$ 2 confusion matrix for 2-class classification [15].*

	Predicted class		
		Positive	Negative
True class	Positive	a	b
	Negative	c	d

3.5.1 Cross validation

Cross validation is a popular methodology to assess how well the classifier generalizes. The core idea of cross validation is to divide dataset, once or several times, into subsets. And every subset (the validation samples) is then used for testing, while the remaining

subsets (the training samples) are used for training the classifier. Let's assume our data set named D, and the dataset D would be randomly divided into k disjoint subsets D_1, D_2, $\cdots$, D_k in k-fold cross-validation. If $|D| = n$, then the size of each subset is n/k. Then the classifier is trained and tested k times; each time it is trained on $k-1$ subsets and tested on the remaining one subset [2]. The accuracy A_{cv} of the cross validation is the mean of the accuracy derived in all the k cases of cross validation:

$$A_{cv} = \frac{1}{k}\sum_{i=1}^{k} A_i^{'} \tag{3.22}$$

where A_i^i is the i^{th} accuracy. The estimated error E_{cv} is then the average of the k errors:

$$E_{cv} = \frac{1}{k}\sum_{i=1}^{k} E_i^{'} \tag{3.23}$$

where $E_i^{'}$ is the i^{th} error.

Leave-one-out cross validation (LOOCV) is the specific case of k-fold cross validation, where $k = n$ [2]. That means each subset only contains a single sample. Especially, LOOCV is usually used in the tasks where the samples are less than one hundred with very small number of features.

3.5.2 Bootstrap

Bootstrap is a recently developed non-parametric technique for making certain kinds of statistical inferences [13]. For our dataset D ($|D| = n$), sampling n elements uniformly from the dataset with replacement builds a proper bootstrap set. Since the sampling process is performed with replacement, the probability of any element being chosen after n times is $1-(1-1/n)^n \approx 1-e^{-1} \approx 0.632$. The accuracy A_{boot} is obtained by using the bootstrap set for training and the rest of the original dataset for testing.

Although the training datasets and the testing datasets are the same, some models are not entirely stable in some cases. In addition, the result is most likely to change if we randomly balance the dataset before training. Thus, we have to calculate the standard error to better compare the resulting performance statistics. For instance, the standard error of a sample mean $\overline{x}$ is

$$SE(\overline{x}) = \frac{s}{\sqrt{n}} \tag{3.24}$$

where s is the standard deviation of the sample mean.

4 RESULTS AND DISCUSSION

This chapter contains the results of the computational research conducted in this thesis. We have classified each dataset with the help of two different sentiment analysis approaches namely AFINN and Stanford Recursive Neural Tensor Network (RNTN). The results of 2-class and 3-class comparisons could provide a more thorough comparison among AFINN and RNTN. Then the discussions on our results are also presented.

To make more robust comparison among AFINN and RNTN, the hypothesis test is needed in each experiment. Table 4.1 shows the null hypothesis H_0 and alternative hypothesis H_1 of the hypothesis test. If p-value > 0.05, we cannot reject H_0; if p-value < 0.05, we can reject H_0 and accept H_1.

Table 4.1. *The hypothesis test in each experiment.*

Test	Hypothesis
1	H_0: The LS regression line has the same performance with the diagonal line for representing our datasets. H_1: The LS regression line does not have the same performance with the diagonal line for representing our datasets.

4.1 2-Class comparisons

In 2-class comparisons, there are no results of the neutral class because they can only detect the positivity or negativity of a sentence. Additionally, the comparisons include all datasets namely Comments_NYT, Comments_TED, Comments_YTB, Myspace, Tweets _RND_I, Tweets_RND_III and Tweets_Semeval but excluding the neutral sentences. In the following section, the results of bootstrap and cross-validation tests are separately illustrated.

4.1.1 Bootstrap

Table 4.2 shows the fundamental errors and their corresponding standard errors (SE) for AFINN, and Table 4.3 presents the fundamental errors and their corresponding standard errors for RNTN.

The diagonal model denotes a model with slope = 1 and intercept = 0, which means RNTN and AFINN have the same performance under the evaluated metric. The Least

Table 4.2. *The fundamental errors of 2-class bootstrap with seven datasets for sentiment analysis, classifier = AFINN.*

Metrics	Comments_NYT	Comments_TED	Comments_YTB	Myspace	Tweets_RND_I	Tweets_RND_III	Tweets_Semeval
TPR (%)	41.03 ± 0.64	40.14 ± 1.32	65.98 ± 0.81	78.33 ± 1.6	54.5 ± 1.58	56.83 ± 1.72	68.07 ± 0.93
TNR (%)	16.2 ± 0.52	15.89 ± 1.37	11.27 ± 0.78	7.5 ± 0.81	14.85 ± 0.73	19.43 ± 1.66	11.34 ± 0.36
FPR (%)	39.33 ± 0.78	40.96 ± 2.01	20.04 ± 0.61	8.81 ± 1.04	27.42 ± 1.05	20.98 ± 0.99	16.7 ± 0.62
FNR (%)	3.43 ± 0.2	3.01 ± 0.45	2.7 ± 0.31	5.36 ± 0.57	3.23 ± 0.36	2.76 ± 0.52	3.89 ± 0.26

Table 4.3. *The fundamental errors of 2-class bootstrap with seven datasets for sentiment analysis, classifier = RNTN.*

Metrics	Comments_NYT	Comments_TED	Comments_YTB	Myspace	Tweets_RND_I	Tweets_RND_III	Tweets_Semeval
TPR (%)	17.9 ± 0.51	27.67 ± 1.19	44.92 ± 1.1	41.07 ± 1.69	18.25 ± 0.64	32.28 ± 1.36	20.07 ± 0.64
TNR (%)	43.37 ± 0.61	38.9 ± 1.64	22.09 ± 0.59	11.67 ± 0.94	35.72 ± 1.19	34.23 ± 1.89	24.64 ± 0.85
FPR (%)	12.16 ± 0.37	17.95 ± 1.53	9.22 ± 0.74	4.64 ± 0.9	6.55 ± 0.54	6.18 ± 0.64	3.4 ± 0.22
FNR (%)	26.57 ± 0.66	15.48 ± 1.23	23.77 ± 1.04	42.62 ± 1.66	39.48 ± 1.22	27.32 ± 0.84	51.9 ± 0.92

Squares (LS) regression model and the diagonal model for the standard errors of fundamental errors of AFINN and RNTN are exhibited in Figure 4.1, and Figure 4.2 exhibits the LS regression model and the diagonal model for the fundamental errors of AFINN and RNTN.

Table 4.4 provides the hypothesis testing results for Figure 4.1 and Table 4.5 shows the hypothesis testing results for Figure 4.2. The p-value (Comparison) indicate the results of our hypothesis test in Table 4.1.

Table 4.4. *The hypothesis testing results for Figure 4.1.*

Metrics	Intercept (%)	p-value (Intercept)	Slope (%)	p-value (Slope)	R^2	p-value (Comparison)
SE_TPR	0.2761	0.5816	0.6043	0.1568	0.3567	0.2976
SE_TNR	0.2456	0.3184	0.9616	0.0078	0.7859	0.4007
SE_FPR	-0.0935	0.6767	0.7879	0.0091	0.7729	0.2054
SE_FNR	0.4658	0.1994	1.6141	0.0945	0.4589	0

By analyzing the p-value (Comparison) in Table 4.4, we can conclude that: H_0 cannot be rejected for (a) AFINN-RNTN SE of TPR in Figure 4.1; H_0 cannot be rejected for (b) AFINN-RNTN SE of TNR in Figure 4.1; H_0 cannot be rejected for (c) AFINN-RNTN SE of FPR in Figure 4.1; H_0 can be rejected for (d) AFINN-RNTN SE of FNR in Figure 4.1.

By analyzing the p-value (Comparison) in Table 4.5, we can conclude that: H_0 can be rejected for (a) AFINN-RNTN TPR in Figure 4.2; H_0 can be rejected for (b) AFINN-RNTN TNR in Figure 4.2; H_0 can be rejected for (c) AFINN-RNTN FPR in Figure 4.2; H_0 can be rejected for (d) AFINN-RNTN FNR in Figure 4.2.

Table 4.6 and Table 4.7 indicate the overall performance assessment results for AFINN and RNTN, respectively.

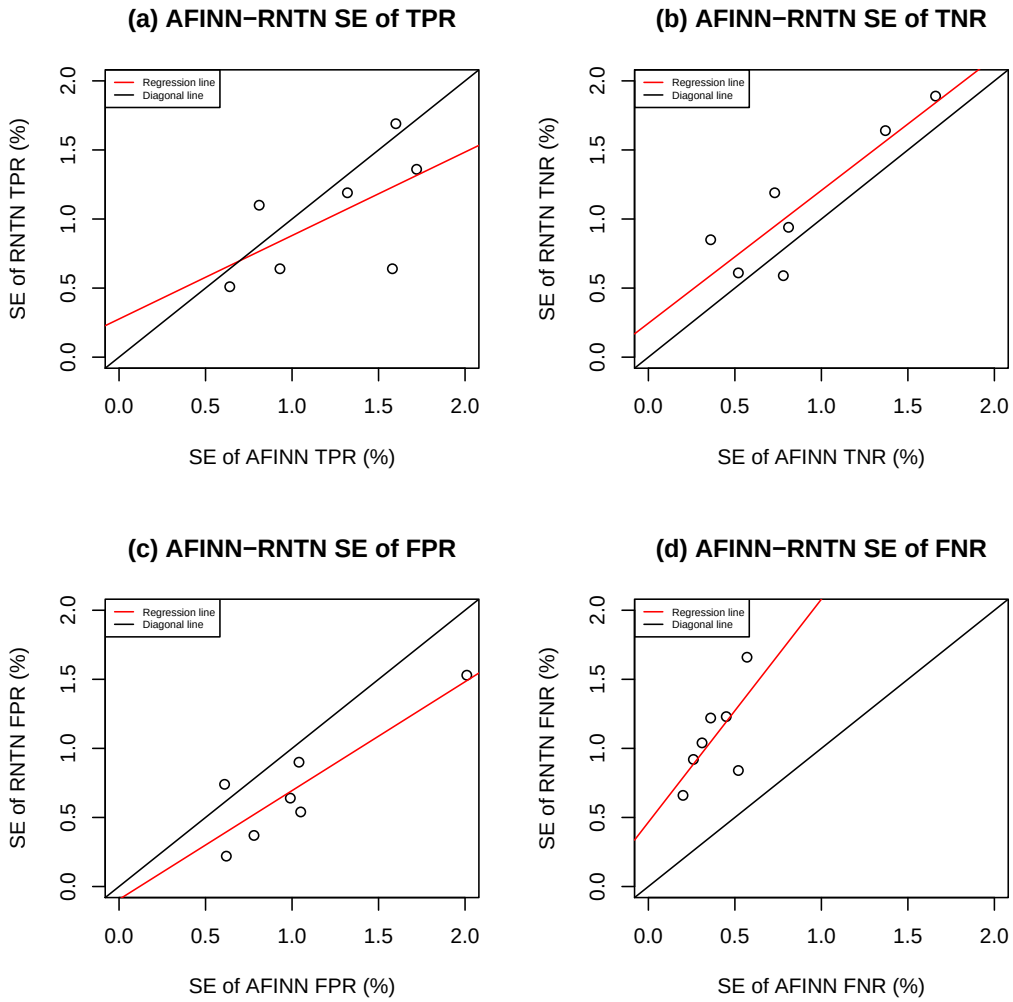

Figure 4.1. *The LS regression model (the red line) and the diagonal model (the black line) for the standard errors of fundamental errors of AFINN and RNTN in 2-class bootstrap.*

Table 4.5. *The hypothesis testing results for Figure 4.2*

Metrics	Intercept (%)	p-value (Intercept)	Slope (%)	p-value (Slope)	R^2	p-value (Comparison)
TPR	3.3916	0.8504	0.4407	0.1866	0.3189	0.001
TNR	-3.1402	0.7374	2.4109	0.0116	0.751	0.0032
FPR	-0.6027	0.8277	0.3691	0.0124	0.7447	0.0097
FNR	2.4413	0.8915	8.6157	0.1291	0.3974	1e-04

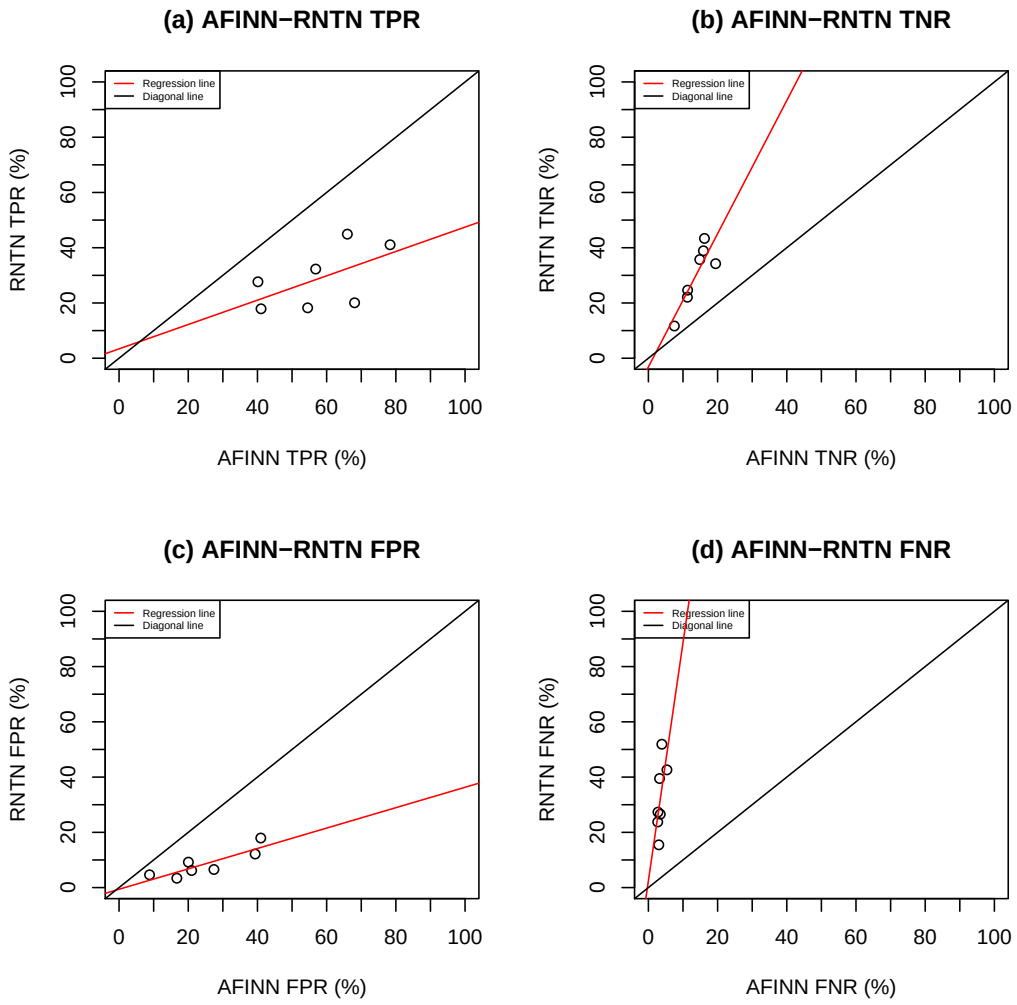

Figure 4.2. *The LS regression model (the red line) and the diagonal model (the black line) for the fundamental errors of AFINN and RNTN in 2-class bootstrap.*

Table 4.6. *The performance assessment results of 2-class bootstrap with seven datasets for sentiment analysis, classifier = AFINN.*

Metrics	Comments_NYT	Comments_TED	Comments_YTB	Myspace	Tweets_RND_I	Tweets_RND_III	Tweets_Semeval
Accuracy (%)	57.23 ± 0.72	56.03 ± 2.14	77.25 ± 0.47	85.83 ± 1.38	69.34 ± 1.2	76.26 ± 0.78	79.41 ± 0.69
Precision_Pos (%)	51.07 ± 0.81	49.65 ± 1.89	76.7 ± 0.68	89.85 ± 1.22	66.45 ± 1.43	73.03 ± 1.17	80.28 ± 0.79
Recall_Pos (%)	92.27 ± 0.46	92.94 ± 1.09	96.09 ± 0.42	93.56 ± 0.7	94.33 ± 0.72	95.33 ± 0.87	94.59 ± 0.36
F1_Pos (%)	65.71 ± 0.69	64.57 ± 1.76	85.28 ± 0.35	91.64 ± 0.88	77.92 ± 1.14	82.62 ± 0.74	86.83 ± 0.53
Precision_Neg (%)	82.55 ± 0.75	83.82 ± 2.47	80.37 ± 2.17	58.04 ± 4.23	82.17 ± 1.68	87.72 ± 1.97	74.47 ± 1.58
Recall_Neg (%)	29.2 ± 0.99	28.16 ± 2.58	35.81 ± 1.7	46.46 ± 3.56	35.07 ± 1.27	47.51 ± 2.86	20.49 ± 0.67
F1_Neg (%)	43.05 ± 1.08	41.71 ± 3	49.33 ± 1.86	51.25 ± 3.64	49.05 ± 1.38	61.09 ± 2.42	52.41 ± 0.8
Macro_*F1* (%)	54.38 ± 0.76	53.14 ± 2.27	67.31 ± 0.97	71.44 ± 2.13	63.48 ± 0.98	71.86 ± 1.21	69.62 ± 0.56
Absolute Error	4232	642	1110	238	1404	584	1260

Table 4.7. *The performance assessment results of 2-class bootstrap with seven datasets for sentiment analysis, classifier = RNTN.*

Metrics	Comments_NYT	Comments_TED	Comments_YTB	Myspace	Tweets_RND_I	Tweets_RND_III	Tweets_Semeval
Accuracy (%)	61.27 ± 0.71	66.58 ± 1.93	67.01 ± 1.03	52.74 ± 1.58	53.97 ± 0.89	66.5 ± 0.82	44.71 ± 0.87
Precision_Pos (%)	59.51 ± 1.15	61.02 ± 2.49	82.98 ± 1.29	89.94 ± 1.78	73.72 ± 1.57	84.14 ± 1.16	85.47 ± 0.92
Recall_Pos (%)	40.28 ± 1.11	64.26 ± 2.46	65.41 ± 1.37	49.07 ± 1.77	31.65 ± 0.94	54.04 ± 1.29	27.89 ± 0.86
F1_Pos (%)	47.99 ± 1.05	62.31 ± 2	73.05 ± 1.07	63.29 ± 1.54	44.15 ± 0.89	65.67 ± 0.88	42 ± 1.01
Precision_Neg (%)	62.03 ± 0.84	71.57 ± 2.08	48.3 ± 1.37	21.52 ± 1.6	47.51 ± 1.48	55.33 ± 1.75	32.19 ± 1.08
Recall_Neg (%)	78.1 ± 0.61	68.47 ± 2.6	70.81 ± 1.7	72.54 ± 3.82	84.58 ± 0.99	84.38 ± 1.79	87.86 ± 0.7
F1_Neg (%)	69.12 ± 0.64	69.77 ± 1.98	57.25 ± 1.08	32.83 ± 2.01	60.68 ± 1.26	66.71 ± 1.65	47.03 ± 1.18
Macro_*F1* (%)	58.55 ± 0.76	66.04 ± 1.89	65.1 ± 0.97	48.06 ± 1.41	52.41 ± 0.77	66.19 ± 0.79	44.52 ± 0.87
Absolute Error	3834	488	1610	794	2108	824	3384

Figure 4.3 shows the LS regression model and the diagonal model for the standard errors of performance assessment results, and Table 4.8 provides the hypothesis testing results for Figure 4.3. Figure 4.4 exhibits the same models for the absolute values of performance assessment results, and Table 4.9 provides the hypothesis testing results for Figure 4.4.

Table 4.8. *The hypothesis testing results for Figure 4.3.*

Metrics	Intercept (%)	p-value (Intercept)	Slope (%)	p-value (Slope)	R^2	p-value (Comparison)
SE_Accuracy	0.3904	0.1218	0.6907	0.0116	0.7512	0.8116
SE_Precision (Pos)	0.2414	0.4862	1.0852	0.0095	0.7692	0.1813
SE_Recall (Pos)	0.3875	0.4303	1.5341	0.0623	0.5335	0.0022
SE_F1 (Pos)	0.6488	0.0602	0.6402	0.0683	0.5177	0.1385
SE_Precision (Neg)	0.9969	0.03	0.2169	0.186	0.3195	0.1577
SE_Recall (Neg)	-0.1945	0.6464	0.9957	0.0028	0.8563	0.729
SE_F1 (Neg)	0.5404	0.0643	0.4243	0.0087	0.7767	0.1824
SE_Macro_*F1*	0.3342	0.1176	0.5766	0.0058	0.8088	0.5032
Absolute Error	2.4327	0.1892	0.7069	0.0284	0.6507	0.332

By analyzing the p-value (Comparison) in Table 4.8, these conclusions can be obtained: H_0 cannot be rejected for (a) AFINN-RNTN SE of Accuracy in Figure 4.3; H_0 cannot be rejected for (b) AFINN-RNTN SE of Macro_*F1* in Figure 4.3; H_0 cannot be rejected for (c) AFINN-RNTN SE of Precision_Pos in Figure 4.3; H_0 can be rejected for (d) AFINN-RNTN SE of Recall_Pos in Figure 4.3; H_0 cannot be rejected for (e) AFINN-RNTN SE of F1_Pos in Figure 4.3; H_0 cannot be rejected for (f) AFINN-RNTN SE of Precision_Neg in Figure 4.3; H_0 cannot be rejected for (g) AFINN-RNTN SE of Recall_Neg in Figure 4.3; H_0 cannot be rejected for (h) AFINN-RNTN SE of F1_Neg in Figure 4.3; H_0 cannot be rejected for (i) AFINN-RNTN SE of Absolute Error in Figure 4.3.

By analyzing the p-value (Comparison) in Table 4.9, these conclusions can be obtained: H_0 can be rejected for (a) AFINN-RNTN Accuracy in Figure 4.4; H_0 cannot be rejected

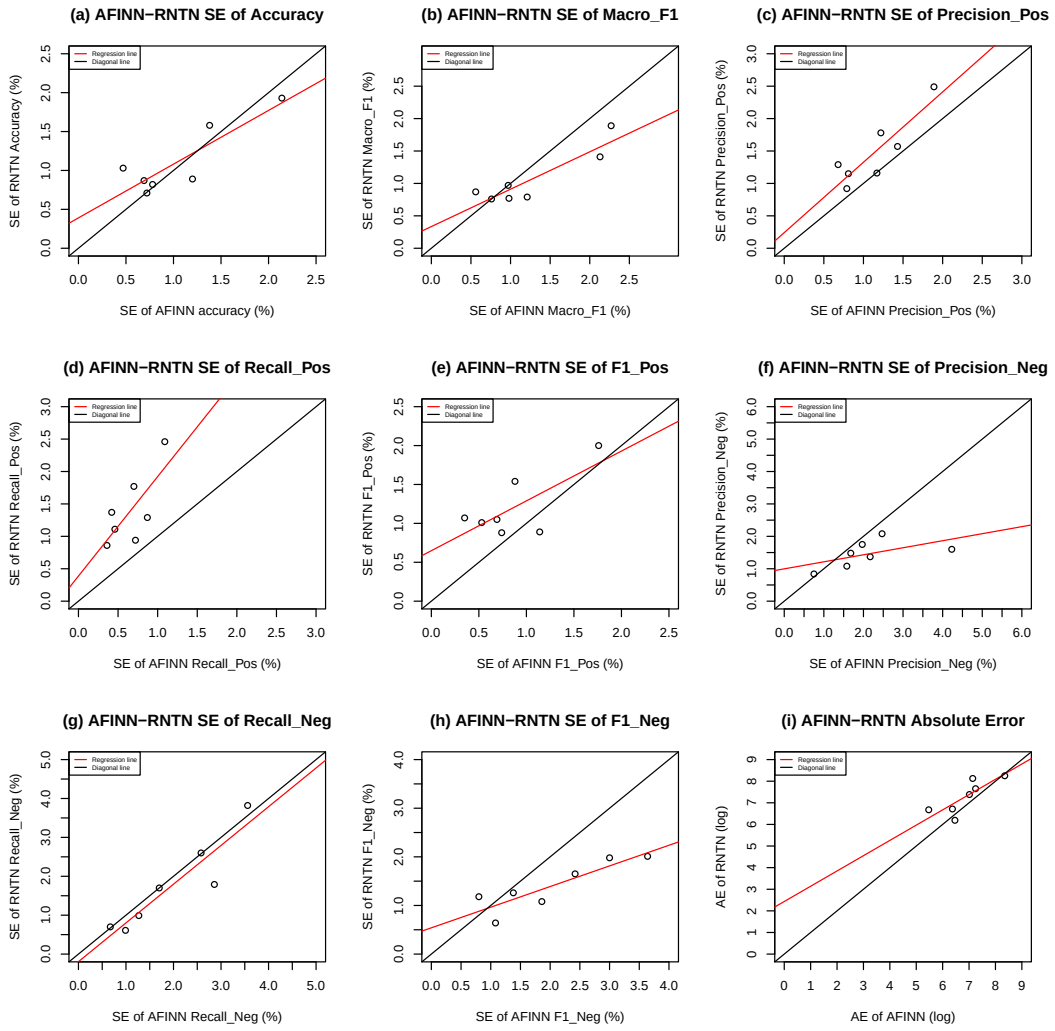

Figure 4.3. *The LS regression model (the red line) and the diagonal model (the black line) for the standard errors of performance assessment results of AFINN and RNTN in 2-class bootstrap.*

for (b) AFINN-RNTN Macro_*F1* in Figure 4.4; H_0 cannot be rejected for (c) AFINN-RNTN Precision_Pos in Figure 4.4; H_0 can be rejected for (d) AFINN-RNTN Recall_Pos in Figure 4.4; H_0 can be rejected for (e) AFINN-RNTN F1_Pos in Figure 4.4; H_0 can be rejected for (f) AFINN-RNTN Precision_Neg in Figure 4.4; H_0 can be rejected for (g) AFINN-RNTN Recall_Neg in Figure 4.4; H_0 can be rejected for (h) AFINN-RNTN F1_Neg in Figure 4.4; H_0 cannot be rejected for (i) AFINN-RNTN Absolute Error in Figure 4.4.

4.1.2 Cross validation

Table 4.10 shows the fundamental errors and their corresponding standard errors for AFINN, and Table 4.11 presents the fundamental errors and their corresponding standard errors for RNTN.

Figure 4.5 and Figure 4.6 exhibit the LS regression model and the diagonal model for

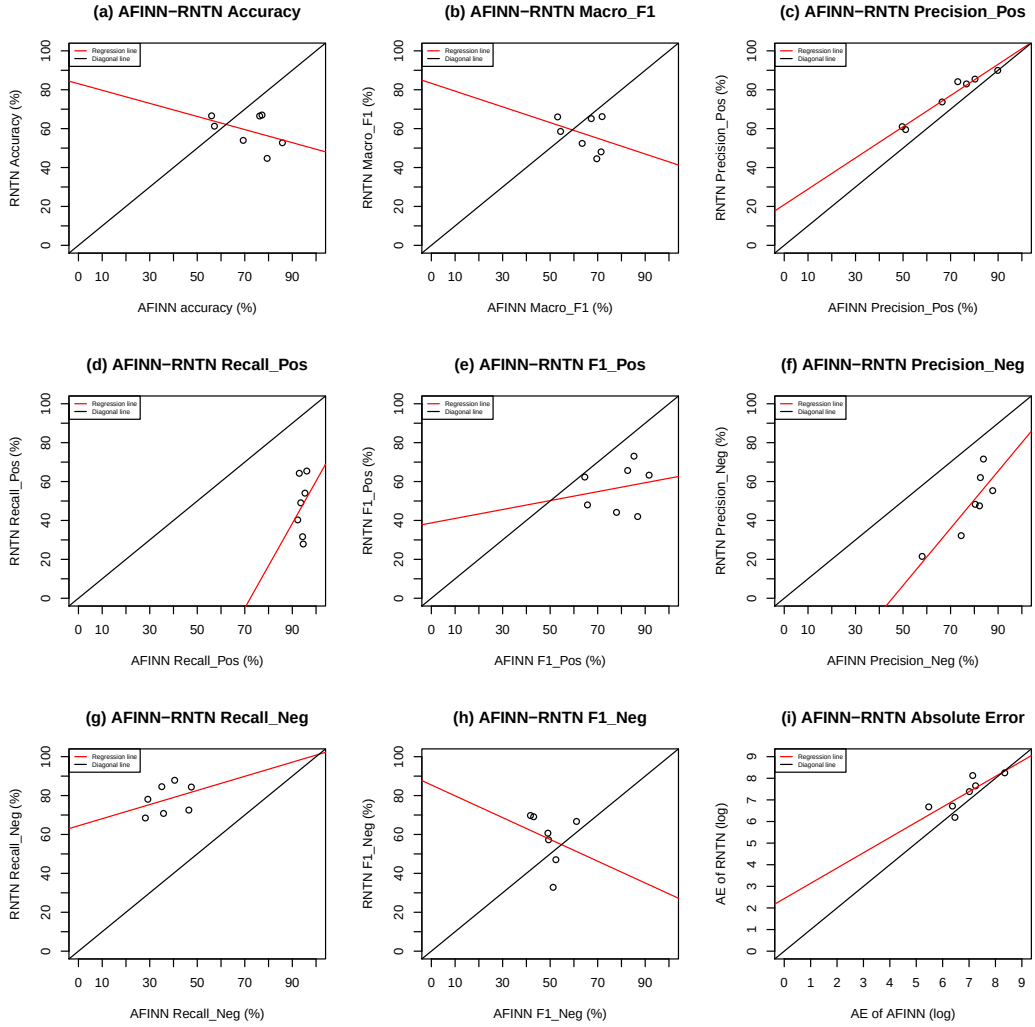

Figure 4.4. *The LS regression model (the red line) and the diagonal model (the black line) for the performance assessment results of AFINN and RNTN in 2-class bootstrap.*

Table 4.9. *The hypothesis testing results for Figure 4.4.*

Metrics	Intercept (%)	p-value (Intercept)	Slope (%)	p-value (Slope)	R^2	p-value (Comparison)
Accuracy	83.0927	0.0135	-0.3368	0.3231	0.1937	0.0253
Precision_Pos	20.8898	0.0108	0.8019	1e-04	0.9586	0.3461
Recall_Pos	-157.4075	0.7465	2.1763	0.675	0.0381	0
F1_Pos	38.7351	0.3797	0.2296	0.6678	0.0399	0.0011
Precision_Neg	-66.8825	0.0971	1.4689	0.0167	0.7142	9e-04
Recall_Neg	64.4688	0.0098	0.3634	0.4224	0.1323	0
F1_Neg	85.4698	0.1185	-0.5602	0.5639	0.0709	0.0184
Macro_*F1*	83.3123	0.045	-0.4039	0.4414	0.1225	0.0551
Absolute Error	2.4327	0.1892	0.7069	0.0284	0.6507	0.332

Table 4.10. *The fundamental errors of 2-class cross-validation with seven datasets for sentiment analysis, classifier = AFINN.*

Metrics	Comments_NYT	Comments_TED	Comments_YTB	Myspace	Tweets_RND_I	Tweets_RND_III	Tweets_Semeval
TPR (%)	42.02 ± 0.37	41.64 ± 1.6	66.6 ± 0.98	80.24 ± 1.17	55.28 ± 1	55.37 ± 1.21	67.84 ± 0.47
TNR (%)	16.51 ± 0.44	12.88 ± 1.57	11.23 ± 0.73	7.26 ± 0.67	15.24 ± 0.74	19.92 ± 0.63	11.9 ± 0.73
FPR (%)	38.12 ± 0.37	42.47 ± 2.03	19.51 ± 0.69	8.45 ± 0.9	25.76 ± 0.58	21.71 ± 1.13	16.86 ± 0.57
FNR (%)	3.35 ± 0.28	3.01 ± 0.79	2.66 ± 0.25	4.05 ± 0.85	3.71 ± 0.36	3.01 ± 0.44	3.4 ± 0.4

Table 4.11. *The fundamental errors of 2-class cross-validation with seven datasets for sentiment analysis, classifier = RNTN.*

Metrics	Comments_NYT	Comments_TED	Comments_YTB	Myspace	Tweets_RND_I	Tweets_RND_III	Tweets_Semeval
TPR (%)	17.72 ± 0.58	26.99 ± 1.6	44.8 ± 0.95	40.48 ± 1.36	19.3 ± 0.68	30.24 ± 1.12	18.95 ± 0.85
TNR (%)	41.49 ± 0.51	38.9 ± 1.29	21.72 ± 1.04	12.38 ± 1.11	35.59 ± 1.21	35.12 ± 1.14	25.29 ± 0.57
FPR (%)	13.13 ± 0.25	16.44 ± 1.14	9.02 ± 0.42	3.33 ± 0.73	5.41 ± 0.73	6.5 ± 0.61	3.46 ± 0.22
FNR (%)	27.66 ± 0.51	17.67 ± 1.16	24.47 ± 0.9	43.81 ± 0.98	39.69 ± 0.46	28.13 ± 1.11	52.29 ± 1.06

the standard errors and absolute values of fundamental errors of AFINN and RNTN, respectively.

Table 4.12 provides the hypothesis testing results for Figure 4.5 and Table 4.13 shows the hypothesis testing results for Figure 4.6.

Table 4.12. *The hypothesis testing results for Figure 4.5.*

Metrics	Intercept (%)	p-value (Intercept)	Slope (%)	p-value (Slope)	R^2	p-value (Comparison)
SE_TPR	0.321	0.1962	0.7196	0.0171	0.7111	0.8125
SE_TNR	0.6036	0.0797	0.48	0.1961	0.3078	0.2323
SE_FPR	0.1445	0.3487	0.4926	0.0149	0.7263	0.2205
SE_FNR	0.5788	0.0587	0.6316	0.217	0.2851	0.0038

By analyzing the p-value (Comparison) in Table 4.12, we can conclude that: H_0 cannot be rejected for (a) AFINN-RNTN SE of TPR in Figure 4.5; H_0 cannot be rejected for (b) AFINN-RNTN SE of TNR in Figure 4.5; H_0 cannot be rejected for (c) AFINN-RNTN SE of FPR in Figure 4.5; H_0 can be rejected for (d) AFINN-RNTN SE of FNR in Figure 4.5.

By analyzing the p-value (Comparison) in Table 4.13, we can conclude that: H_0 can be rejected for (a) AFINN-RNTN TPR in Figure 4.6; H_0 can be rejected for (b) AFINN-RNTN TNR in Figure 4.6; H_0 can be rejected for (c) AFINN-RNTN FPR in Figure 4.6; H_0 can be rejected for (d) AFINN-RNTN FNR in Figure 4.6.

Table 4.14 and Table 4.15 indicate the performance assessment results for AFINN and RNTN, respectively.

Figure 4.7 shows the LS regression model and the diagonal model for the standard errors of performance assessment results, and Table 4.16 provides the hypothesis testing results for Figure 4.7. Figure 4.8 exhibits the same models for the absolute values of

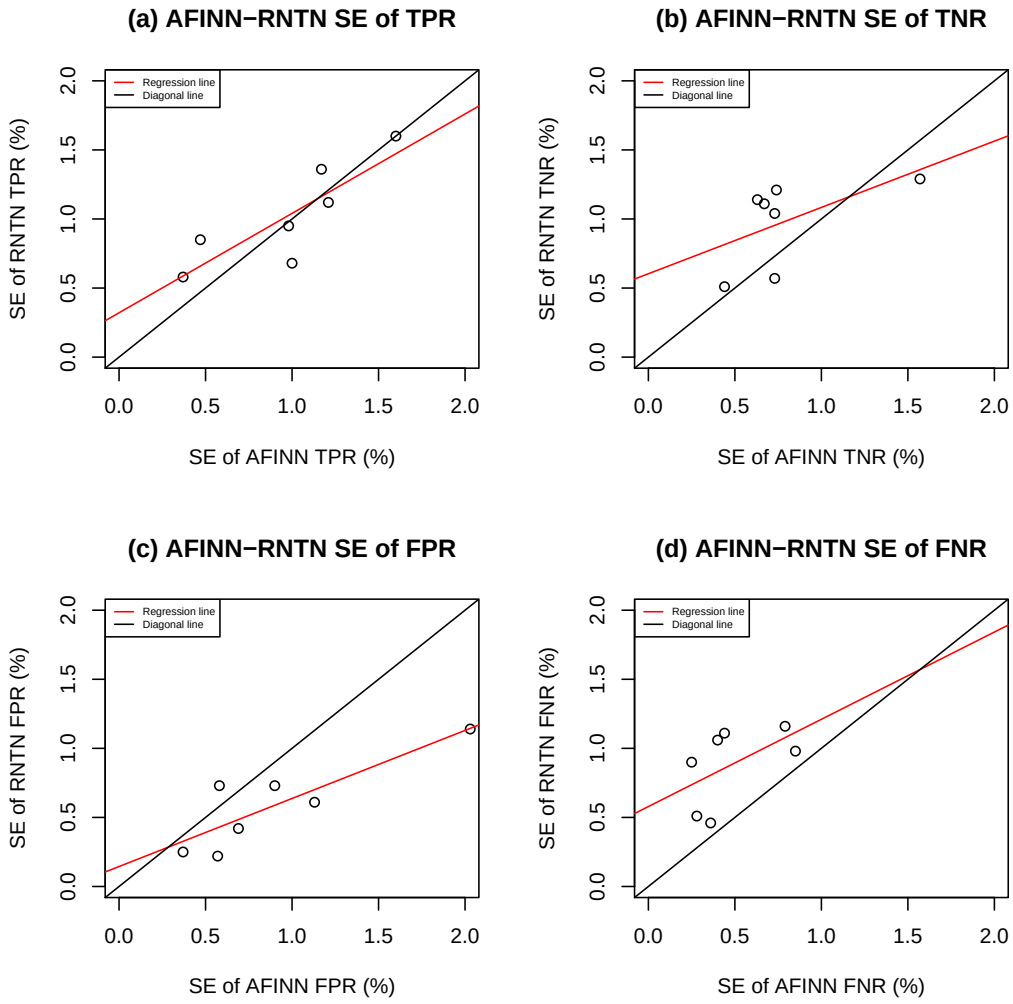

Figure 4.5. *The LS regression model (the red line) and the diagonal model (the black line) for the standard errors of fundamental errors of AFINN and RNTN in 2-class cross-validation.*

Table 4.13. *The hypothesis testing results for Figure 4.6.*

Metrics	Intercept (%)	p-value (Intercept)	Slope (%)	p-value (Slope)	R^2	p-value (Comparison)
TPR	2.5561	0.8849	0.4415	0.1761	0.3316	8e-04
TNR	1.9585	0.8482	2.0727	0.03	0.6433	0.0014
FPR	-1.2168	0.584	0.3807	0.0043	0.8307	0.0097
FNR	-26.3414	0.3901	18.0297	0.0841	0.4806	1e-04

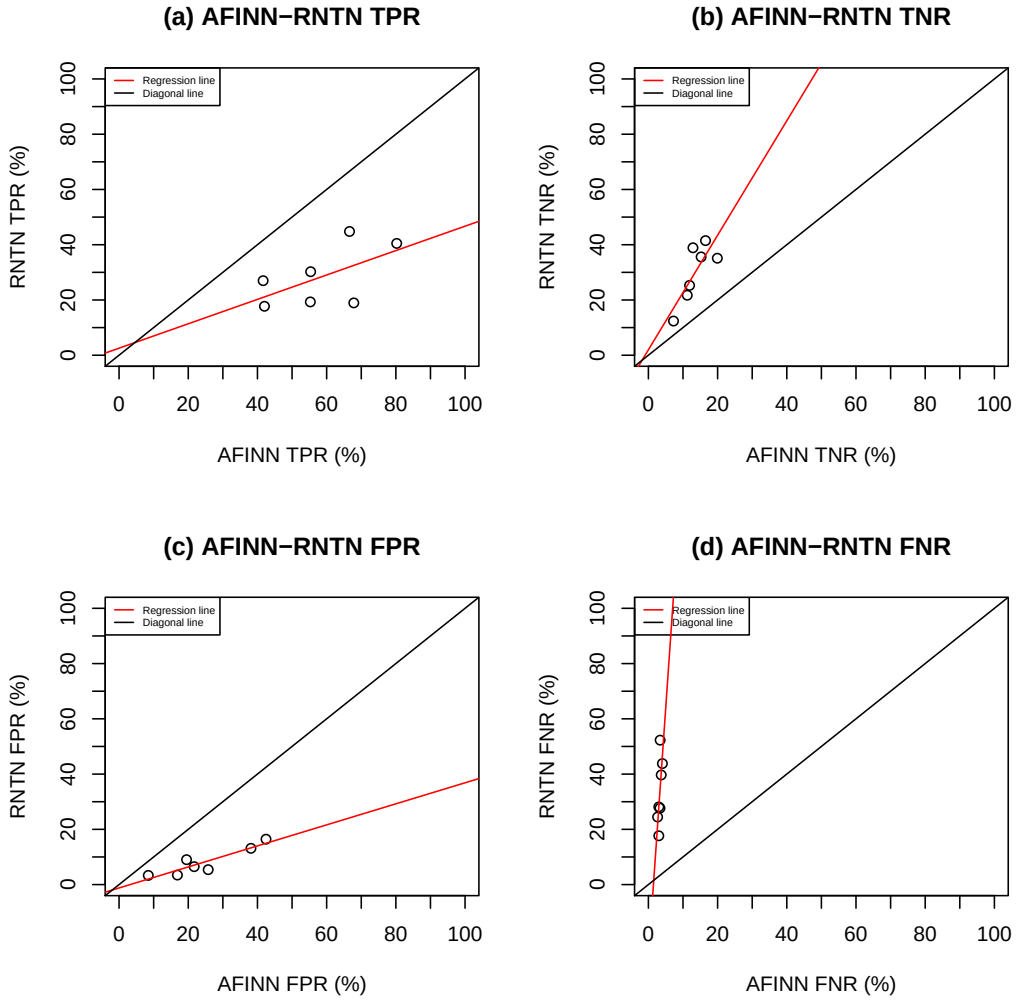

Figure 4.6. *The LS regression model (the red line) and the diagonal model (the black line) for the fundamental errors of AFINN and RNTN in 2-class cross-validation.*

Table 4.14. *The performance assessment results of 2-class cross-validation with seven datasets for sentiment analysis, classifier = AFINN.*

Metrics	Comments_NYT	Comments_TED	Comments_YTB	Myspace	Tweets_RND_I	Tweets_RND_III	Tweets_Semeval
Accuracy (%)	58.53 ± 0.4	54.52 ± 2.31	77.83 ± 0.56	87.5 ± 1.15	70.52 ± 0.78	75.28 ± 1.14	79.74 ± 0.6
Precision_Pos (%)	52.43 ± 0.36	49.62 ± 1.99	77.33 ± 0.83	90.49 ± 0.98	68.18 ± 0.78	71.85 ± 1.41	80.12 ± 0.57
Recall_Pos (%)	92.63 ± 0.59	93.29 ± 1.79	96.18 ± 0.33	95.21 ± 0.99	93.67 ± 0.66	94.83 ± 0.77	95.26 ± 0.5
F1_Pos (%)	66.95 ± 0.34	64.59 ± 1.87	85.7 ± 0.46	92.74 ± 0.7	78.91 ± 0.7	81.68 ± 0.99	87.01 ± 0.35
Precision_Neg (%)	83.16 ± 1.32	79.75 ± 5.81	80.41 ± 2.03	65.79 ± 4.45	80.42 ± 1.55	87.27 ± 1.79	77.5 ± 2.46
Recall_Neg (%)	30.2 ± 0.69	23.34 ± 2.86	36.35 ± 1.55	47.03 ± 4.16	37.07 ± 1.34	48.06 ± 1.71	41.22 ± 2
F1_Neg (%)	44.26 ± 0.83	35.61 ± 3.84	49.92 ± 1.66	53.7 ± 3.82	50.65 ± 1.44	61.73 ± 1.38	53.62 ± 2.15
Macro_*F1* (%)	55.61 ± 0.49	50.1 ± 2.62	67.81 ± 0.85	73.22 ± 2.19	64.78 ± 0.88	71.71 ± 1.11	70.32 ± 1.21
Absolute Error	4106	664	1082	210	1350	608	1240

Table 4.15. *The performance assessment results of 2-class cross-validation with seven datasets for sentiment analysis, classifier = RNTN.*

Metrics	Comments_NYT	Comments_TED	Comments_YTB	Myspace	Tweets_RND_I	Tweets_RND_III	Tweets_Semeval
Accuracy (%)	59.21 ± 0.59	65.89 ± 1.79	66.52 ± 0.82	52.86 ± 1.58	54.89 ± 0.86	65.37 ± 1.25	44.25 ± 1.05
Precision_Pos (%)	57.34 ± 0.91	61.92 ± 2.74	83.24 ± 0.73	92.31 ± 1.68	78.64 ± 2.11	82.31 ± 1.62	84.33 ± 1.32
Recall_Pos (%)	39.02 ± 1.14	60.27 ± 2.61	64.7 ± 1.04	47.97 ± 1.28	32.65 ± 0.82	51.82 ± 1.58	26.62 ± 1.23
F1_Pos (%)	46.41 ± 1.05	60.93 ± 2.44	72.75 ± 0.72	63.09 ± 1.44	46.02 ± 0.9	63.49 ± 1.43	40.37 ± 1.51
Precision_Neg (%)	60.01 ± 0.6	68.8 ± 1.95	46.95 ± 1.72	21.91 ± 1.71	47.17 ± 0.99	55.53 ± 1.5	32.65 ± 0.86
Recall_Neg (%)	75.94 ± 0.51	70.39 ± 1.73	70.5 ± 1.36	78.85 ± 4.71	86.68 ± 1.84	84.35 ± 1.41	87.99 ± 0.68
F1_Neg (%)	67.04 ± 0.51	69.48 ± 1.58	56.24 ± 1.5	34.15 ± 2.38	61.06 ± 1.21	66.87 ± 1.33	47.56 ± 0.93
Macro_*F1* (%)	56.72 ± 0.68	65.2 ± 1.86	64.5 ± 0.93	48.62 ± 1.64	53.54 ± 0.73	65.18 ± 1.24	43.97 ± 1.09
Absolute Error	4038	498	1634	792	2066	852	3412

performance assessment results, and Table 4.17 provides the hypothesis testing results for Figure 4.8.

Table 4.16. *The hypothesis testing results for Figure 4.7.*

Metrics	Intercept (%)	p-value (Intercept)	Slope (%)	p-value (Slope)	R^2	p-value (Comparison)
SE_Accuracy	0.5437	0.0171	0.5957	0.0067	0.7988	0.6274
SE_Precision (Pos)	0.6462	0.1747	0.9518	0.0488	0.573	0.0594
SE_Recall (Pos)	0.5037	0.0707	1.0967	0.0059	0.8078	0.0526
SE_F1 (Pos)	0.6844	0.0385	0.8686	0.0226	0.679	0.0511
SE_Precision (Neg)	0.7405	0.0596	0.2136	0.0761	0.4988	0.0667
SE_Recall (Neg)	-0.3965	0.5671	1.0493	0.0136	0.7356	0.6451
SE_F1 (Neg)	0.5416	0.1653	0.3736	0.0416	0.5973	0.1357
SE_Macro_*F1*	0.4211	0.0077	0.5585	3e-04	0.9375	0.627
Absolute Error	2.6057	0.1693	0.6862	0.0332	0.6295	0.2954

By analyzing the p-value (Comparison) in Table 4.16, these conclusions can be obtained: H_0 cannot be rejected for (a) AFINN-RNTN SE of Accuracy in Figure 4.7; H_0 cannot be rejected for (b) AFINN-RNTN SE of Macro_*F1* in Figure 4.7; H_0 cannot be rejected for (c) AFINN-RNTN SE of Precision_Pos in Figure 4.7; H_0 can be rejected for (d) AFINN-RNTN SE of Recall_Pos in Figure 4.7; H_0 cannot be rejected for (e) AFINN-RNTN SE of F1_Pos in Figure 4.7; H_0 cannot be rejected for (f) AFINN-RNTN SE of Precision_Neg in Figure 4.7; H_0 cannot be rejected for (g) AFINN-RNTN SE of Recall_Neg in Figure 4.7; H_0 cannot be rejected for (h) AFINN-RNTN SE of F1_Neg in Figure 4.7; H_0 cannot be rejected for (i) AFINN-RNTN SE of Absolute Error in Figure 4.7.

By analyzing the p-value (Comparison) in Table 4.17, these conclusions can be obtained: H_0 can be rejected for (a) AFINN-RNTN Accuracy in Figure 4.8; H_0 cannot be rejected for (b) AFINN-RNTN Macro_*F1* in Figure 4.8; H_0 cannot be rejected for (c) AFINN-RNTN Precision_Pos in Figure 4.8; H_0 can be rejected for (d) AFINN-RNTN Recall_Pos in Fig-

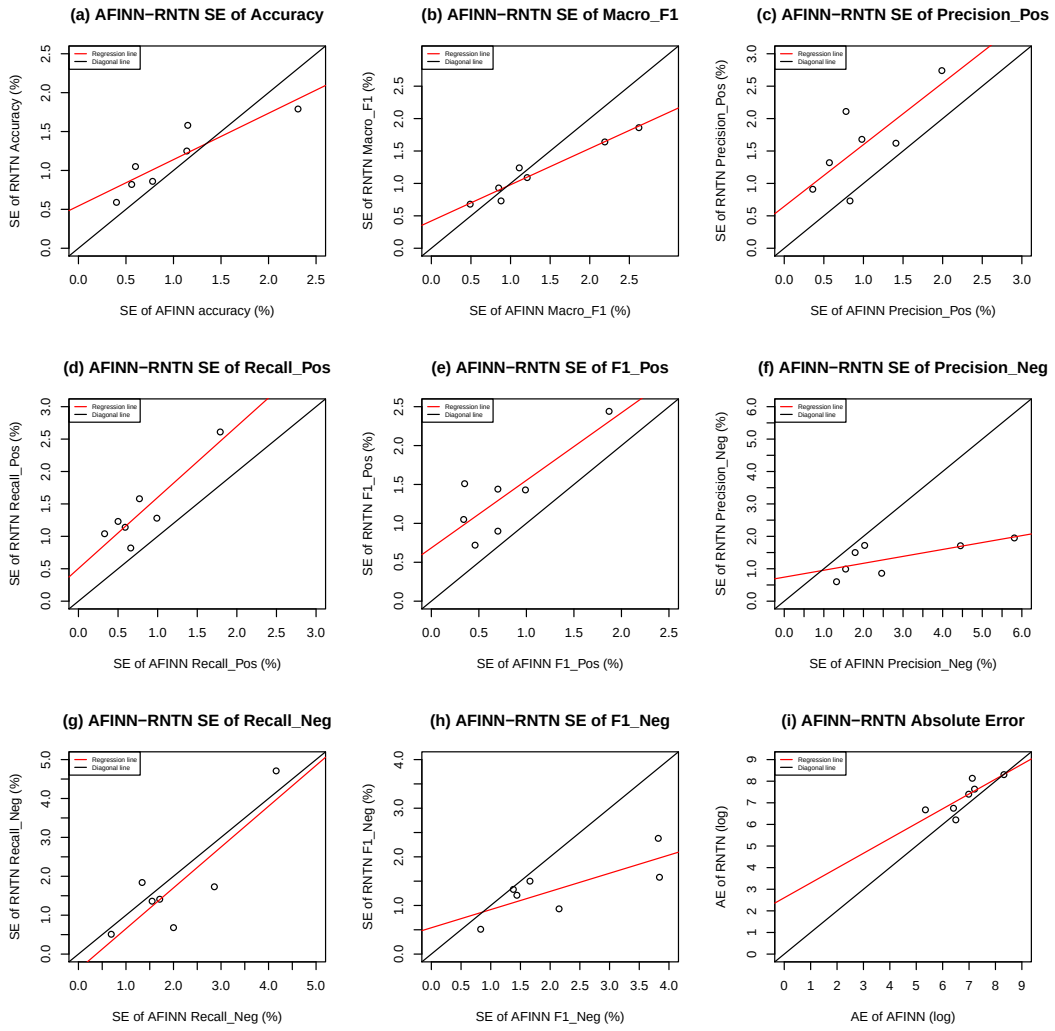

Figure 4.7. *The LS regression model (the red line) and the diagonal model (the black line) for the standard errors of performance assessment results of AFINN and RNTN in 2-class cross-validation.*

ure 4.8; H_0 can be rejected for (e) AFINN-RNTN F1_Pos in Figure 4.8; H_0 can be rejected for (f) AFINN-RNTN Precision_Neg in Figure 4.8; H_0 can be rejected for (g) AFINN-RNTN Recall_Neg in Figure 4.8; H_0 can be rejected for (h) AFINN-RNTN F1_Neg in Figure 4.8; H_0 cannot be rejected for (i) AFINN-RNTN Absolute Error in Figure 4.8.

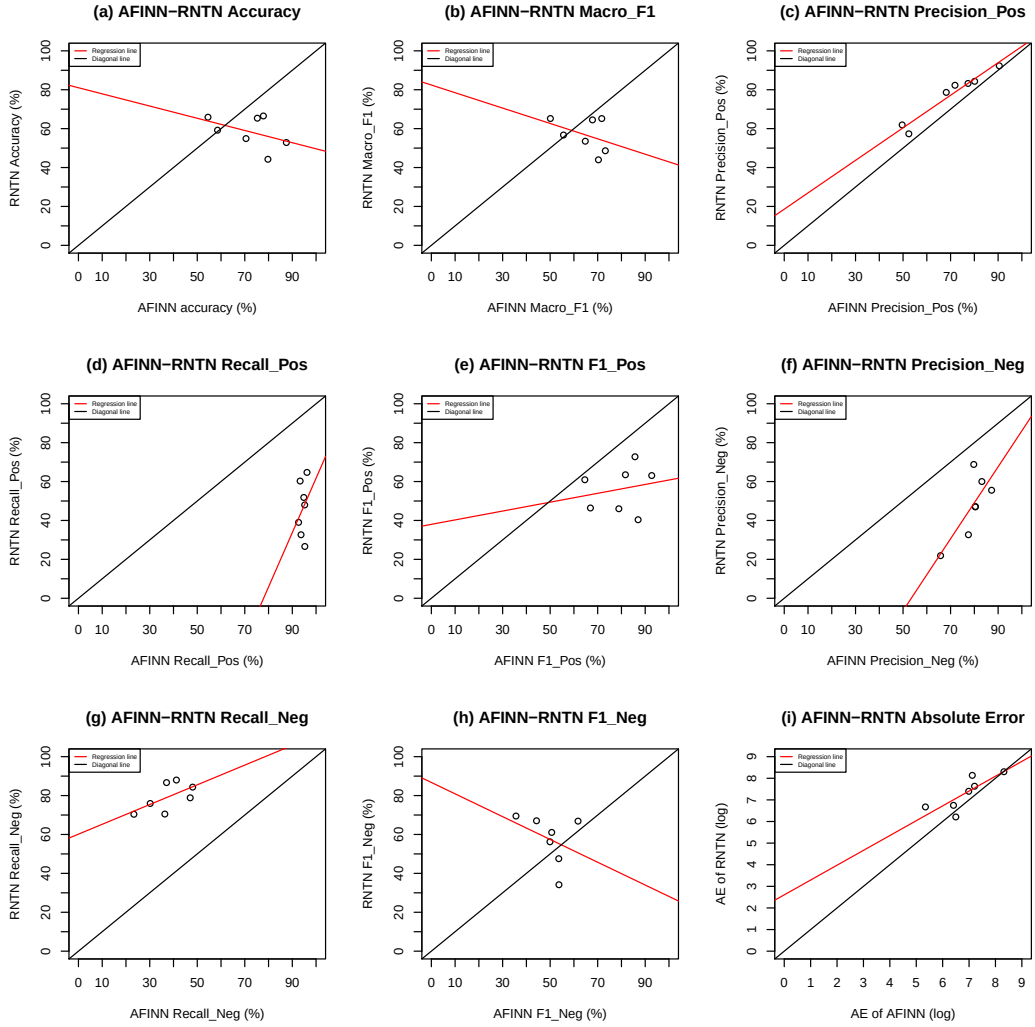

Figure 4.8. *The LS regression model (the red line) and the diagonal model (the black line) for the performance assessment results of AFINN and RNTN in 2-class cross-validation.*

Table 4.17. *The hypothesis testing results for Figure 4.8.*

Metrics	Intercept (%)	p-value (Intercept)	Slope (%)	p-value (Slope)	R^2	p-value (Comparison)
Accuracy	81.0252	0.0109	-0.3139	0.3167	0.1983	0.0223
Precision_Pos	18.5929	0.0386	0.8366	3e-04	0.9412	0.3453
Recall_Pos	-217.8393	0.6523	2.7954	0.5869	0.0631	0
F1_Pos	37.999	0.3855	0.2279	0.6666	0.0402	8e-04
Precision_Neg	-98.8787	0.1317	1.8495	0.0441	0.5886	2e-04
Recall_Neg	60.1702	0.0031	0.5071	0.1442	0.3744	0
F1_Neg	86.743	0.0447	-0.586	0.4052	0.1418	0.0629
Macro_*F1*	82.4256	0.0257	-0.3952	0.3707	0.162	0.0558
Absolute Error	2.6057	0.1693	0.6862	0.0332	0.6295	0.2954

4.2 3-Class comparisons

Different from 2-class comparisons, the datasets used in the 3-class comparisons contain a considerable number of neutral messages. In a word, the results of 3-class comparisons include neutral outputs. Therefore, the comparisons involve all sentences from the datasets namely Comments_NYT, Comments_TED, Comments_YTB, Myspace, Tweets _RND_I, Tweets_RND_III and Tweets_Semeval.

4.2.1 Bootstrap

Table 4.18 shows the fundamental errors and their corresponding standard errors for AFINN, and Table 4.19 presents the fundamental errors and their corresponding standard errors for RNTN.

Table 4.18. *The fundamental errors of 3-class bootstrap with seven datasets for sentiment analysis, classifier = AFINN.*

Metrics	Comments_NYT	Comments_TED	Comments_YTB	Myspace	Tweets_RND_I	Tweets_RND_III	Tweets_Semeval
TPR (%)	14.3 $\pm$ 0.42	24.4 $\pm$ 1.75	28.04 $\pm$ 0.99	41.71 $\pm$ 1.5	18.78 $\pm$ 0.7	11.9 $\pm$ 0.34	22.69 $\pm$ 0.84
TNR (%)	15.14 $\pm$ 0.3	14.76 $\pm$ 1.59	7.6 $\pm$ 0.48	4.1 $\pm$ 0.7	7.95 $\pm$ 0.43	6.59 $\pm$ 0.27	5.91 $\pm$ 0.22
TNeR (%)	3.7 $\pm$ 0.25	8.45 $\pm$ 1.17	17.21 $\pm$ 0.49	12.19 $\pm$ 0.53	29.62 $\pm$ 0.63	45.85 $\pm$ 0.83	31.87 $\pm$ 0.9
FPR (%)	8.54 $\pm$ 0.44	15.24 $\pm$ 1.09	14.34 $\pm$ 0.69	6.86 $\pm$ 0.6	16.49 $\pm$ 0.59	19.68 $\pm$ 0.75	14.43 $\pm$ 0.57
FNR (%)	4.07 $\pm$ 0.22	4.64 $\pm$ 0.52	5.07 $\pm$ 0.36	6.19 $\pm$ 0.84	6.38 $\pm$ 0.39	5.77 $\pm$ 0.38	7.36 $\pm$ 0.46
FNeR (%)	54.26 $\pm$ 0.56	32.5 $\pm$ 1.11	27.74 $\pm$ 0.74	28.95 $\pm$ 1.04	20.78 $\pm$ 0.59	10.21 $\pm$ 0.44	17.73 $\pm$ 0.56

Table 4.19. *The fundamental errors of 3-class bootstrap with seven datasets for sentiment analysis, classifier = RNTN.*

Metrics	Comments_NYT	Comments_TED	Comments_YTB	Myspace	Tweets_RND_I	Tweets_RND_III	Tweets_Semeval
TPR (%)	8.73 $\pm$ 0.46	18.81 $\pm$ 1.74	21.58 $\pm$ 0.6	19.14 $\pm$ 0.87	7.44 $\pm$ 0.33	7.28 $\pm$ 0.48	6.03 $\pm$ 0.21
TNR (%)	40.12 $\pm$ 1.02	37.62 $\pm$ 1.97	15.92 $\pm$ 0.77	9.62 $\pm$ 1.29	20.02 $\pm$ 0.74	10.53 $\pm$ 0.52	12.22 $\pm$ 0.33
TNeR (%)	1.87 $\pm$ 0.22	2.5 $\pm$ 0.72	7.8 $\pm$ 0.4	6.48 $\pm$ 0.85	3.44 $\pm$ 0.3	14.07 $\pm$ 0.68	3.32 $\pm$ 0.27
FPR (%)	3.22 $\pm$ 0.3	4.76 $\pm$ 0.89	5.13 $\pm$ 0.4	2.19 $\pm$ 0.45	4.21 $\pm$ 0.37	4.07 $\pm$ 0.21	2.68 $\pm$ 0.15
FNR (%)	28.34 $\pm$ 0.76	22.38 $\pm$ 1.45	33.96 $\pm$ 0.82	47.24 $\pm$ 1.85	60.56 $\pm$ 0.92	59.5 $\pm$ 1.06	71.4 $\pm$ 0.46
FNeR (%)	17.73 $\pm$ 0.59	13.93 $\pm$ 0.97	15.6 $\pm$ 0.68	15.33 $\pm$ 1.02	4.33 $\pm$ 0.31	4.55 $\pm$ 0.36	4.37 $\pm$ 0.35

Figure 4.9 and Figure 4.10 exhibit the LS regression model and the diagonal model for the standard errors and absolute values of fundamental errors of AFINN and RNTN, respectively.

Table 4.20 provides the hypothesis testing results for Figure 4.9 and Table 4.21 shows the hypothesis testing results for Figure 4.10.

Considering the p-value (Comparison) in Table 4.20, we can conclude that: H_0 cannot be rejected for (a) AFINN-RNTN SE of TPR in Figure 4.9; H_0 cannot be rejected for (b) AFINN-RNTN SE of TNR in Figure 4.9; H_0 cannot be rejected for (c) AFINN-RNTN SE of TNeR in Figure 4.9; H_0 can be rejected for (d) AFINN-RNTN SE of FPR in Figure 4.9;

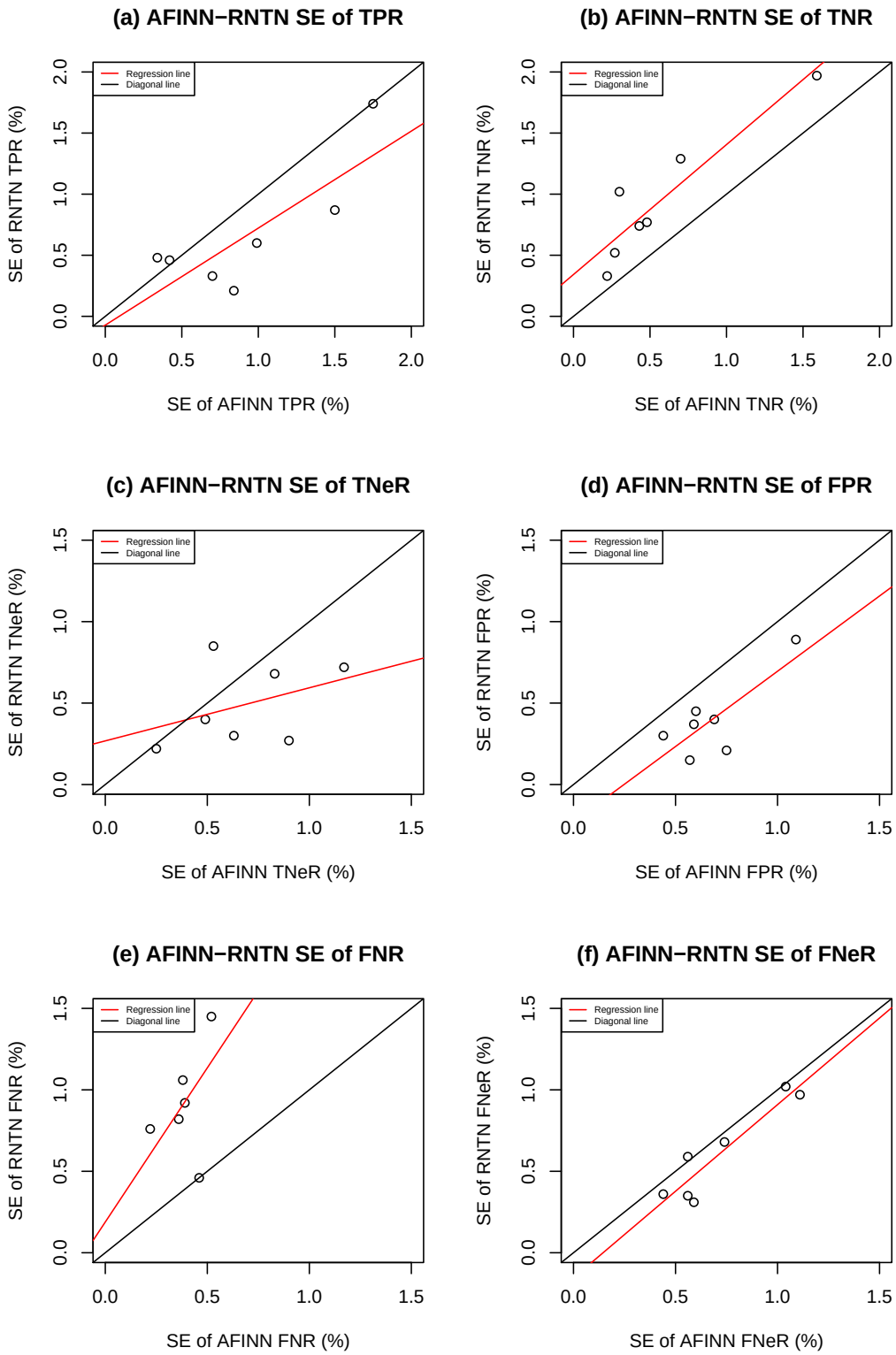

Figure 4.9. *The LS regression model (the red line) and the diagonal model (the black line) for the standard errors of fundamental errors of AFINN and RNTN in 3-class bootstrap.*

Table 4.20. *The hypothesis testing results for Figure 4.9.*

Metrics	Intercept (%)	p-value (Intercept)	Slope (%)	p-value (Slope)	R^2	p-value (Comparison)
SE_TPR	-0.0718	0.8002	0.794	0.0266	0.6592	0.3208
SE_TNR	0.3424	0.0579	1.0634	0.0027	0.8578	0.1767
SE_TNeR	0.268	0.3385	0.3259	0.384	0.1539	0.1511
SE_FPR	-0.2288	0.3565	0.9242	0.0345	0.6241	0.0222
SE_FNR	0.1888	0.5817	1.8923	0.0348	0.6231	0.0043
SE_FNeR	-0.1541	0.3241	1.0633	0.0023	0.8676	0.4615

H_0 can be rejected for (e) AFINN-RNTN SE of FNR in Figure 4.9; H_0 cannot be rejected for (f) AFINN-RNTN SE of FNeR in Figure 4.9.

Table 4.21. *The hypothesis testing results for Figure 4.10.*

Metrics	Intercept (%)	p-value (Intercept)	Slope (%)	p-value (Slope)	R^2	p-value (Comparison)
TPR	1.3249	0.8097	0.4927	0.0656	0.5246	0.0357
TNR	-4.9245	0.0674	2.9093	0	0.9729	0.0473
TNeR	1.657	0.529	0.1873	0.1097	0.4301	0.0332
FPR	1.6781	0.2242	0.1518	0.1334	0.3906	0.001
FNR	-39.0549	0.0643	15.1156	0.0033	0.8465	7e-04
FNeR	1.012	0.7752	0.3578	0.023	0.677	0.0198

Considering the p-value (Comparison) in Table 4.21, we can conclude that: H_0 can be rejected for (a) AFINN-RNTN TPR in Figure 4.10; H_0 can be rejected for (b) AFINN-RNTN TNR in Figure 4.10; H_0 can be rejected for (c) AFINN-RNTN TNeR in Figure 4.10; H_0 can be rejected for (d) AFINN-RNTN FPR in Figure 4.10; H_0 can be rejected for (e) AFINN-RNTN FNR in Figure 4.10; H_0 can be rejected for (f) AFINN-RNTN FNeR in Figure 4.10.

Table 4.22 and Table 4.23 indicate the performance assessment results for AFINN and RNTN, respectively.

Figure 4.11 shows the LS regression model and the diagonal model for the standard errors of performance assessment results, and Table 4.24 provides the hypothesis testing results for Figure 4.11. Figure 4.12 exhibits the same models for the absolute values of performance assessment results, and Table 4.25 provides the hypothesis testing results for Figure 4.12.

Considering the p-value (Comparison) in Table 4.24, these conclusions can be obtained:

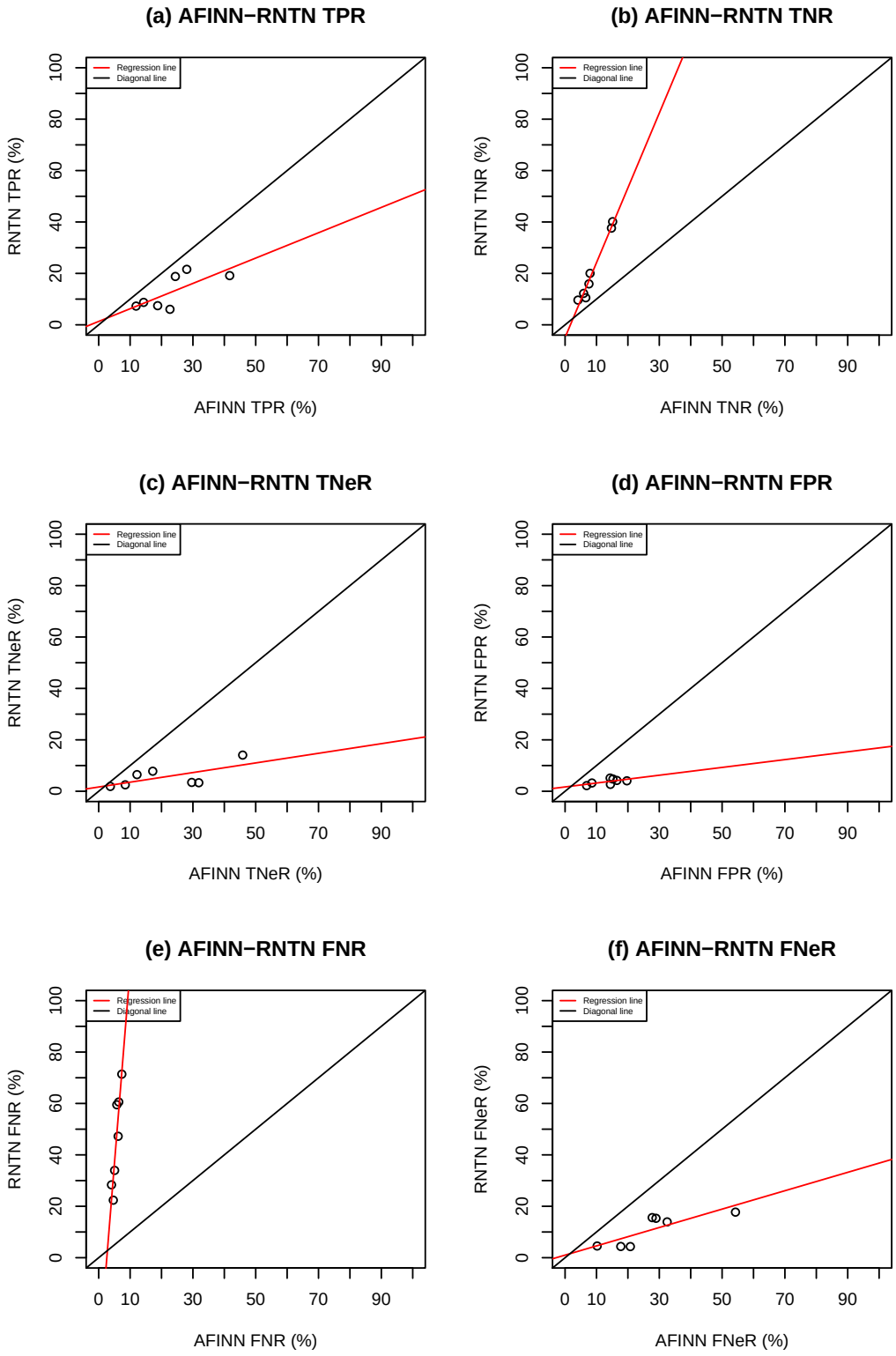

Figure 4.10. *The LS regression model (the red line) and the diagonal model (the black line) for the fundamental errors of AFINN and RNTN in 3-class bootstrap.*

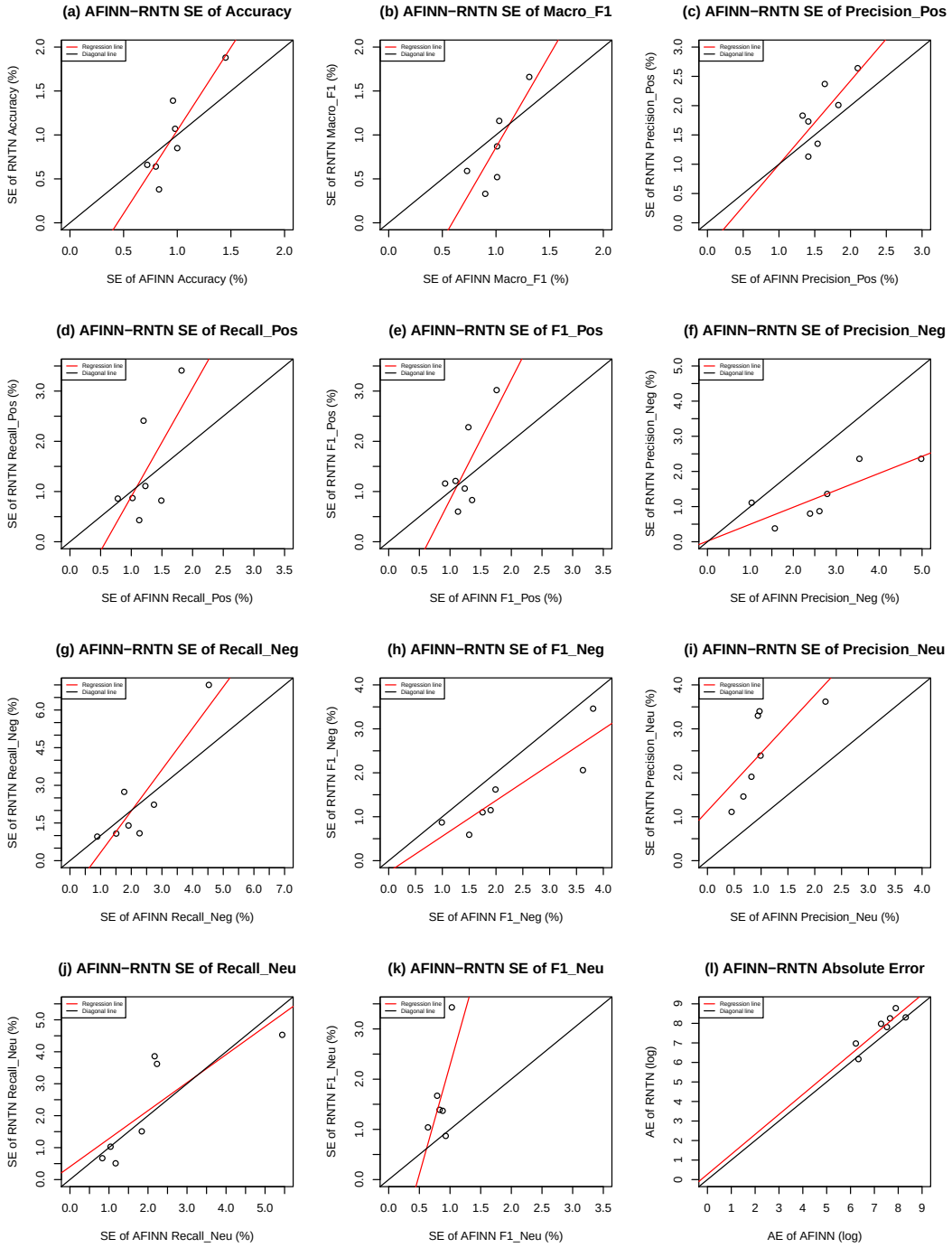

Figure 4.11. *The LS regression model (the red line) and the diagonal model (the black line) for the standard errors of performance assessment results of AFINN and RNTN in 3-class bootstrap.*

Table 4.22. *The performance assessment results of 3-class bootstrap with seven datasets for sentiment analysis, classifier = AFINN.*

Metrics	Comments_NYT	Comments_TED	Comments_YTB	Myspace	Tweets_RND_I	Tweets_RND_III	Tweets_Semeval
Accuracy (%)	33.14 ± 0.72	47.62 ± 1.45	52.84 ± 1	58 ± 0.96	56.35 ± 0.8	64.34 ± 0.98	60.48 ± 0.83
Precision_Pos (%)	62.65 ± 1.83	61.32 ± 2.1	66.11 ± 1.54	85.74 ± 1.33	53.19 ± 1.64	37.85 ± 1.41	61.06 ± 1.41
Recall_Pos (%)	33.32 ± 0.78	67.63 ± 1.82	59.13 ± 1.49	61 ± 1.23	61.54 ± 1.02	63.93 ± 1.2	60.96 ± 1.13
F1_Pos (%)	43.42 ± 0.92	64.2 ± 1.76	62.35 ± 1.36	71.22 ± 1.09	56.98 ± 1.24	47.45 ± 1.3	60.96 ± 1.13
Precision_Neg (%)	78.86 ± 1.03	73.78 ± 4.98	59.77 ± 2.79	38.55 ± 3.54	55.49 ± 2.61	53.55 ± 2.39	44.9 ± 1.57
Recall_Neg (%)	29.01 ± 0.89	29.19 ± 2.74	32.53 ± 1.77	32.67 ± 4.53	34.42 ± 1.51	52.04 ± 2.27	42.97 ± 1.91
F1_Neg (%)	42.35 ± 0.99	41.64 ± 3.62	41.93 ± 1.99	34.04 ± 3.81	42.38 ± 1.75	52.52 ± 1.9	43.73 ± 1.5
Precision_Neu (%)	6.4 ± 0.45	20.21 ± 2.2	38.3 ± 0.67	29.66 ± 0.94	58.79 ± 0.99	81.76 ± 0.82	64.22 ± 0.97
Recall_Neu (%)	79.82 ± 2.23	59.59 ± 5.44	58.9 ± 1.84	63.49 ± 2.17	63.87 ± 0.83	66.83 ± 1.04	64.97 ± 1.17
F1_Neu (%)	11.82 ± 0.79	30.05 ± 3.08	46.33 ± 0.88	40.27 ± 1.03	61.16 ± 0.64	73.51 ± 0.83	64.55 ± 0.93
Macro_*F1* (%)	32.53 ± 0.73	45.3 ± 1.94	50.2 ± 1.01	48.51 ± 1.31	53.51 ± 1.01	57.83 ± 1.03	56.41 ± 0.9
Absolute Error	4076	562	1858	504	2113	1450	2689

Table 4.23. *The performance assessment results of 3-class bootstrap with seven datasets for sentiment analysis, classifier = RNTN.*

Metrics	Comments_NYT	Comments_TED	Comments_YTB	Myspace	Tweets_RND_I	Tweets_RND_III	Tweets_Semeval
Accuracy (%)	50.71 ± 0.66	58.93 ± 1.88	45.31 ± 0.85	35.24 ± 1.39	30.89 ± 0.64	31.88 ± 1.07	21.56 ± 0.38
Precision_Pos (%)	73.11 ± 2.01	79.78 ± 2.64	80.83 ± 1.35	90.31 ± 1.83	64.17 ± 2.37	63.68 ± 1.73	69.3 ± 1.13
Recall_Pos (%)	20.27 ± 0.86	52.16 ± 3.41	45.53 ± 0.82	28.03 ± 1.11	24.4 ± 0.87	38.99 ± 2.41	16.24 ± 0.43
F1_Pos (%)	31.67 ± 1.16	62.48 ± 3.02	58.2 ± 0.83	42.6 ± 1.21	35.23 ± 1.06	48.14 ± 2.28	26.28 ± 0.6
Precision_Neg (%)	58.57 ± 1.11	62.56 ± 2.36	31.88 ± 1.36	16.95 ± 2.36	24.84 ± 0.87	15.06 ± 0.8	14.61 ± 0.38
Recall_Neg (%)	76.48 ± 0.96	75.04 ± 2.23	68.26 ± 2.74	74.03 ± 7	86.56 ± 1.08	82.29 ± 1.09	88.33 ± 1.4
F1_Neg (%)	66.28 ± 0.87	68.06 ± 2.06	43.34 ± 1.62	27.24 ± 3.46	38.55 ± 1.1	25.4 ± 1.15	25.05 ± 0.59
Precision_Neu (%)	9.59 ± 1.11	14.4 ± 3.62	33.38 ± 1.46	29.54 ± 3.3	44.08 ± 2.39	75.4 ± 1.91	43.39 ± 3.4
Recall_Neu (%)	40.36 ± 3.62	17.2 ± 4.53	26.73 ± 1.51	33.52 ± 3.86	7.43 ± 0.67	20.54 ± 1.03	6.75 ± 0.51
F1_Neu (%)	15.39 ± 1.67	0	29.56 ± 1.37	31.18 ± 3.43	12.65 ± 1.04	32.21 ± 1.39	11.66 ± 0.87
Macro_*F1* (%)	37.78 ± 0.59	0	43.7 ± 0.87	33.68 ± 1.66	28.81 ± 0.52	35.25 ± 1.16	21 ± 0.33
Absolute Error	4053	478	2465	1065	3863	2917	6504

H_0 cannot be rejected for (a) AFINN-RNTN SE of Accuracy in Figure 4.11; H_0 cannot be rejected for (b) AFINN-RNTN SE of Macro_*F1* in Figure 4.11; H_0 cannot be rejected for (c) AFINN-RNTN SE of Precision_Pos in Figure 4.11; H_0 cannot be rejected for (d) AFINN-RNTN SE of Recall_Pos in Figure 4.11; H_0 cannot be rejected for (e) AFINN-RNTN SE of F1_Pos in Figure 4.11; H_0 can be rejected for (f) AFINN-RNTN SE of Precision_Neg in Figure 4.11; H_0 cannot be rejected for (g) AFINN-RNTN SE of Recall_Neg in Figure 4.11; H_0 cannot be rejected for (h) AFINN-RNTN SE of F1_Neg in Figure 4.11; H_0 can be rejected for (i) AFINN-RNTN SE of Precision_Neu in Figure 4.11; H_0 cannot be rejected for (j) AFINN-RNTN SE of Recall_Neu in Figure 4.11; H_0 cannot be rejected for (k) AFINN-RNTN SE of F1_Neu in Figure 4.11; H_0 cannot be rejected for (l) AFINN-RNTN SE of Absolute Error in Figure 4.11.

Considering the p-value (Comparison) in Table 4.25, these conclusions can be obtained: H_0 can be rejected for (a) AFINN-RNTN Accuracy in Figure 4.12; H_0 can be rejected for (b) AFINN-RNTN Macro_*F1* in Figure 4.12; H_0 cannot be rejected for (c) AFINN-RNTN Precision_Pos in Figure 4.12; H_0 can be rejected for (d) AFINN-RNTN Recall_Pos in Figure 4.12; H_0 can be rejected for (e) AFINN-RNTN F1_Pos in Figure 4.12; H_0 can be

Table 4.24. *The hypothesis testing results for Figure 4.11.*

Metrics	Intercept (%)	p-value (Intercept)	Slope (%)	p-value (Slope)	R^2	p-value (Comparison)
SE_Accuracy	-0.8387	0.1228	1.8904	0.0091	0.7733	0.9254
SE_Precision (Pos)	-0.4321	0.6698	1.4285	0.0589	0.5429	0.1838
SE_Recall (Pos)	-1.2636	0.3971	2.1632	0.099	0.4502	0.5723
SE_F1 (Pos)	-1.5595	0.2787	2.3951	0.0624	0.5332	0.4772
SE_Precision (Neg)	0.017	0.972	0.4823	0.0268	0.6579	0.0327
SE_Recall (Neg)	-1.3216	0.209	1.6486	0.0067	0.799	0.8855
SE_F1 (Neg)	-0.2558	0.5833	0.8124	0.0062	0.8041	0.2227
SE_Precision (Neu)	1.1317	0.1254	1.3165	0.0601	0.5396	0.0016
SE_Recall (Neu)	0.4035	0.5974	0.8767	0.0259	0.6624	0.8583
SE_F1 (Neu)	-2.0669	0.4289	4.3474	0.1875	0.3866	0.2419
SE_Macro_*F1*	-1.2491	0.1731	2.1076	0.0473	0.667	0.9874
Absolute Error	0.2721	0.8786	1.0223	0.0068	0.7978	0.3224

Table 4.25. *The hypothesis testing results for Figure 4.12.*

Metrics	Intercept (%)	p-value (Intercept)	Slope (%)	p-value (Slope)	R^2	p-value (Comparison)
Accuracy	88.7165	0.0072	-0.9295	0.0559	0.5514	0.0222
Precision_Pos	37.2278	0.0068	0.609	0.0062	0.8044	0.0627
Recall_Pos	-2.2141	0.9389	0.5917	0.2587	0.2451	4e-04
F1_Pos	17.7759	0.6314	0.4431	0.488	0.1007	0.0063
Precision_Neg	-43.5029	0.0266	1.3065	0.0026	0.86	0.0153
Recall_Neg	61.4001	0.0035	0.4793	0.1945	0.3097	0
F1_Neg	63.6817	0.3744	-0.5086	0.7517	0.0219	0.7771
Precision_Neu	1.8747	0.7247	0.7906	6e-04	0.9233	0.592
Recall_Neu	-37.8446	0.4487	0.9125	0.25	0.2529	0
F1_Neu	19.6662	0.1714	0.0492	0.8343	0.0123	0.0224
Macro_*F1*	51.7927	0.0549	-0.3697	0.3876	0.19	0.0022
Absolute Error	0.2721	0.8786	1.0223	0.0068	0.7978	0.3224

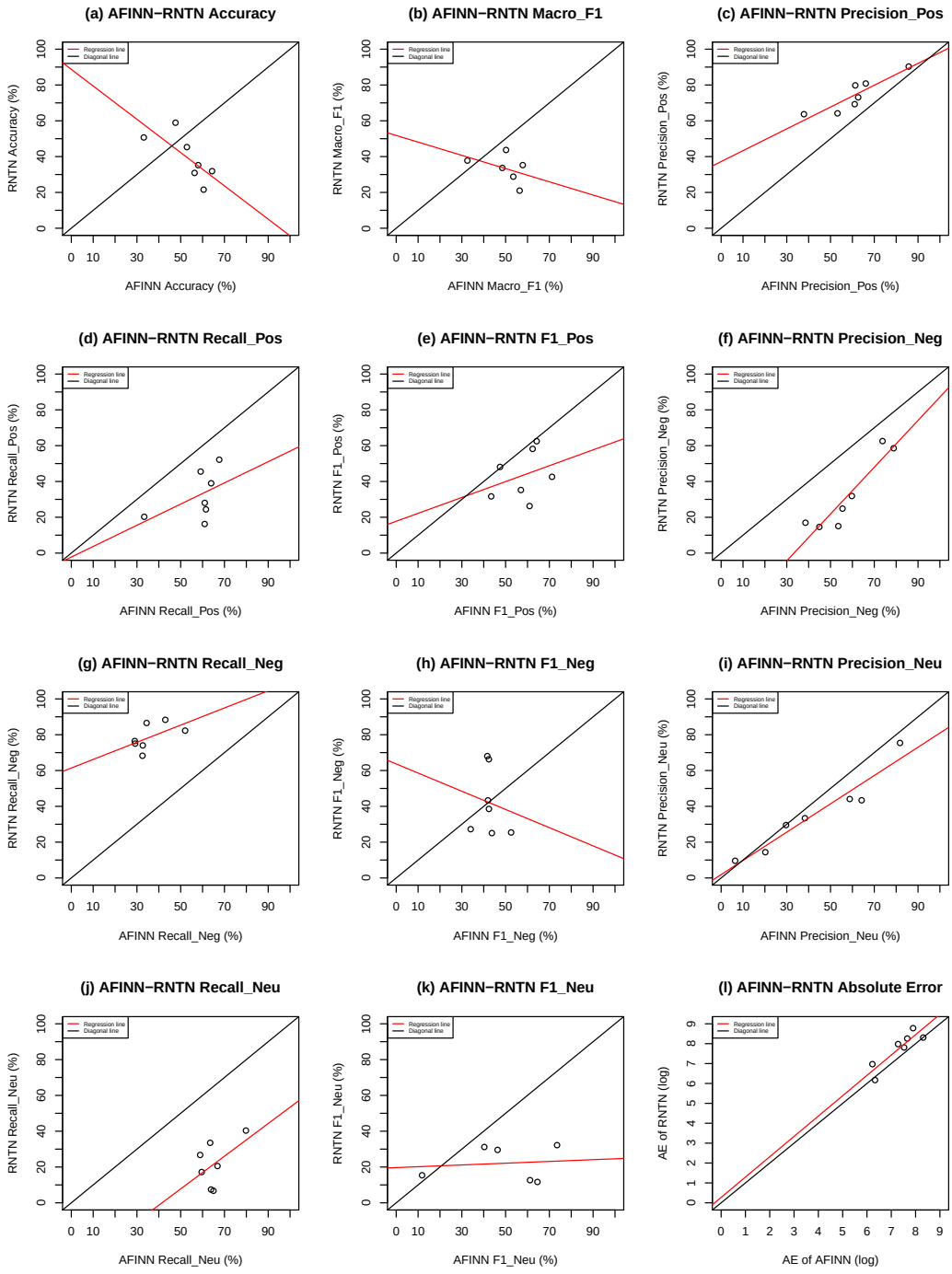

Figure 4.12. *The LS regression model (the red line) and the diagonal model (the black line) for the performance assessment results of AFINN and RNTN in 3-class bootstrap.*

rejected for (f) AFINN-RNTN Precision_Neg in Figure 4.12; H_0 can be rejected for (g) AFINN-RNTN Recall_Neg in Figure 4.12; H_0 cannot be rejected for (h) AFINN-RNTN F1_Neg in Figure 4.12; H_0 cannot be rejected for (i) AFINN-RNTN Precision_Neu in Figure 4.12; H_0 can be rejected for (j) AFINN-RNTN Recall_Neu in Figure 4.12; H_0 can be rejected for (k) AFINN-RNTN F1_Neu in Figure 4.12; H_0 cannot be rejected for (l) AFINN-RNTN Absolute Error in Figure 4.12.

4.2.2 Cross validation

Table 4.26 shows the fundamental errors and their corresponding standard errors for AFINN, and Table 4.27 presents the fundamental errors and their corresponding standard errors for RNTN.

Table 4.26. *The fundamental errors of 3-class cross-validation with seven datasets for sentiment analysis, classifier = AFINN.*

Metrics	Comments_NYT	Comments_TED	Comments_YTB	Myspace	Tweets_RND_I	Tweets_RND_III	Tweets_Semeval
TPR (%)	14.3 ± 0.42	24.4 ± 1.75	28.04 ± 0.99	41.71 ± 1.5	18.78 ± 0.7	11.9 ± 0.34	22.69 ± 0.84
TNR (%)	15.14 ± 0.3	14.76 ± 1.59	7.6 ± 0.48	4.1 ± 0.7	7.95 ± 0.43	6.59 ± 0.27	5.91 ± 0.22
TNeR (%)	3.7 ± 0.25	8.45 ± 1.17	17.21 ± 0.49	12.19 ± 0.53	29.62 ± 0.63	45.85 ± 0.83	31.87 ± 0.9
FPR (%)	8.54 ± 0.44	15.24 ± 1.09	14.34 ± 0.69	6.86 ± 0.6	16.49 ± 0.59	19.68 ± 0.75	14.43 ± 0.57
FNR (%)	4.07 ± 0.22	4.64 ± 0.52	5.07 ± 0.36	6.19 ± 0.84	6.38 ± 0.39	5.77 ± 0.38	7.36 ± 0.46
FNeR (%)	54.26 ± 0.56	32.5 ± 1.11	27.74 ± 0.74	28.95 ± 1.04	20.78 ± 0.59	10.21 ± 0.44	17.73 ± 0.56

Table 4.27. *The fundamental errors of 3-class cross-validation with seven datasets for sentiment analysis, classifier = RNTN.*

Metrics	Comments_NYT	Comments_TED	Comments_YTB	Myspace	Tweets_RND_I	Tweets_RND_III	Tweets_Semeval
TPR (%)	8.98 ± 0.37	19.22 ± 0.85	23.07 ± 0.53	19.19 ± 1.17	7.42 ± 0.62	7.54 ± 0.71	6.64 ± 0.39
TNR (%)	40.91 ± 0.57	34.08 ± 1.22	15.37 ± 1.02	9.62 ± 0.69	19.23 ± 1.22	10.77 ± 0.64	12.07 ± 0.83
TNeR (%)	1.79 ± 0.25	2.12 ± 0.74	8.14 ± 0.6	6.84 ± 0.73	4.03 ± 0.39	13.39 ± 0.51	3.54 ± 0.28
FPR (%)	3.25 ± 0.24	6.02 ± 0.55	5.12 ± 0.29	2.18 ± 0.42	3.9 ± 0.25	4.18 ± 0.19	2.56 ± 0.29
FNR (%)	27.95 ± 0.73	22.99 ± 1.43	32.93 ± 0.75	47.54 ± 0.84	60.91 ± 0.89	59.55 ± 0.86	70.85 ± 1.09
FNeR (%)	17.12 ± 0.56	15.57 ± 0.99	15.37 ± 0.51	14.64 ± 1.26	4.51 ± 0.45	4.58 ± 0.28	4.33 ± 0.26

Figure 4.13 and Figure 4.14 exhibit the LS regression model and the diagonal model for the standard errors and absolute values of fundamental errors of AFINN and RNTN, respectively.

Table 4.28 provides the hypothesis testing results for Figure 4.13 and Table 4.29 shows the hypothesis testing results for Figure 4.14.

Considering the p-value (Comparison) in Table 4.28, we can conclude that: H_0 cannot be rejected for (a) AFINN-RNTN SE of TPR in Figure 4.13; H_0 cannot be rejected for (b) AFINN-RNTN SE of TNR in Figure 4.13; H_0 cannot be rejected for (c) AFINN-RNTN SE of TNeR in Figure 4.13; H_0 can be rejected for (d) AFINN-RNTN SE of FPR in Figure 4.13; H_0 can be rejected for (e) AFINN-RNTN SE of FNR in Figure 4.13; H_0 cannot be rejected for (f) AFINN-RNTN SE of FNeR in Figure 4.13.

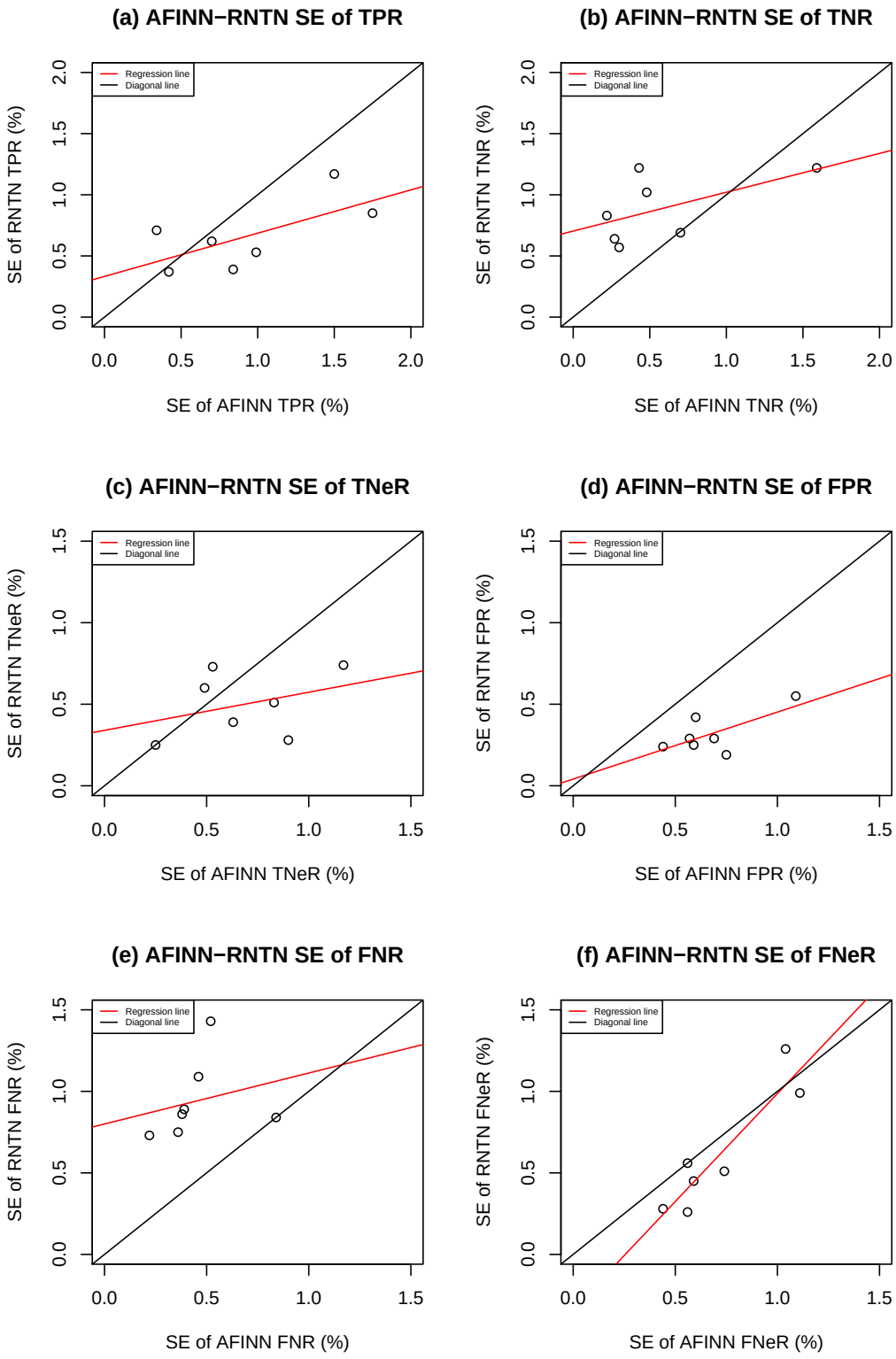

Figure 4.13. *The LS regression model (the red line) and the diagonal model (the black line) for the standard errors of fundamental errors of AFINN and RNTN in 3-class cross-validation.*

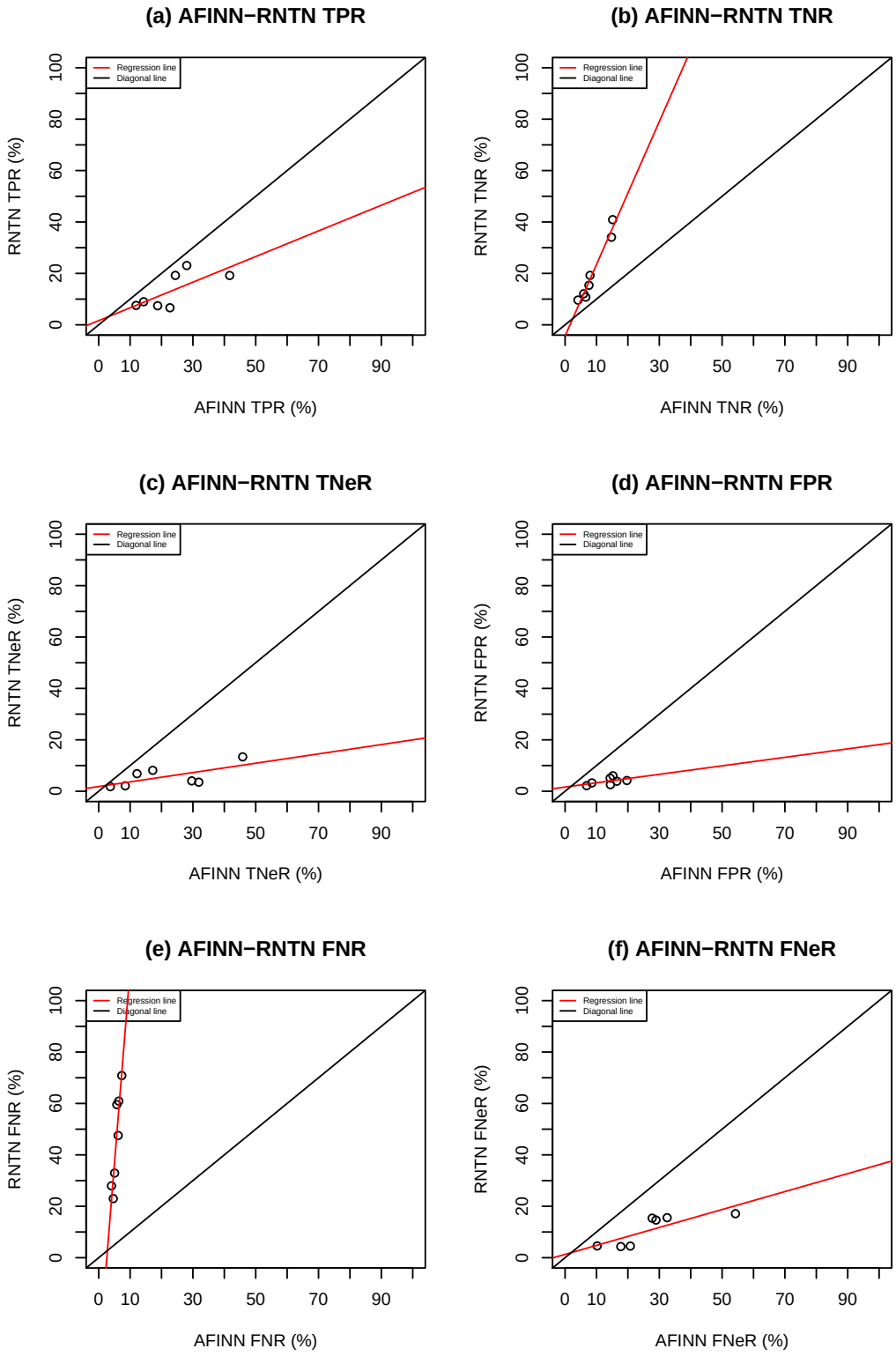

Figure 4.14. *The LS regression model (the red line) and the diagonal model (the black line) for the fundamental errors of AFINN and RNTN in 3-class cross-validation.*

Table 4.28. *The hypothesis testing results for Figure 4.13.*

Metrics	Intercept (%)	p-value (Intercept)	Slope (%)	p-value (Slope)	R^2	p-value (Comparison)
SE_TPR	0.3325	0.1371	0.3536	0.1041	0.4404	0.2377
SE_TNR	0.7033	0.0057	0.3175	0.1926	0.3119	0.14
SE_TNeR	0.3396	0.1587	0.2338	0.4366	0.125	0.1625
SE_FPR	0.041	0.7782	0.4108	0.0904	0.4673	0.0029
SE_FNR	0.8	0.0302	0.3124	0.593	0.0611	4e-04
SE_FNeR	-0.3375	0.1442	1.3239	0.0036	0.8413	0.5332

Table 4.29. *The hypothesis testing results for Figure 4.14.*

Metrics	Intercept (%)	p-value (Intercept)	Slope (%)	p-value (Slope)	R^2	p-value (Comparison)
TPR	1.6246	0.7817	0.4986	0.0759	0.4994	0.0424
TNR	-4.3526	0.1444	2.7803	1e-04	0.9585	0.0481
TNeR	1.8324	0.4679	0.1815	0.1049	0.4389	0.0336
FPR	1.6326	0.3656	0.1651	0.2107	0.2918	0.0011
FNR	-39.2039	0.0619	15.1253	0.0032	0.8491	7e-04
FNeR	1.2958	0.7193	0.3489	0.0265	0.6593	0.02

Considering the p-value (Comparison) in Table 4.29, we can conclude that: H_0 can be rejected for (a) AFINN-RNTN TPR in Figure 4.14; H_0 can be rejected for (b) AFINN-RNTN TNR in Figure 4.14; H_0 can be rejected for (c) AFINN-RNTN TNeR in Figure 4.14; H_0 can be rejected for (d) AFINN-RNTN FPR in Figure 4.14; H_0 can be rejected for (e) AFINN-RNTN FNR in Figure 4.14; H_0 can be rejected for (f) AFINN-RNTN FNeR in Figure 4.14.

Table 4.30 and Table 4.31 indicate the performance assessment results for AFINN and RNTN, respectively.

Figure 4.15 shows the LS regression model and the diagonal model for the standard errors of performance assessment results, and Table 4.32 provides the hypothesis testing results for Figure 4.15. Figure 4.16 exhibits the same models for the absolute values of performance assessment results, and Table 4.33 provides the hypothesis testing results for Figure 4.16.

Considering the p-value (Comparison) in Table 4.32, these conclusions can be obtained: H_0 cannot be rejected for (a) AFINN-RNTN SE of Accuracy in Figure 4.15; H_0 cannot be rejected for (b) AFINN-RNTN SE of Macro_*F1* in Figure 4.15; H_0 cannot be rejected for (c) AFINN-RNTN SE of Precision_Pos in Figure 4.15; H_0 cannot be rejected for (d) AFINN-

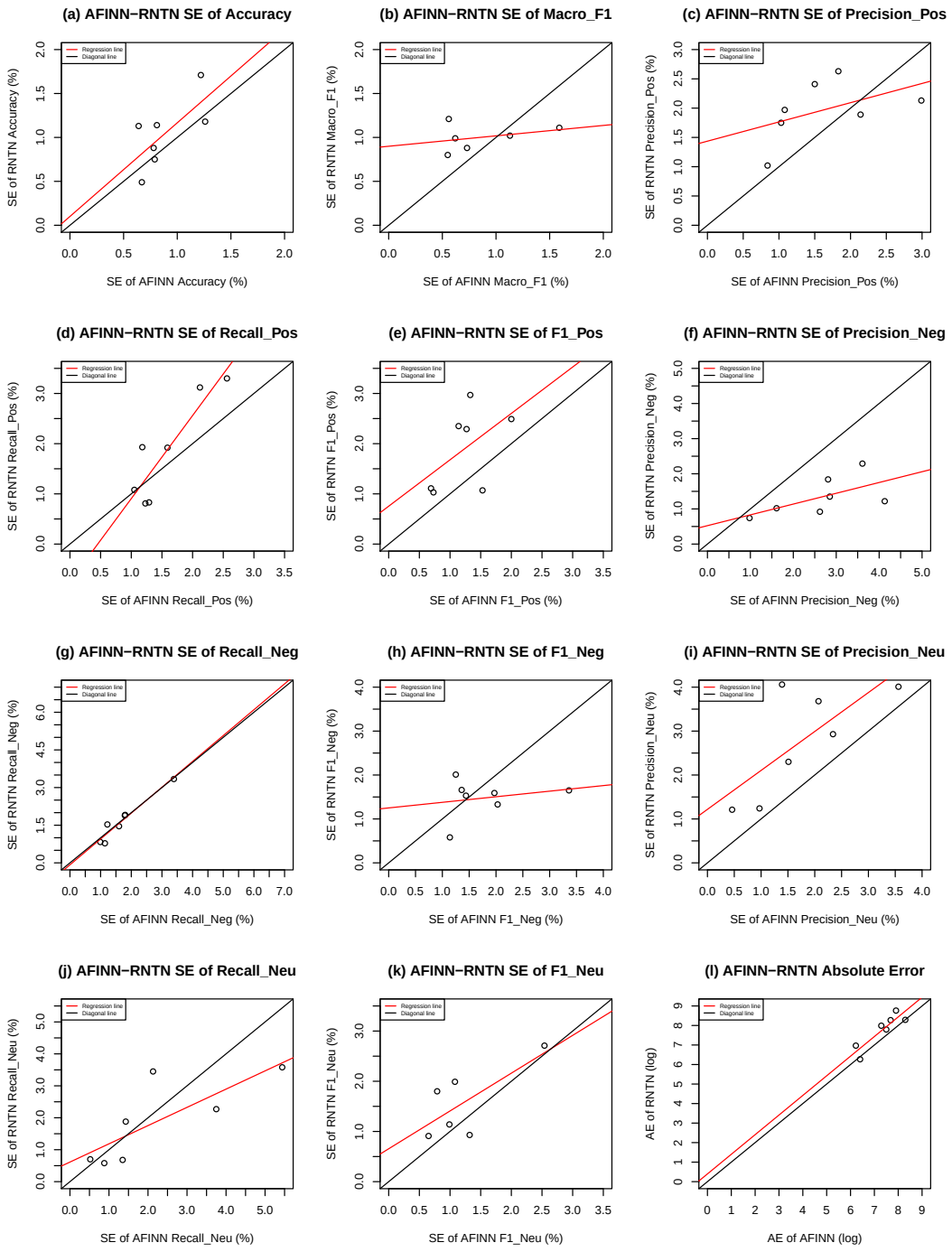

Figure 4.15. *The LS regression model (the red line) and the diagonal model (the black line) for the standard errors of performance assessment results of AFINN and RNTN in 3-class cross-validation.*

Table 4.30. *The performance assessment results of 3-class cross-validation with seven datasets for sentiment analysis, classifier = AFINN.*

Metrics	Comments_NYT	Comments_TED	Comments_YTB	Myspace	Tweets_RND_I	Tweets_RND_III	Tweets_Semeval
Accuracy (%)	34.56 ± 0.67	44.92 ± 1.22	54.01 ± 0.64	58.21 ± 1.26	55.17 ± 0.78	64.1 ± 0.79	60.02 ± 0.81
Precision_Pos (%)	62.63 ± 2.14	58.05 ± 2.99	68.58 ± 0.84	85.86 ± 1.03	52.49 ± 1.5	40.44 ± 1.08	60.09 ± 1.83
Recall_Pos (%)	35.04 ± 1.23	65.26 ± 2.56	60.81 ± 1.05	60.31 ± 1.59	59.17 ± 1.18	65.07 ± 2.12	61.35 ± 1.29
F1_Pos (%)	44.91 ± 1.53	60.8 ± 2	64.4 ± 0.73	70.77 ± 1.27	55.51 ± 1.14	49.82 ± 1.33	60.42 ± 0.69
Precision_Neg (%)	80.66 ± 0.98	74.04 ± 3.61	59.76 ± 2.81	41.77 ± 4.13	54.3 ± 2.85	48.54 ± 2.62	42.92 ± 1.61
Recall_Neg (%)	29.88 ± 0.99	27.22 ± 1.8	34.28 ± 1.22	37.65 ± 3.39	34.25 ± 1.14	47.32 ± 1.79	40.55 ± 1.6
F1_Neg (%)	43.54 ± 1.14	39.38 ± 1.97	43.26 ± 1.25	39.12 ± 3.36	41.7 ± 1.36	47.75 ± 2.03	41.59 ± 1.44
Precision_Neu (%)	6.76 ± 0.46	17.84 ± 3.56	38.05 ± 1.51	30.25 ± 2.07	57.33 ± 1.39	80.78 ± 0.97	64.24 ± 2.34
Recall_Neu (%)	81.28 ± 2.13	59.05 ± 5.44	58.29 ± 1.43	63.04 ± 3.75	62.69 ± 1.35	66.99 ± 0.52	64.67 ± 0.88
F1_Neu (%)	12.46 ± 0.79	24.96 ± 4.09	45.8 ± 1.08	40.71 ± 2.54	59.76 ± 0.99	73.23 ± 0.65	64.25 ± 1.32
Macro_*F1* (%)	33.64 ± 0.55	41.71 ± 1.44	51.15 ± 0.56	50.2 ± 1.59	52.32 ± 0.73	56.93 ± 1.13	55.42 ± 0.62
Absolute Error	4012	603	1807	504	2167	1466	2717

Table 4.31. *The performance assessment results of 3-class cross-validation with seven datasets for sentiment analysis, classifier = RNTN.*

Metrics	Comments_NYT	Comments_TED	Comments_YTB	Myspace	Tweets_RND_I	Tweets_RND_III	Tweets_Semeval
Accuracy (%)	51.68 ± 0.49	55.42 ± 1.71	46.57 ± 1.13	35.65 ± 1.18	30.68 ± 0.88	31.69 ± 0.75	22.26 ± 1.14
Precision_Pos (%)	73.46 ± 1.89	76.11 ± 2.13	81.8 ± 1.02	90.33 ± 1.75	64.78 ± 2.41	63.44 ± 1.97	72.35 ± 2.63
Recall_Pos (%)	21.14 ± 0.81	52.49 ± 3.3	47.16 ± 1.08	28.57 ± 1.92	23.54 ± 1.93	38.29 ± 3.12	18.23 ± 0.83
F1_Pos (%)	32.76 ± 1.07	61.49 ± 2.49	59.77 ± 1.03	42.99 ± 2.29	34.29 ± 2.35	47.43 ± 2.97	28.98 ± 1.11
Precision_Neg (%)	59.44 ± 0.74	59.81 ± 2.29	31.68 ± 1.84	16.84 ± 1.22	23.92 ± 1.35	15.32 ± 0.92	14.57 ± 1.02
Recall_Neg (%)	77.5 ± 0.83	70.38 ± 1.91	67.86 ± 1.53	76.16 ± 3.34	85.96 ± 0.78	83.01 ± 1.89	87.68 ± 1.46
F1_Neg (%)	67.24 ± 0.58	64.38 ± 1.59	43.05 ± 2.01	27.35 ± 1.65	37.25 ± 1.66	25.77 ± 1.33	24.88 ± 1.53
Precision_Neu (%)	9.45 ± 1.21	11.85 ± 4.01	34.48 ± 2.3	32.44 ± 3.68	47.28 ± 4.06	74.55 ± 1.24	44.92 ± 2.93
Recall_Neu (%)	37.09 ± 3.45	10.84 ± 3.58	28.48 ± 1.88	34.34 ± 2.27	8.67 ± 0.68	19.85 ± 0.7	7.16 ± 0.58
F1_Neu (%)	14.99 ± 1.8	0	31.12 ± 1.99	32.72 ± 2.71	14.61 ± 1.14	31.29 ± 0.91	12.31 ± 0.93
Macro_*F1* (%)	38.33 ± 0.8	0	44.64 ± 1.21	34.35 ± 1.11	28.72 ± 0.88	34.83 ± 1.02	22.05 ± 0.99
Absolute Error	3982	528	2428	1063	3915	2948	6398

RNTN SE of Recall_Pos in Figure 4.15; H_0 cannot be rejected for (e) AFINN-RNTN SE of F1_Pos in Figure 4.15; H_0 can be rejected for (f) AFINN-RNTN SE of Precision_Neg in Figure 4.15; H_0 cannot be rejected for (g) AFINN-RNTN SE of Recall_Neg in Figure 4.15; H_0 cannot be rejected for (h) AFINN-RNTN SE of F1_Neg in Figure 4.15; H_0 can be rejected for (i) AFINN-RNTN SE of Precision_Neu in Figure 4.15; H_0 cannot be rejected for (j) AFINN-RNTN SE of Recall_Neu in Figure 4.15; H_0 cannot be rejected for (k) AFINN-RNTN SE of F1_Neu in Figure 4.15; H_0 cannot be rejected for (l) AFINN-RNTN SE of Absolute Error in Figure 4.15.

Considering the p-value (Comparison) in Table 4.33, these conclusions can be obtained: H_0 can be rejected for (a) AFINN-RNTN Accuracy in Figure 4.16; H_0 can be rejected for (b) AFINN-RNTN Macro_*F1* in Figure 4.16; H_0 cannot be rejected for (c) AFINN-RNTN Precision_Pos in Figure 4.16; H_0 can be rejected for (d) AFINN-RNTN Recall_Pos in Figure 4.16; H_0 can be rejected for (e) AFINN-RNTN F1_Pos in Figure 4.16; H_0 can be rejected for (f) AFINN-RNTN Precision_Neg in Figure 4.16; H_0 can be rejected for (g) AFINN-RNTN Recall_Neg in Figure 4.16; H_0 cannot be rejected for (h) AFINN-RNTN F1_Neg in Figure 4.16; H_0 cannot be rejected for (i) AFINN-RNTN Precision_Neu in

Table 4.32. *The hypothesis testing results for Figure 4.15.*

Metrics	Intercept (%)	p-value (Intercept)	Slope (%)	p-value (Slope)	R^2	p-value (Comparison)
SE_Accuracy	0.1006	0.8309	1.0658	0.0816	0.4862	0.2791
SE_Precision (Pos)	1.4361	0.0294	0.3284	0.2749	0.231	0.2938
SE_Recall (Pos)	-0.7499	0.2658	1.6551	0.0059	0.8077	0.5095
SE_F1Pos	0.7543	0.4347	0.9241	0.2309	0.271	0.0156
SE_Precision (Neg)	0.5239	0.3527	0.307	0.1488	0.3678	0.0175
SE_Recall (Neg)	-0.0777	0.7437	1.0297	4e-04	0.9356	0.9517
SE_F1 (Neg)	1.2507	0.0497	0.1271	0.6348	0.0486	0.3252
SE_Precision (Neu)	1.218	0.1577	0.8865	0.0609	0.5372	0.0707
SE_Recall (Neu)	0.6133	0.3249	0.5708	0.0381	0.6102	0.6713
SE_F1 (Neu)	0.6551	0.2611	0.753	0.1072	0.5173	0.6794
SE_Macro_*F1*	0.8994	0.0049	0.1185	0.5217	0.1095	0.7067
Absolute Error	0.3968	0.8219	1.0053	0.0069	0.7965	0.309

Table 4.33. *The hypothesis testing results for Figure 4.16.*

Metrics	Intercept (%)	p-value (Intercept)	Slope (%)	p-value (Slope)	R^2	p-value (Comparison)
Accuracy	90.6004	0.0036	-0.9711	0.0311	0.6386	0.0229
Precision_Pos	35.5954	0.0014	0.6379	9e-04	0.9098	0.0579
Recall_Pos	-4.5178	0.8832	0.6414	0.2525	0.2506	3e-04
F1_Pos	13.2964	0.7291	0.5278	0.4326	0.127	0.0043
Precision_Neg	-42.4655	0.0033	1.2907	2e-04	0.9467	0.0179
Recall_Neg	55.7439	0.0146	0.6305	0.1918	0.3127	0
F1_Neg	90.3971	0.4633	-1.157	0.6844	0.0358	0.5988
Precision_Neu	2.0542	0.6819	0.8149	4e-04	0.934	0.6643
Recall_Neu	-29.3892	0.5127	0.7723	0.2794	0.2273	0
F1_Neu	20.0225	0.1677	0.0571	0.8112	0.016	0.0305
Macro_*F1*	52.2569	0.0721	-0.3692	0.435	0.1581	0.0027
Absolute Error	0.3968	0.8219	1.0053	0.0069	0.7965	0.309

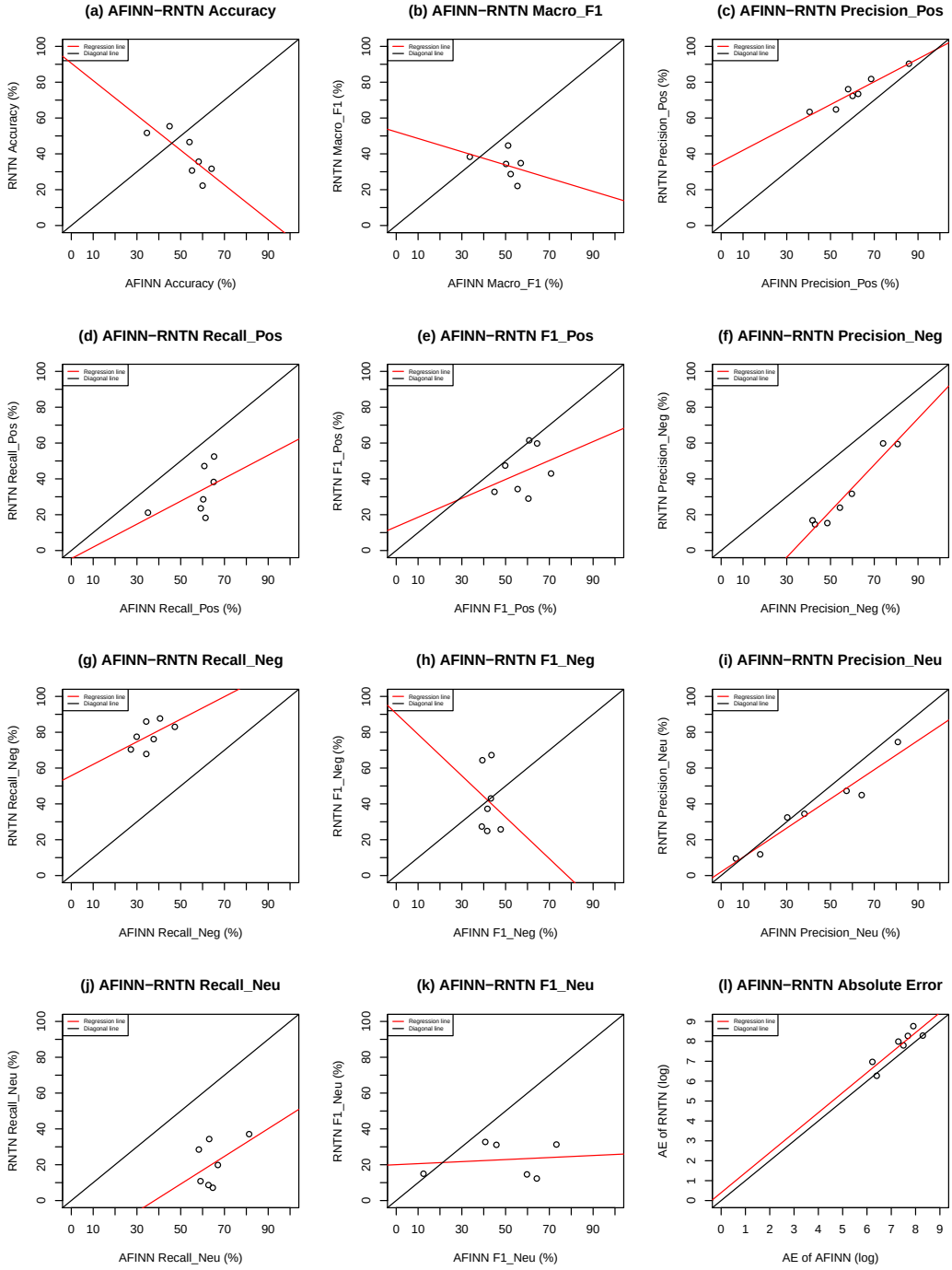

Figure 4.16. *The LS regression model (the red line) and the diagonal model (the black line) for the performance assessment results of AFINN and RNTN in 3-class cross-validation.*

Figure 4.16; H_0 can be rejected for (j) AFINN-RNTN Recall_Neu in Figure 4.16; H_0 can be rejected for (k) AFINN-RNTN F1_Neu in Figure 4.16; H_0 cannot be rejected for (l) AFINN-RNTN Absolute Error in Figure 4.16.

4.3 Discussion

We start the discussions of our experiments by comparing all previous results. Because bootstrap test and cross-validation test have almost the same results for both 2-class and 3-class comparisons, we focus on the analysis of the bootstrap test in this part.

Let's consider 2-class comparisons firstly. For the fundamental errors, AFINN does not have same performance with RNTN across all datasets considering both standard errors and absolute values. By analyzing the p-value (Comparison) in Table 4.4, we cannot reject H_0 for the standard error of TPR, TNR, and FPR in Figure 4.1, but we can reject H_0 for the standard error of FNR in Figure 4.1. However, based on the p-value (Comparison) in Table 4.5, we can reject H_0 for the absolute value of TPR, TNR, FPR, and FNR in Figure 4.2. Therefore, whether considering both standard errors and absolute values or considering only absolute values of the fundamental errors, AFINN does not perform identically with RNTN for all datasets.

According to the p-value (Comparison) in Table 4.8, we cannot reject H_0 for the standard error of Accuracy, Precision_Pos, F1_Pos, Precision_Neg, Recall_Neg, F1_Neg, Macro_*F1*, and Absolute Error in Figure 4.3, but we can reject H_0 for the standard error of Recall_Pos in Figure 4.3. By analyzing the p-value (Comparison) in Table 4.9, we cannot reject H_0 for the absolute value of Macro_*F1*, Precision_Pos and Absolute Error in Figure 4.4, but we can reject H_0 for the absolute value of Accuracy, Recall_Pos, F1_Pos, Precision_Neg, Recall_Neg, and F1_Neg in Figure 4.4. If we consider both standard errors and absolute values of the overall performance assessment results, we can conclude that AFINN does not perform identically with RNTN for all datasets on Accuracy, Recall_Pos, F1_Pos, Precision_Neg, Recall_Neg, and F1_Neg; however, AFINN has identical performance with RNTN for all datasets on Macro_*F1*, Precision_Pos and Absolute Error. If only absolute values of the overall performance assessment results are taken into account, we can conclude that AFINN does not have same performance with RNTN for all datasets on Accuracy, Recall_Pos, F1_Pos, Precision_Neg, Recall_Neg, and F1_Neg; however, AFINN has identical performance with RNTN for all datasets on Macro_*F1*, Precision_Pos and Absolute Error.

Next is the 3-class comparisons. In terms of the p-value (Comparison) in Table 4.20, we cannot reject H_0 for the standard error of TPR, TNR, TNeR and FNeR in Figure 4.9, but we can reject H_0 for the standard error of FPR and FNR in Figure 4.9. Based on the p-value (Comparison) in Table 4.21, we can reject H_0 for the absolute value of TPR, TNR, TNeR, FPR, FNR and FNeR in Figure 4.10. Thus, AFINN does not have identical performance with RNTN across all datasets considering both standard errors

and absolute values of the fundamental errors. If we do not consider standard errors of the fundamental errors, we can conclude that AFINN does not have same performance with RNTN for all datasets, either.

By analyzing the p-value (Comparison) in Table 4.24, we can reject H_0 for the standard error of Precision_Neg and Precision_Neu in Figure 4.11, but we cannot reject H_0 for the standard error of Accuracy, Precision_Pos, Recall_Pos, F1_Pos, Recall_Neg, F1_Neg, Recall_Neu, F1_Neu, Macro_*F1* and Absolute Error in Figure 4.11. According to the p-value (Comparison) in Table 4.25, we can reject H_0 for the absolute value of Accuracy, Recall_Pos, F1_Pos, Precision_Neg, Recall_Neg, Recall_Neu, F1_Neu and Macro_*F1* in Figure 4.12, but we cannot reject H_0 for the absolute value of Precision_Pos, F1_Neg, Precision_Neu and Absolute Error in Figure 4.12. If we consider both standard errors and absolute values of the overall performance assessment results, we can conclude that AFINN does not perform identically with RNTN for all datasets on Accuracy, Recall_Pos, F1_Pos, Precision_Neg, Recall_Neg, Precision_Neu, Recall_Neu, F1_Neu and Macro_*F1*; however, AFINN has identical performance with RNTN for all datasets on Precision_Pos, F1_Neg and Absolute Error. If only absolute values of the overall performance assessment results are taken into account, we can conclude that AFINN does not have same performance with RNTN for all datasets on Accuracy, Recall_Pos, F1_Pos, Precision_Neg, Recall_Neg, Recall_Neu, F1_Neu and Macro_*F1*; however, AFINN has identical performance with RNTN for all datasets on Precision_Pos, F1_Neg, Precision_Neu and Absolute Error.

From the overall performance assessment results, we can easily note that AFINN and RNTN yield with large variations across the different datasets. By analyzing accuracy and Macro-*F1* in Table 4.6 and Table 4.7, AFINN obtains better performance in Comments_YTB, Myspace, Tweets_RND_I, Tweets_RND_III, and Tweets _Semeval; RNTN acquires better performance in Comments_NYT and Comment_TED. It statistically states that no single method (AFINN or RNTN) can always achieve the best prediction performance across all datasets in terms of accuracy and Macro-*F1*. And the same situation also exists in 3-class comparisons.

The seven datasets can be divided into two specific contexts: Social Networks (Myspace, Tweets_RND_I, Tweets_RND_III, and Tweets _Semeval) and Comments (Comments_NYT, Comment_TED, and Comments_YTB). AFINN performs much better in the context of Social Network than RNTN. One possible reason is the difference in average number of phrases of the dataset. By observing Table 3.2, the data of Social Network and Comments_YTB have higher average number of phrases than Comments_NYT and Comments_TED. Another possible reason can be that there are more complicated topics and opinions in the context of comments, such as science and culture. However, people express more straightforward opinions on economy, products and politics in the context of Social Network. In our comparisons, AFINN also performs better in Comments_YTB, which more or less indicates that AFINN has wider application contexts than RNTN.

In 3-class comparisons, F1_Neu and Macro_*F1* scores are 0 in Table 4.23 and Table 4.31.

However, F1_Neu and Macro_*F1* scores exist in Table 4.22 and Table 4.30. Moreover, this situation does not occur in 2-class comparisons. It states that RNTN is more biased towards classes which contain more messages, given neutral class is quite smaller compared to positive and negative classes. Therefore, AFINN could have more predication effectiveness on skewed classes.

Dependent on the above analysis, we can note RNTN is more specialized and AFINN is more generic in sentence-level sentiment analysis. For example, AFINN can achieve 44.92% accuracy for Comments_TED and 60.02% accuracy for Tweets_Semeval in Table 4.30; RNTN can achieve 55.42% accuracy for Comments_TED and 22.26% accuracy for Tweets_Semeval in Table 4.31. Even RNTN performs better than AFINN for Comments_TED, the difference is only 10.5%. However, the accuracy of AFINN is 37.76% higher than that of RNTN for Tweets_Semeval.

Another important metric is the computation time. Because AFINN belongs to the lexicon-based methods, the training time is quite short and can be omitted. However, RNTN is basically a deep learning approach, thus, it needs 3 – 5 hours to train a proper model. In addition, AFINN only needs about 0.05s to predict one novel sentence, while RNTN takes approximately 0.25s.

5 CONCLUSION

Various sentiment analysis methods have been adopted to analyze moods of unstructured sentences in online networks, including lexicon-based and machine-learning methods. To obtain better performance, many natural language processing techniques are widely used in sentiment analysis, such as tokenization, filtering, lemmatization, stemming, linguistic processing, and etc.

In this thesis, we have compared two different sentiment analysis techniques: one simple lexicon-based method namely AFINN and one more complicated machine-learning method namely RNTN. We present a thorough comparison between AFINN and RNTN using seven labeled datasets that cover different types of data sources. Our effort quantifies the prediction performance of AFINN and RNTN across all datasets. To obtain more robust conclusions, we use bootstrap and cross validation to assess their performance, respectively.

In 2-class and 3-class comparisons, AFINN and RNTN do not have identical performance across all datasets whether considering both standard errors and absolute values or considering only absolute values of the fundamental errors. In 2-class comparisons, if considering both standard errors and absolute values of the overall performance assessment results, AFINN does not perform identically with RNTN for all datasets on Accuracy, Recall_Pos, F1_Pos, Precision_Neg, Recall_Neg, and F1_Neg; however, AFINN has identical performance with RNTN for all datasets on Macro_*F1*, Precision_Pos and Absolute Error. If only absolute values of the overall performance assessment results are taken into account in 2-class comparisons, AFINN does not have same performance with RNTN for all datasets on Accuracy, Recall_Pos, F1_Pos, Precision_Neg, Recall_Neg, and F1_Neg; however, AFINN has identical performance with RNTN for all datasets on Macro_*F1*, Precision_Pos and Absolute Error. In 3-class comparisons, if we consider both standard errors and absolute values of the overall performance assessment results, AFINN does not perform identically with RNTN for all datasets on Accuracy, Recall_Pos, F1_Pos, Precision_Neg, Recall_Neg, Precision_Neu, Recall_Neu, F1_Neu and Macro_*F1*; however, AFINN has identical performance with RNTN for all datasets on Precision_Pos, F1_Neg and Absolute Error. If only absolute values of the overall performance assessment results are taken into account in 3-class comparisons, AFINN does not have same performance with RNTN for all datasets on Accuracy, Recall_Pos, F1_Pos, Precision_Neg, Recall_Neg, Recall_Neu, F1_Neu and Macro_*F1*; however, AFINN has identical performance with RNTN for all datasets on Precision_Pos, F1_Neg, Precision_Neu and Abso-

lute Error.

Furthermore, we highlight that the prediction performance of AFINN and RNTN varies considerably from one dataset to another. AFINN performs better on Comments_YTB, Myspace, Tweets_RND_I, Tweets_RND_III, and Tweets _Semeval, while RNTN obtains better performance on Comments_TED and Comments_NYT. This suggests that sentiment analysis methods cannot be used as "off-the-shelf" methods for novel datasets. More important, we state that the performance of AFINN and RNTN is different in specific contexts. RNTN has much worse performance in the context of Social Network; to some extent, AFINN has wider sentiment analysis contexts than RNTN. Therefore, it is important that researchers and companies carry out context analysis before applying a sentiment analysis method.

Among many findings, we also show that AFINN could have more prediction effectiveness on skewed classes while RNTN is more biased toward classes which contain more sentences. Additionally, RNTN takes much more computation time in sentiment prediction. For example, 3 – 5 hours are needed for RNTN to train a proper model and it takes about 5 times the average testing time of AFINN to predict single sentence. Our findings suggest that AFINN is more simple, more generic and takes less computation resources than RNTN in sentiment analysis.

The datasets we used in this thesis cover a wide range of sources with three classes (positive, negative and neutral), including the contexts of Social Network and Comments. Additionally, they are labeled by human or Amazon Mechanical Turk (AMT) with relative high level of agreement. Thus, the seven datasets can be built as a representative standard benchmark not only for sentiment analysis but also for other text mining tasks, such as text summarization, topic modelling and document clustering. To better understand the performance of methods in types of data, we can also extend the benchmark with datasets of another specific context – Reviews in the future.

In the future, we would focus on further improving our comparisons. We can extend our experiments with more evaluated metrics, such as the overall performance ranking and Friedman's Test which allows us to verify whether the methods present similar performance across different datasets. Moreover, we can explore whether the methods have biases toward the polarity. Since prediction performance varies considerably from one dataset to another, a more generic sentiment analysis method is needed. In our opinion, it is not necessary to use one complicated method to complete the specific task if a simple approach can also achieve the same performance. Recent efforts have provided many sentiment analysis methods which are widely used in their knowledge fields. In this thesis, we only carry out a comparison between AFINN and RNTN which are the typical representation of lexicon-based methods and machine-learning methods, respectively. However, we could continue to search for the simplest sentiment analysis technique but with the best performance.

REFERENCES

[1] E. Alpaydin. *Introduction to machine learning*. MIT press, 2009.

[2] S. Arlot, A. Celisse, et al. A survey of cross-validation procedures for model selection. In: *Statistics surveys* 4 (2010), 40–79.

[3] M. F. Azam, A. Musa, M. Dehmer, O. P. Yli-Harja, and F. Emmert-Streib. Global Genetics Research in Prostate Cancer: A Text Mining and Computational Network Theory Approach. In: *Frontiers in genetics* 10 (2019), 70.

[4] P. Baldi, S. Brunak, and F. Bach. *Bioinformatics: the machine learning approach*. MIT press, 2001.

[5] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. In: *Journal of computational science* 2.1 (2011), 1–8.

[6] M. M. Bradley and P. J. Lang. *Affective norms for English words (ANEW): Instruction manual and affective ratings*. Tech. rep. 1999.

[7] E. Cambria, R. Speer, C. Havasi, and A. Hussain. Senticnet: A publicly available semantic resource for opinion mining. In: *2010 AAAI Fall Symposium Series*. 2010.

[8] M.-S. Chen, J. Han, and P. S. Yu. Data mining: an overview from a database perspective. In: *IEEE Transactions on Knowledge and data Engineering* 8.6 (1996), 866–883.

[9] G. G. Chowdhury. Natural language processing. In: *Annual review of information science and technology* 37.1 (2003), 51–89.

[10] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In: *Proceedings of the 25th international conference on Machine learning*. ACM. 2008, 160–167.

[11] T. M. Cover, P. E. Hart, et al. Nearest neighbor pattern classification. In: *IEEE transactions on information theory* 13.1 (1967), 21–27.

[12] P. S. Dodds and C. M. Danforth. Measuring the happiness of large-scale written expression: Songs, blogs, and presidents. In: *Journal of happiness studies* 11.4 (2010), 441–456.

[13] B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

[14] F. Emmert-Streib and M. Dehmer. Defining data science by a data-driven quantification of the community. In: *Machine Learning and Knowledge Extraction* 1.1 (2019), 235–251.

[15] F. Emmert-Streib, S. Moutari, and M. Dehmer. A comprehensive survey of error measures for evaluating binary decision making in data science. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* (2019), e1303.

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

[17] W. Fan, L. Wallace, S. Rich, and Z. Zhang. Tapping the power of text mining. In: *Communications of the ACM* 49.9 (2006), 76–82.

[18] S. Federici, S. Montemagni, and V. Pirrelli. Shallow parsing and text chunking: a view on underspecification in syntax. In: *Cognitive science research paper-university of Sussex CSRP* (1996), 35–44.

[19] R. Feldman. Techniques and applications for sentiment analysis. In: *Communications of the ACM* 56.4 (2013), 82–89.

[20] 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). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf*. 2014.

[21] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. In: *CS224N Project Report, Stanford* 1.12 (2009).

[22] P. Gonçalves, M. Araújo, F. Benevenuto, and M. Cha. Comparing and combining sentiment analysis methods. In: *Proceedings of the first ACM conference on Online social networks*. ACM. 2013, 27–38.

[23] P. Gonçalves, F. Benevenuto, and M. Cha. Panas-t: A pychometric scale for measuring sentiments on twitter. In: *arXiv preprint arXiv:1308.1857* (2013).

[24] V. Gupta, G. S. Lehal, et al. A survey of text mining techniques and applications. In: *Journal of emerging technologies in web intelligence* 1.1 (2009), 60–76.

[25] D. J. Hand. Data Mining. In: *Encyclopedia of Environmetrics* 2 (2006).

[26] M. Hearst. What is text mining. In: *SIMS, UC Berkeley* (2003).

[27] A. Hotho, A. Nürnberger, and G. Paaß. A brief survey of text mining. In: *Ldv Forum*. Vol. 20. 1. 2005, 19–62.

[28] M. Hu and B. Liu. Mining and summarizing customer reviews. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2004, 168–177.

[29] A. G. Jivani et al. A comparative study of stemming algorithms. In: *Int. J. Comp. Tech. Appl* 2.6 (2011), 1930–1938.

[30] K. S. Jones. Natural language processing: a historical review. In: *Current issues in computational linguistics: in honour of Don Walker*. Springer, 1994, 3–16.

[31] A. Kao and S. R. Poteet. *Natural language processing and text mining*. Springer Science & Business Media, 2007.

[32] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In: *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics. 2003, 423–430.

[33] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. In: *biometrics* (1977), 159–174.

[34] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. In: *nature* 521.7553 (2015), 436.

[35] R. Lichtig. *The history and development of text based natural language processing*. 2011. URL: https://ethw.org/The_History_of_Natural_Language_

Processing080207010024/http://www.808multimedia.com/winnt/kernel.htm (visited on 05/06/2019).

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

[37] J. B. Lovins. Development of a stemming algorithm. In: *Mech. Translat. & Comp. Linguistics* 11.1-2 (1968), 22–31.

[38] C. D. Manning, C. D. Manning, and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[39] P. Nakov, Z. Kozareva, A. Ritter, S. Rosenthal, V. Stoyanov, and T. Wilson. *SemEval-2013 Task 2: Sentiment Analysis in Twitter*. 2013.

[40] S. Narr, M. Hulfenhaus, and S. Albayrak. Language-independent twitter sentiment analysis. In: *Knowledge discovery and machine learning (KDML), LWA* (2012), 12–14.

[41] R. Navigli. Word sense disambiguation: A survey. In: *ACM computing surveys (CSUR)* 41.2 (2009), 10.

[42] F. Å. Nielsen. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In: *arXiv preprint arXiv:1103.2903* (2011).

[43] N. Oliveira, P. Cortez, and N. Areal. On the predictability of stock market behavior using stocktwits sentiment and posting volume. In: *Portuguese Conference on Artificial Intelligence*. Springer. 2013, 355–365.

[44] C. E. Osgood, G. J. Suci, and P. H. Tannenbaum. *The measurement of meaning*. 47. University of Illinois press, 1957.

[45] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics. 2005, 115–124.

[46] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics. 2002, 79–86.

[47] B. Pang, L. Lee, et al. Opinion mining and sentiment analysis. In: *Foundations and Trends® in Information Retrieval* 2.1–2 (2008), 1–135.

[48] N. Pappas and A. Popescu-Belis. Sentiment analysis of user comments for one-class collaborative filtering over ted talks. In: *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM. 2013, 773–776.

[49] J. Plisson, N. Lavrac, D. Mladenić, et al. A rule based approach to word lemmatization. In: (2004).

[50] G. Qiu, B. Liu, J. Bu, and C. Chen. Expanding domain sentiment lexicon through double propagation. In: *IJCAI*. Vol. 9. 2009, 1199–1204.

[51] J. R. Quinlan. Induction of decision trees. In: *Machine learning* 1.1 (1986), 81–106.

[52] A. Ratnaparkhi. Maximum entropy models for natural language ambiguity resolution. In: (1998).

[53] F. N. Ribeiro, M. Araújo, P. Gonçalves, M. A. Gonçalves, and F. Benevenuto. Sentibench-a benchmark comparison of state-of-the-practice sentiment analysis methods. In: *EPJ Data Science* 5.1 (2016), 23.

[54] I. Rish et al. An empirical study of the naive Bayes classifier. In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. 22. 2001, 41–46.

[55] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In: *European conference on computer vision*. Springer. 2006, 430–443.

[56] S. Ruder. *A Review of the Neural History of Natural Language Processing*. 2018. URL: http://blog.aylien.com/a-review-of-the-recent-history-of-natural-language-processing/ (visited on 05/06/2019).

[57] H. Saif, M. Fernández, Y. He, and H. Alani. On stopwords, filtering and data sparsity for sentiment analysis of twitter. In: (2014).

[58] P. Schachter and T. Shopen. Parts-of-speech systems. In: *Language typology and syntactic description* 1 (1985), 3–61.

[59] B. Scholkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[60] J. Silge and D. Robinson. *Text mining with R: A tidy approach*. " O'Reilly Media, Inc.", 2017.

[61] C. Silva and B. Ribeiro. The importance of stop word removal on recall values in text categorization. In: *Neural Networks, 2003. Proceedings of the International Joint Conference on*. Vol. 3. IEEE. 2003, 1661–1666.

[62] 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*. 2013, 1631–1642.

[63] Y. R. Tausczik and J. W. Pennebaker. The psychological meaning of words: LIWC and computerized text analysis methods. In: *Journal of language and social psychology* 29.1 (2010), 24–54.

[64] *Text mining for word processing and sentiment analysis using 'dplyr', 'ggplot2', and other tidy tools*. https://cran.r-project.org/web/packages/tidytext/index.html. Accessed: 2019-05-06.

[65] M. Thelwall. The Heart and soul of the web? Sentiment strength detection in the social web with SentiStrength. In: *Cyberemotions*. Springer, 2017, 119–134.

[66] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In: *Icwsm* 10.1 (2010), 178–185.

[67] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[68] V. Yadav, H. Elchuri, et al. Serendio: Simple and Practical lexicon based approach to Sentiment Analysis. In: *Second Joint Conference on Lexical and Computational*

Semantics (SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Vol. 2. 2013, 543–548.

[69] C. Zhang, D. Zeng, J. Li, F.-Y. Wang, and W. Zuo. Sentiment analysis of Chinese documents: From sentence to document level. In: *Journal of the American Society for Information Science and Technology* 60.12 (2009), 2474–2487.