Identifying Product Defects by Applying a Predictive Model to Customer Reviews

by Titus Hei Yeung Fong

B.S. in Electrical Engineering, May 2016, University of Illinois at Urbana-Champaign
Master of Computer Science, May 2018, University of Illinois at Urbana-Champaign

A Praxis submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Engineering

August 31, 2020

Praxis directed by

Shahryar Sarkani
Adjunct Professor of Engineering Management and Systems Engineering

John Fossaceca
Professorial Lecturer of Engineering Management and Systems Engineering

The School of Engineering and Applied Science of The George Washington University certifies that Titus Hei Yeung Fong has passed the Final Examination for the degree of Doctor of Engineering as of July 31, 2020. This is the final and approved form of the Praxis.

Identifying Product Defects by Applying a Probabilistic Model to Customer Reviews

Titus Hei Yeung Fong

Praxis Research Committee:

Shahryar Sarkani, Adjunct Professor of Engineering Management and Systems Engineering, Praxis Co-Director

John Fossaceca, Professorial Lecturer of Engineering Management and Systems Engineering, Praxis Co-Director

Amir Etemadi, Associate Professor of Engineering and Applied Science, Committee Member

Dedication

I would like to dedicate this research to my parents, Robert and Viola, who encouraged, inspired, and supported me throughout this journey.

Acknowledgements

First and above all, I give glory to God for His Grace and blessings to walk me through all the challenges in this doctoral program.

I wish to thank Dr. Shahryar Sarkani and Dr. John Fossaceca for their guidance and support throughout the studying, research, and writing processes. I would also like to thank Dr. Samson Tai for providing me research suggestions.

Abstract of Praxis

Identifying Product Defects by Applying a Probabilistic Model to Customer Reviews

The challenge for consumer product engineering teams to manually explore their product’s defects from online customer reviews (OCR) delays product recall and recovery processes. In today's product life cycle, there is no practical method to automatically transfer the massive amount of valuable online customer reviews, such as design, performance, and serviceability feedback, to the product engineering teams. This lack of an early detection mechanism for problems often increases the risks of a product recall, potentially causing billions of dollars in economic loss, loss of company credibility, and loss of market penetration.

This research explores two different kinds of Recurrent Neural Network (RNN) models and one Latent Dirichlet Allocation (LDA) topic model to extract product defect information from OCRs. This research also proposes a novel approach, combined with RNN and LDA models, to provide engineers with an early view of product defects. The proposed approach first employs the RNN models for sentiment analysis on customer reviews to identify negative reviews and reviews that mention product defects, then applies the LDA model to retrieve a summary of key defect insight words from these reviews.

Results of this praxis show that engineering teams can discover early signs of potential defects and opportunities for improvement when using this novel approach on eight of the bestselling Amazon home furnishing products. This combined approach is able to locate the keywords of these products’ defects and issues that customers

mentioned the most in their OCRs, which allows the engineering team to take required mitigation actions earlier and proactively stop the diffusion of the detective products.

Table of Contents

List of Figures

List of Tables

List of Symbols

α	Parameter of the Dirichlet prior distribution on each OCRs - topic of an LDA topic model
B	Bias of LSTM functions
β	Parameter of the Dirichlet prior distribution on each topic-word of an LDA topic model
C	Memory Cell of LSTM model
h	Previous cell memory of LSTM model
K	Number of Latent topics is set to be outputted of the LDA topic model
M	Number of OCR input to the LDA topic model
N	Number of words in a given OCRs of the LDA topic model
tanh	Hyperbolic tangent function
W	Weight of LSTM functions
$w_{i,j}$	Current word in the j^{th} word in the i^{th} OCRs of an LDA topic model
X	Text representation vector of LSTM model
$z_{i,j}$	Current topic for current word $w_{i,j}$ of an LDA topic model
σ	Sigmoid function
θ_i	Topic distribution for the i^{th} OCRs in the LDA topic model
φ_k	Word distribution for topic *k* of the LDA topic model

List of Acronyms

DoD	Department of Defense
CNN	Convolutional Neural Network
CR	Customer Requirements
CPSC	United States Consumer Product Safety Commission
ICOSE	International Council on Systems Engineering
IR	Information Retrieval
k-NN	k-Nearest Neighbors
LDA	Latent Dirichlet Allocation
LSTM	Long Short-Term Memory
ML	Machine Learning
NB	Naive Bayes
NLP	Natural Language Processing
RNN	Recurrent Neural Network
SVM	Support-Vector Machines
TSV	Tab-Separated Values
OCR	Online Customer Reviews
U.S.	United States

Chapter 1—Introduction

1.1 Background

While companies have been using advanced quality control tools for product development, defective products can still be found on the market and product recall events often happen (Ahsan, 2013). These defective product events can cost a company billions of dollars of loss and bring significant brand reputation impact that lasts for an extended period. A good example is the recent 2016 Samsung Galaxy Note 7 explosions recall event. This event cost Samsung more than $5 billion of loss and the subsequent loss of sales in the electronics industry (Jeong, 2017).

Social media and e-commerce sites have enabled companies to review customer's feedback about their products. This feedback includes customer complaints and defective information about the products. In the Samsung Galaxy Note 7 recall event, study showed that there were early customer reports of overheating problems through online customer feedback before Samsung realized and took actions (Mack, 2016; Tibken et al., 2017; Bleaney et al., 2018). It is, however, a challenge for the engineering teams to manually read through a large amount of customer feedback available online and be able to discover early signs of product defect, delaying the engineering teams to take necessary recovery actions.

In this praxis, research is presented to explore methods of extracting product defects information from online customer reviews (OCRs) and to demonstrate a novel predictive model using Recurrent Neural Network (RNN) and Latent Dirichlet Allocation (LDA) topic model to warn engineering teams about potential product recall. This new predictive model provides the engineering team with an early view into product defects, which enables the team to take required mitigation actions earlier and to proactively stop the spread of the detective products, hence preventing further potential damages and economic losses to the company.

1.2 Research Motivation

With the growth of the new era of Web 2.0, various social media and e-commerce sites such as Amazon.com and Twitter.com provide online virtual communities for consumers to share their feedback on different products and services. These customer feedbacks provide valuable information for the company and the engineering team, such as feedback on the design, performance, and serviceability of the product. While there is a large amount of such information available on these sites, there is a lack of an effective method to automatically distill this information for the engineering team.

As technology progresses with faster computers and better computational algorithms, we are now able to collect, process, and extract useful information from a large amount of textual customer feedback. In the field of business and product design, studies (Lee et al., 2010; Suryadi et al., 2019) show that machine learning predictive models can successfully extract useful information from OCRs, including customer buying patterns and customer requirements on future product design. This praxis provides the foundation for applying machine learning predictive models to detect defective product information from OCRs.

RNN is a type of neural network model for analyzing time-series data. This model is able to solve problems involving sequences of word order textual data. LDA is a type of a generative probabilistic model for discovering a set of topics best describes a collection of discrete data. Researchers have proposed using neural networks and topic models for sentiment classification and extraction of textual information. In this praxis, a new predictive model with this RNN and LDA approach is demonstrated to extract defective information in the area of product defect management.

1.3 Problem Statement

It is costly and tedious for product engineers to manually extract product defects from online customer reviews (Ahsan, 2013; Bleaney et al., 2018) resulting in damages to company's reputation (Jarrell et al., 1985) and loss of revenue (Sanchanta et al., 2010).

In current product development, companies using state-of-the-art tools and techniques find it difficult to make perfect products, and products often need to be recalled (Ahsan, 2013). While there is a large amount of customer feedback mentioning product issues available online in social media and e-commerce sites, it is difficult for engineers to extract this information manually in a timely manner. The recent 2016 Samsung Galaxy Note 7 defective product recall event cost Samsung more than $5 billion (Jeong, 2017). There were early signs of defects from online customer review sites found before Samsung took any actions. This lack of early detection of issues often delays defective product recovery actions and widens the spread of the defective products, causing further economic loss to the company.

1.4 Thesis Statement

A predictive model using customer sentiment analysis and topic modeling is required to provide engineers with an early view into product defects.

Customer reviews on social media provide valuable information back to the company. RNN is a type of neural network for analyzing time-series data which can be used as a predictive model to solve the problems involving sequences of word order textual data. By using RNN, this predictive model can build relations between words in semantic meaning on sentiment analysis, which can increase the accuracy beyond that of the traditional bag-of-words method (Rose et al., 2011). This RNN model along with LDA topic model allows classifying the meaning of reviews for early detection of the defect. Being able to determine timely engineering action, such as recall, is essential for the company because as the time to action increases, the recovery will be more challenging (Hora et al., 2011).

1.5 Research Objectives

The primary objectives of this research are to understand the information in OCRs pertaining to product defects and recall management, as well as to demonstrate a novel predictive model with a new RNN and LDA approach that can provide an early view into product defects for engineering teams. Detailed objectives are as follows:

1. To retrieve the product defect information contained in OCRs pertaining to the engineering and product development team.
2. To assess the RNN approach in predicting the customers' sentiment of OCRs.
3. To assess the RNN approach in classifying OCRs that mention useful product defects information for engineering teams.
4. To develop a predictive model with LDA approach to automatically identify a set of product defects words that best describe the associated product.

1.6 Research Questions and Hypotheses

The following research questions and hypotheses are the focus of this study based on the research objectives.

1.6.1 Research Questions

RQ1: How well can a RNN model accurately predict OCRs that contain scores with less than three (out of five) star customer ratings?

RQ2: Can a RNN model extract and differentiate OCRs that include information about defective products from OCRs that do not include information about defective products?

RQ3: Can a LDA topic model identify a set of key product defect topics for a single product using OCRs?

1.6.2 Research Hypotheses

H1: A predictive model can identify OCRs containing negative sentiments with 70% accuracy.

H2: A predictive model can identify OCRs containing defect information with 70% accuracy.

H3: A predictive model can identify 70% of key product defect topics for a single product using OCRs.

1.7 Scope of Research

The scope of this research is to explore the viability of a probabilistic model with RNN to analyze, extract, and identify defective product information from OCRs. This consists of collection of customer online review data with manual labels for supervised learning, quantitative models for testing the hypothesizes associated with identifying product defects, and verification of models.

The new proposed approach includes two quantitative RNN classifiers and one LDA topic model. The first RNN classifier differentiates negative OCRs from non-negative OCRs. The second RNN classifier differentiates OCRs contain defect information from OCRs that do not contain defect information. The LDA topic model, which combines the first and the second RNN classifiers, generates a set of key product defect topics for a product using OCRs.

The input data consists of 9000 randomly selected OCRs, and their star ratings, from the furniture section of the Amazon's Customer Review Public Dataset. This customer review dataset will be used as input for the RNN classifiers.

1.8 Research Limitations

In this research, while defective products are identified, it is limited by the following factors:

- The use of neural networks on textual data is a relatively new area in product defect management and different kind of neural network architectures have been proposed. The performance of the model is limited by the chosen one-layer Long short-term memory (LSTM) RNN algorithm.
- The performance of the model is limited on the number of input OCRs. This study only includes the randomly selected 9000 OCRs from the furniture section of Amazon's Customer Review Public Dataset as training and testing dataset. While the performance of the model may improve with more input OCRs, to manually

create the supervised learning OCRs dataset was not within the timeframe of this praxis.

- The scope of this research is limited to the home furnishing industry. The study only drew data from the furniture section of Amazon's Customer Review Public Dataset. The same model may be deployed to other industries for identifying defective products for future experiments, but this was not within the scope of this research.

1.9 Organization of Praxis

This Praxis consists of five chapters, as follows:

Chapter 1 begins with the background and the research motivation, then continues with the research objectives, questions, and hypotheses. It ends with the scope of research and the limitations of this research.

Chapter 2 provides relevant literature reviews that begin with risk management, and engineering recalls managements. Then, it covers methods of how companies and engineering teams mine customer opinions and what actionable steps they take in response. It is supplemented by a review of Natural Language Processing algorithms on sentiment analysis and classifying textual data, as well as the LDA topic model. The Chapter ends with a review on RNN for sentiment classification along with other neural network algorithms.

Chapter 3 presents the three methods used in this research and testing of the hypotheses. These methods include two quantitative RNN classifiers for identifying negative reviews and reviews containing defect information and the one LDA topic model for generating key product defect topics.

Chapter 4 covers the results and analyses of the statistical methods presented in Chapter 3.

Chapter 5 closes the praxis with a discussion of results and a conclusion. It also includes discussions of the contributions to the body of knowledge and recommendations for future research in the area.

Chapter 2—Literature Review

2.1 Introduction

This chapter provides a comprehensive literature review of the topics related to product defect management, opinion mining with customer reviews, as well as Natural Language Processing (NLP), and neural network algorithms. The purpose of this review is to provide a summary of the research that has been published on these topics and to analyze the existing body of technical knowledge.

The chapter begins with a review of the scholarship pertaining to risk management and product defect management. Risk management is one of the most important topics in engineering management and system management because it includes identifying, analyzing, treating, and monitoring the risks, so as to detect any defects in the product lifecycle. This section also offers reviews on the existing proactive risk management methods currently used.

The chapter then continues with analysis of literature regarding opinion mining with online customer reviews (OCRs). Companies have been collecting customer feedback and opinions, mainly for product improvement and marketing purposes. This section provides a review on the existing methods used by companies to collect customer opinions, especially OCRs and the actionable steps taken when processing this information.

This chapter will also give a review of current research on textual data processing as well as text classification. OCRs are textual data that need to be processed before being applied to sentiment analysis algorithms. This section covers research on the data cleansing process as well as traditional methods of sentiment analysis, such as traditional Naive Bayes and support-vector. This section also includes a discussion on Latent Dirichlet Allocation (LDA) topic model, an unsupervised learning method for extracting OCRs information.

Next, the chapter provides a review of current research on neural networks, especially the Recurrent Neural Network (RNN) approach which is in use in this research. This section examines scholastic material about the architecture of RNN as well as technical analysis on the neural network.

The chapter concludes with a summary of findings and potential implications of the current study.

2.2 Risk Management and Product Defect Management Reviews

2.2.1 Risk Management Process

In system engineering and engineering management, managing associated risk is an essential topic in enabling a successful development process. According to the Department of Defense (DoD) Risk, Issue, and Opportunity Management Guide, risk is the combination of 1) the probability of an undesired event or condition, and 2) the consequences, impact, or severity of the undesired event (DoD, 2015). In order to proactively address risk in the field of engineering, the International Council on Systems Engineering (ICOSE) and the DoD have established and standardized a series of risk management processes that became one of the INCOSE systems engineering life cycle technical management processes designated as ISO 15288 (Walden, 2015). In International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) 16085:2006, a risk management process includes the following steps: risk planning, risk profile management, risk analysis, risk monitoring, risk treatment, and risk management process evaluation (IEEE, 2006). Risk management processes continue throughout the life cycle of a product or service in order to reduce risks to an acceptable level.

In ensuring a successful risk management process, Hubbard and Witty suggested that risk management models and methods should be empirical, verifiable, and have a direct impact on business outcomes (Hubbard, 2009; Witty, 2015). In 2016 and in conjunction with Avaya Labs Research, a global provider of business communication and collaboration systems, Hackbarth et al. proposed a similar empirical proactive risk management plan named the data-driven software quality improvement method to reduce software risks in the software industry. This method includes prioritizing development resources and assigning sole ownership on the high-risk areas, and it develops risk indicators to monitor effectiveness. It was successfully implemented and empirically verified at Avaya. With this implementation, the average Customer Quality Metric (CQM) dropped from 2.9 percent to below 1 percent, and the average Implementation Quality Index (IQI) improved by 50 percent between 2012 and 2014 (Hackbarth et al., 2016).

Hackbarth et al. also found that while most defects were found and fixed in the product development and testing processes, customer feedback was also an essential source of defect findings. It accounted for less than 1 percent of the customer service requests.

2.2.2 Product Defect and Product Recalls

In today's product life cycle, companies like Avaya that use advanced quality assurance tools and techniques find it difficult to make perfect products, and these products often need to be recalled. Due to increased product complexity and more stringent product safety legislation (Dawar et al., 2000), studies show that the trend of product recall events is increasing and no industry is immune from a product recall event (Gallozzi, 2007; De Matos, 2007). In the auto industry, analysis of car recall data shows every carmaker has experienced recalls and the top eight manufacturers of the recalled products have contributed about 70% of total recalls during the last ten years (Ahsan, 2013). Recalls happen when the manufacturer does not address the issue or was unaware of it before the product was distributed to the market. The main two reasons for recalls are 1) a consequence of design flaws, which make the product fail to meet required safety standards, and 2) manufacture defects in products that do not conform to specifications such as poor craftsmanship (Lyles et al., 2008). According to Beamish (Beamish et al., 2008), in the toy manufacturing industry, design flaws contributed 70.8% of the recalls, while only 12.2% of the recalls were from manufacture.

These defective product recall events can cost a company billions of dollars and bring significant impact that lasts for an extended period. The U.S. Consumer Product Safety Commission (CPSC) has concluded that deaths, injuries, and property damage from consumer product incidents cost the nation more than $1 trillion annually (CPSC, 2019). An excellent example of a single product recall event is the Toyota vehicle recall in 2009 due to the floor mat and accelerators problems. It cost the company up to $5 billion, including substantial litigation fees and marketing efforts (Sanchanta et al., 2010). In the electronics industry, the recent 2016 Samsung Galaxy Note 7 explosions recall is also an influential one. This event cost Samsung more than $5 billion and resulted in subsequent loss of sales (Jeong, 2017). An empirical, event-time analysis found that product recall events have a direct impact on the company's equity price for two months after the events' release (Pruitt et al. 1986). These events expose companies not only to economic damages, but also negative consequences such as loss of goodwill, loss in product liability suits, and loss to their rivals (Jarrell et al., 1986).

2.2.3 Time to Recall and Product Recall Strategy

The strategy and the timing for recall of a defective product has a direct impact on both the finances and reputation of a company. Time is an essential factor during a defective product recall event. The longer the defective product remains in the marketing and distribution process, the harder it is for the company to take recovery action, and poses more potential for injuries (Berman, 1999; Hora et al., 2011). A study analyzing over 500 product recalls for 15 years showed that recall strategy adopted by the company, the type of product defect, and the supply chain entity that issues the recall has a direct impact on the time required by the company to make a recall to a product. The study showed that companies with a preventive recall strategy in place, such as continuously identifying product defects and initiating voluntarily recalls, have a shorter time to recall than companies with a reactive approach such as initiated recall after a hazard is reported. The same study also suggested that it takes manufacturers longer to issue a recall due to design flaws because of traceability issues (Hora et al., 2011).

There has been research proving a positive customer and public relations impact on the company will result when a proactive recall strategy is implemented during a product defect event (Mango., 2012, Siomkos et al., 1994, Souiden et al., 2009). Magno surveyed 217 subjects and suggested that when a company takes responsible actions during product defect event, it resulted in positive sentiment on the post-recall brand attitude from the customers. In contrast, companies that chose to assign blame and make an opportunistic response harm their post-recall brand attitude (Magno., 2012). Siomkos et al. conducted a study with 384 people on customer perceptions of product recall events and showed that consumers appreciate company transparency, and that their future purchases would be less negatively influenced when companies took improvement campaigns for defective products or voluntarily initiated a product recall on defective products (Siomkos et al., 1994). Another study involving 573 subjects also suggested that companies that voluntarily initiate a product recall during a defective product event, rather than denying a recall, would experience a positive impact on the manufacturer's image. (Souiden et al., 2009).

While it is hard to avoid the existence of defective product risk, the aforementioned studies have shown that time to recall and company's recall strategy have a direct impact on the company's reputation and losses, which supports the fact that when a company has an early view of product defect, it can reduce the loss of business and damage to reputation, and regain consumer trust.

2.3 Opinion Mining with Online Customer Reviews (OCRs)

With the rise of Web 2.0, social media and customer review sites have enabled companies to discover consumer feedback on their products with increased speed and accuracy. Information embedded in CORs has a direct impact on companies and their products (Karakaya et al., 2010). Comparing to "Offline" word of mouth customer opinions, OCRs have a much more significant impact because of its persistent, easily accessible, and open-to-public format (Dellarocas el at. 2007). Companies have been looking at OCRs to improve their product and marketing strategies (Barton, 2013). OCRs enable companies to monitor customer concerns and complaints, as well as to take corrective actions (Karakaya et al., 2010). Some companies even respond to these customer text reviews personally to improve their service (Chan, 2011). There are three areas in which companies are examining and actively researching customer reviews: 1) marketing communications and customer behavior; 2) future product design and features prediction; and 3) product defect and recall management.

2.3.1 Marketing Analysis

Much research done today in the field of business and marketing shows that OCRs and word of mouth opinions have a direct impact on marketing communications and customer behavior, such as buying patterns and sales. Ho-Dac et al. suggested that OCRs and product sales complement and reinforce each other. Their 2013 study on 78 different products with 3341 OCRs suggested that positive or negative online reviews respectively increase or decrease sales and create a feedback loop to create more either positive or negative online reviews (Ho-Dac et al., 2013). Karakaya et al. in a survey of 320 people also found that there is a strong positive relationship between online consumer engagement and future consumer purchases (Karakaya et al., 2010). Wei found that OCRs have a more significant impact on customers' memory, search, and share attitudes for products than celebrity endorsements on consumers' purchasing behavior (Wei, 2012). Elwalda et al. from a survey of 498 responses found customer trust and their future purchases were impacted by the usefulness, enjoyment, and sense of control derived from online customer reviews (Elwalda et al., 2015). A 2008 survey of 482 responses suggested that information gained from OCRs had more influence on buying decisions than speaking with friends (Steffes et al., 2008). To further demonstrate value in OCRs, Lee et al. proposed a natural language processing (NLP) model that can aid companies in collecting customer options for marketing research by phrasing and categorizing text in OCRs (Lee et al., 2011). These studies have shown that information embedded in online customer reviews has a direct impact on customer behavior, and companies have been using it to develop marketing strategies.

2.3.2 Future Product Design

Companies use information embedded in OCRs to discover customer needs on products and to make improvements for future products. In the field of business management and engineering design, the knowledge of Customer Requirements (CRs) is an important piece of information for companies to determine customers' needs and requirements on products, and to formulate their strategies accordingly (Shahi, 2006). A small number of pre-formatted survey data to obtain this CR information has been used for a long time. With the growth of computational power and larger volume of data, there is more active research on using new NLP algorithms to transform the large volume of OCRs' information into CRs and engineer characteristics (ECs) for future product development (Jin et al., 2019). Zhang et al. have constructed a novel product comparison network with text sentiment mining to show how products compare with each other from OCR data. This network provides companies with insight on where their products compare with other similar products on the competitive landscape (Zhang et al., 2013). Chaklader et al. have proposed a novel algorithm using weighted phrase ratings on OCRs to provide customers' feedback on CRs. This method enables companies to determine if the product design is in line with a large amount of customer feedback, such as “too tight” or “too loose” on a single headphone product (Chaklader et al., 2017). Suryadi et al. have proposed a novel algorithm using a trained neural network model to provide product usage context on a single product using OCRs. This algorithm enables product designers to observe CRs, to improve the product, and to show possible marketing opportunities in terms of usage context (Suryadi et al., 2019). Jiang et al. have proposed an algorithm using the fuzzy time series forecasting method on different periods of OCRs for engineers to determine current and future importance weights of product features. This method enables engineers and product designers to determine the importance of different product features during the product design stage. (Jiang et al., 2017). El-Dehaibi et al. using customer online text reviews proposed an algorithm using the NLP method s to predict if the customers’ perception of sustainability on a product is line with the reality (El-Dehaibi et al., 2019). Zhou et al. proposed an algorithm using the Latent Dirichlet Allocation (LDA) method on OCRs to analyze customer needs for product ecosystems (Zhou et al., 2019). These studies have shown that information embedded in OCRs has a direct impact on CRs and customer needs.

2.3.3 Product Defects Discovery

While there are a number of research studies and proposed algorithms for using the previous generation products’ OCRs to provide valuable information to engineers on next generation of product development and product design, there is little attention in academia for using OCRs in the later stages of the product cycle to discover customer complaints and product defects (Abrahams et al. 2012). Abrahams et al. proposed a new algorithm using a sentiment analysis method to classify the type of product defect information (e.g., performance defects, safety defects, non-defect, etc.) embedded in OCRs for vehicles (Abrahams et al., 2012). Abrahams et al. recognized that while traditional sentiment analysis methods can successfully identify complaints in other industries, they fail to distinguish defects from non-defects, and safety from performance defects in the automotive industry. This is because OCRs in the auto industry that mentions safety defects have more positive words, and fewer negative words and subjective expressions than other OCRs. Alternatively, their team spent 11 weeks building and tagging a set of automotive 'smoke' words dataset from the OCRs before doing sentiment analysis. This method has shown success in defect discovery and classification, but it is also highly domain-specific for the automotive industry. Bleaney et al. studied and compared the performance of various classifiers, including Logistic Regression, k-nearest neighbors (k-NN), Support-Vector Machines (SVM), Naïve Bayes (NB) classifier , and Random Forest (Decision Tree) on identifying safety issues ("Mentions Safety Issue," "Does Not Mention Safety Issue") embedded in OCRs in the baby product industry (Bleaney et al., 2018). They found that the Logistic Regression classifier had the highest precision, with 66% in the top 50 reviews surfaced. Zhang et al. proposed an unsupervised learning algorithm using the LDA topic model's method to group and identify key information and words in each type of defect from complaints and negative reviews (Zhang et al., 2019).

While these studies have shown some success in using sentiment analysis methods to extract defective information from OCRs, these studies have not been able to identify individual defective products with OCRs from a product level or accept all OCRs from a single product.

2.4 Natural Language Processing (NLP)

In the field of linguistics and computer science, there is active and ongoing research on how to improve the ways in which computers understand human behavior and language. Research was started as early as in the 1950s, when Alan Turing introduced what is now known as the "Turing test" to determine if a machine can exhibit human intelligence (Turing, 1950). According to the Turing test, artificial intelligence is said to be achieved when human judges cannot differentiate a series of dialogues given by a human from those generated by a machine. While there has not been a machine widely acknowledged as being able to achieve such a test (Landgrebe, 2019), the Turing test can be recognized as the starting point of artificial intelligence research in the academic discipline. The development of Natural Language Processing (NLP), which is a type of artificial intelligence concentrated on understanding and manipulating human language, has achieved practical successes over the years. NLP models have successfully helped researchers solve real-world human text processing problems, including most of the studies found in section 2.3, especially research on using OCRs to extract product defect information (Abrahams et al., 2012; Bleaney et al., 2018; El-Dehaibi et al., 2019). There are three main steps where the NLP data pipeline can turn raw text data into useful information: 1) Text Preprocessing and Cleansing; 2) Text Representation and Word Embeddings; and 3) Sentiment Analysis and Text Classifier.

2.4.1 Text Preprocessing and Cleansing

Text preprocessing and cleansing is an essential step in any data processing for inputting accurate data, and it is still an active research area in the field of data mining and NLP. Missing, inconsistent, and noisy data can contribute to inaccurate data analysis (Han et al., 2000, p. 84). Human-created text data in real life, such as emails, blogs, and online chat data, often contain misspellings, special characters, and contractions, which is not ideal for a computer to process (Agarwal et al., 2008; Tang et al., 2005).

Dirty data can impact the performance of text analysis. Over the years, researchers have been proposing methods to clean and process data before feeding them into computation algorithms. Tang et al. suggested using non-text filters to remove parts that are not needed for text mining and using text normalization layers to convert necessary text into a canonical form to clean text data. Their research on human-created email text data suggested that non-text filters and text normalization layers can improve SVM classifiers' accuracy on detecting a start line and end line in an email, compared to those without (Tang et al., 2005). This SVM Classifier finds the optimally separating hyperplane on the word positions for the start line and end line in an email. On spelling correction, Golding et al. recognized Bayes and Winnow's statistical learning methods can successfully fix misspell situations on valid words, such as substituting "casual" for "causal" (Golding et al., 1999). Wan et al. proposed a novel data cleansing algorithm with SVM that can select and remove the most confidently-identified noisy instances to clean input data and to increase sentiment classifier accuracy (Wan et al., 1999). On a linguistics level, Dickinson et al. and Abney et al. both proposed improving data quality by using statistical learning methods to detect and repair part-of-speech annotation on the text data (Dickinson et al., 2003; Abney et al., 1999).

Dirty data can have an impact on the performance of text classifiers. Agarwal et al., have compared human-created text data using both SVM and NB text classifiers with different levels of noise. The authors found that SVM outperforms NB classifiers on noisy human text data, but SVM required more time and resources on model training. Agarwal et al. also found that noisy data affects model accuracy, text data with 40% feature noise does not affect text classification accuracy as much (Agarwal et al., 2008). Kreek et al. confirmed this result when comparing different levels of noisy text data versus clean data using different SVM, Convolutional Neural Network (CNN), and fastText classifiers. Their research showed that clean data outperformed noisy data by 11% at a 25% noise level, and 25% at a 50% noise level; noisy text data up to a 50% noise level can still be used to create high-quality text predictions (Kreek et al., 2018). While Agarwal et al. and Kreek et al. agree that light data noise does not affect text classifiers performance as much (Agarwal et al., 2008; Kreek et al., 2018), methods and techniques of processing and cleaning text data is still an essential topic in NLP.

2.4.2 Text Representation

Before computer algorithms can analyze raw text data, the data has to be turned into text representation for computers to recognize. Numerous studies have examined the two main approaches for turning raw text data into text representation in the field of NLP: 1) one-hot encoding; and 2) word embeddings.

The one-hot encoding method uses one binary code to represent each categorical data unit. This encoding scheme maps each feature, or word, in binary columns one at a time

(Beck et al., 2000). While the majority of multi-class logistic regression models use the one-hot encoding method for its simplicity, this method creates high dimension data from sparse features, especially for textual data with a huge number of vocabularies (Zhang et al., 2016), which would dramatically increase computational complexity.

Word embedding is another widely adopted method in representing raw textual data to its low-dimension property. This encoding scheme transforms each word into a set of meaningful, real-valued vectors (Rezaeinia et al., 2019). Instead of randomly assigning vectors to words, Mikolov et al. and Pennington et al. have created the two most widely-used pre-trained datasets among researchers for mapping words to vectors, GloVe, and Word2Vector, respectively (Mikolov et al., 2013; Pennington et al., 2014). These datasets map words which have closer English meaning to a closer vector scale in a general sentiment analysis task. An example would be mapping the term “king” in a closer scale to “man” in vector, while further from "woman." Other studies have also built models on top of these two datasets to enhance word embedding in domain-specific sentiment analysis tasks (Rezaeinia et al., 2017; Ren et al., 2016; Tang et al., 2014).

2.4.3 Sentiment Analysis Models and Text Classifiers

Sentiment-analysis, along with text classifiers, became available in the 2000s with the rise of machine learning methods in NLP and the availability of datasets on which to train machine learning algorithms (Pang et al. 2008). As suggested by Nasukawa et al. in 2003, sentiment analysis is “capturing favorability using natural language processing." They proposed a framework for identifying whether the text indicates positive or negative opinions towards the subject using the POS tagged positive or negative sentiment terms that exist in the text (Nasukawa et al., 2003).

Text classifiers not only can classify negative or positive options but can also classify text or documents to assigned topics. A Bag-of-Words (BOW) model, along with the Term Frequency–Inverse Document Frequency (TF-IDF) approach, is a commonly used classification model to retrieve text or document information for its simplicity. The Bag-of-Words (BOW) approach puts the existing words from a text or document into a group representation, disregarding word orders. TF-IDF approach then determines the classification using the frequency of these words in that group and is able to offset terms that appear too often (Zhai et al., 2016, p. 93). Aggarwal and Zhai have also presented other commonly used statistical classifiers (Table 2-1) that are widely used in classifying textual data (Aggarwal et al., 2015, p. 289).

Decision Trees	Decision trees use the hierarchical division of underlying features, or words, to create class partitions for classification. For a given word, decision trees determine the partition that the word most likely belongs to and uses it for the purpose of classification.
Pattern (Rule)-based Classifier	A Pattern-based Classifier constructs a set of word pattern rules corresponding to class labels and uses these rules to determine which class label the text will most likely be in for classification.
SVM Classifier	An SVM Classifier uses linear or non-linear delineation to determine the optimal boundaries between different classes

	in the data space. It uses different partitions of the data space to determine the classification.
Neural Network Classifier	A Neural network is a set of units called neurons that take features or words as input and trains on loss function to determine the optimal boundaries between different classes in data space for classification.
Bayesian Classifier	A Bayesian classifier contracts a probabilistic model with the posterior probability of the text belonging to a class to determine the classification for unseen text.

Table 2-1. Text Classification methods

Aggarwal et al. also suggested that most text classification tasks are different from typical statistical classifications tasks in that textual data has an attribute of high spares and low frequencies (Aggarwal et al., 2015, p. 289). It is critical to select a classifier that accounts for these textual data features. In the context of classifying defective product with customer text reviews, Bleaney et al. have compared tradition Logistic Regression, k-nearest neighbors (k-NN), SVM, NB, and Random Forest (Decision Tree) on OCRs classifier in discovering safety issues in baby products and included using a neural network as a suggestion for future work (Bleaney et al., 2018).

2.4.4 Latent Dirichlet Allocation (LDA)

Another unsupervised NLP method that can discover textual data insight is the Latent Dirichlet Allocation (LDA). LDA is a three-level hierarchical Bayesian model that is able to discover a set of unobserved groups, or topics, that best describe a large collection of observed discrete data. As Blei et al. suggested, the goal of LDA is to “find short descriptions of the members of a collection that enable efficient processing of large collections while preserving the essential statistical relationships that are useful for basic tasks, such as classification, novelty detection, summarization, and similarity and relevance judgments." (Blei et al., 2003). LDA was first proposed and used for discovering population genetics structure in the field of bioengineering in 2000 (Pritchard et al., 2000) and further used in NLP processing on textual data in 2003 (Blei et al., 2003).

Researchers have been using LDA to discover topics from a large collection of OCRs to help industry to gain insight for their product. Santosha et al. and Zhai et al. both suggested using LDA for grouping and producing an effective summary of product features from a large collection of OCRs. Santosha et al. successfully used the product features terms from LDA topic model to build a Feature Ontology Tree for showing product features relationships (Santosha et al. 2016), while Zhai et al. built a semi-supervised LDA with additional probability constraints to show product features linkage between products. Researchers have also suggested LDA topic model can be used on serving industries to discover business insight such as a summary of OCRs on travel and hospitality review sites. Titov et al. and Calheiros et al. both have suggested using LDA outputted topic's terms to discover and analyze customer reviewer's sentiment for businesses to improve customer experience (Titov et al, 2008; Calheiros et al., 2017). In the field of product defect management, Zhang et al. used LDA topic model to discover short summaries of product defects and solutions on a large amount of online product negative reviews and complaints to help engineers and customers to discover product defect information (Zhang et al., 2019).

2.5 Neural Network Architecture and Recurrent Neural Network (RNN)

A neural network is a set of connected computational input/output units that loosely model the biological brain (Han et al., 2012, p. 398). A neural network is able to be trained iteratively with supervised data, can recognize or "Learn" specific patterns embedded in this data, and perform prediction tasks based on these learned patterns without pre-programmed rules. Neural networks are helpful in clustering, classification, and prediction, and are currently being used in a wide variety of applications including speech recognition, computer vision, and face recognition (Abiodun et al., 2018). In the context of this research, the neural network plays an important role in solving text classification problems, especially when using recurrent neural network architecture (Liu et al., 2016). While the idea of this model dates back to 1943 with McColloch’s computation model for neural network (McColloch et al. 1943), the practical model was not put into use until recent years due to an increase of computational power and the findings on backpropagation methods.

2.5.1 Neural Network Layers and Activation Functions

The heart of a neural network is a set of interconnected individual neurons with associated weight and activation functions to determine output decisions. Each layer is made up of multiple neurons to process data. Each neuron is assigned to a random weight that is summed to one in its layer. A typical neural network architecture (Figure 2-1) consists of three connected layers: an input layer, a hidden layer (or multiple hidden layers), and an output layer, as follow:

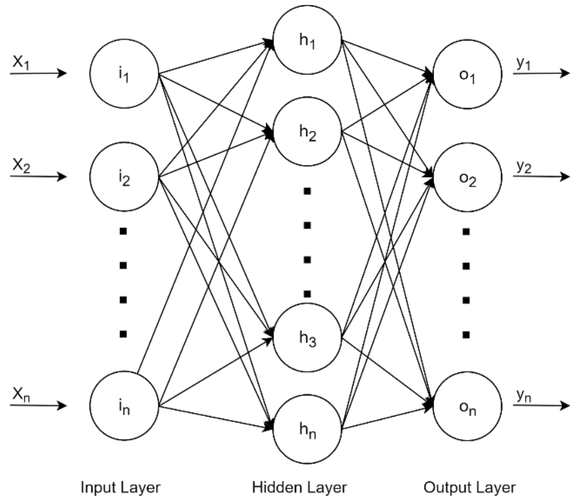


Figure 2-1. Simple Neural Network Architecture

Input data is fed into the neural network by the input layer. Input data would be the word embedded in vector representations in textual classification. In the input layer, neurons take a set of training input vectorized data, multiply it with its associated weight, and pass on to the neurons in the intermediate hidden layer. In the hidden layer, neurons accept the weighted input vector from the input layer and compute output with activation functions. There can be multiple hidden layers, depending on the network architecture. The weighted output then passes to another hidden layer (if there are any) or to output layer for making predictions with activation functions.

Activation functions are mathematical equations that define the output of a neuron based on its relevancy for the networks' prediction. Non-linear and differentiable mathematical equations are typically used to allow backpropagation optimization to model classification problems that are linearly inseparable (Han et al., 2012, p.403). Sigmoid function is a type of activation function that is heavily used in the field of machine learning before the induction of a Rectified Linear unit (ReLu). Given the net input value I_j to neuron unit j, then O_j, the output of neuron unit j, is computed as (Han et al., 2012, p.403):

$$O_j = \frac{1}{1 + e^{-I_j}}$$

Equation 2-1. Equation of Sigmoid Function (Han et al., 2012, p.403)

Sigmoid function is excessively used as an activation function in the hidden layer for the property of taking input ranged in [-Inf; +Inf] and separating the output decision boundary to either closer to a 1 or a 0, as shown in Figure 2-2.

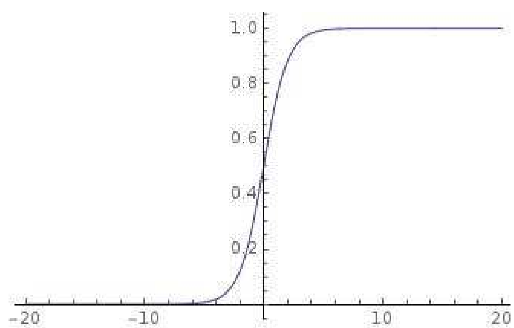


Figure 2-2. Plot of Sigmoid Function

The sigmoid function is also ideal to be used as an activation function in the output layer for two-class classifiers due to its binary separation property (Goodfollow et al., 2016, p. 176). For multi-input classification, Bridle first suggested using the Softmax function in the output layer (Bridle, 1990). A Softmax function is a generalization of the sigmoid function and has a property of normalizing the input vector into a probability distribution over multi-classes that sums up to 1 (Goodfollow et al., 2016, p178). Softmax and sigmoid functions can also be used in the hidden layer, depending on the architecture.

Another commonly used activation function is the Tanh (Equation 2-2), or hyperbolic tangent, activation function. It is heavily used as an activation function inside the RNN LSTM internal cell for updating cell information and output current cell information. The main difference between sigmoid function and Tanh function is that Tanh function separates the output decision boundary to either closer to a 1 or a -1 rather than 0 for sigmoid function. Given the net input value I_j to neuron unit j, then O_j, the output of neuron unit j, as shown in Figure 2-3.

$$\mathrm{O}_j = \frac{e^{I_j} - e^{-I_j}}{e^{-I_j} + e^{I_j}}$$

Equation 2-2. Equation of Tanh Function (Goodfollow et al., 2010, p168)

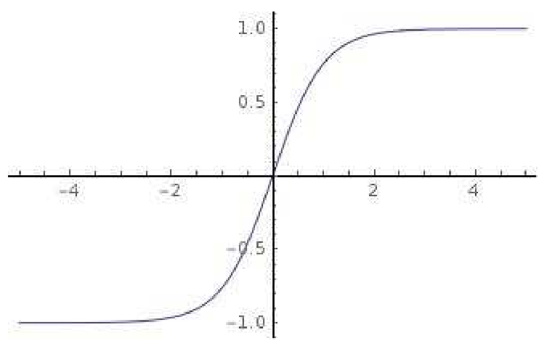


Figure 2-3. Plot of Tanh Function

Jarrett et al. have presented, and was further supported by Nair et al., the non-linear (ReLu) in the late 2000s to improve model learning. Given the net input value I_j to neuron unit j, then O_j, the output of neuron unit j, is computed as:

$$\mathrm{O}_j = max(0,\ I_j\ )$$

Equation 2-3. Equation of ReLu Function (Goodfollow et al., 2010, p168)

Comparing with sigmoidal functions, the ReLu function is easier to train and preserve information, with less vanishing gradient problems (Jarrett et al., 2009; Nair et al., 2010; Goodfollow et al., 2010, p168). The output of ReLu function is the input value, if positive, or else zero, as shown in Figure 2-4:

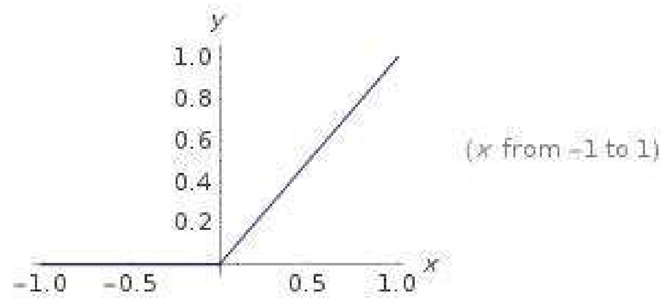


Figure 2-4. Plot of ReLu Function

2.5.2 Gradient-Based Learning and Backpropagation

The most important part of the neural network is the property of being able to update the weights and biases in the neurons over training to make desire outputs. In 1975, Werbos et al. first proposed error propagation, also known as backpropagation, on loss function (Werbos et al., 1975) and put it into practice in the 1980s for training multi-layer neural networks (Rumelhart et al., 1986). Loss, or cost, function is an action that calculates the difference of predicted output values of the neural network with respect to the representation of the provided correct labels to measure how well the neural network is performing. The hope is to slowly minimize this loss function and manipulate the training data by updating the weights and biases in neurons. The backpropagation algorithm allows the erroneous information to transfer back to the hidden layer by computing a derivative of the overall error gradient of the loss function (Russell et al., 2011, p735). The term gradient descent refers to the overall iterative backpropagation process to update the neural network optimization algorithm for finding the local minimum of the loss function. In order to avoid optimization getting stuck at the local minimum and being able to control the learning step size, a learning rate parameter between 0 to 1 is usually applied to the update function (Han et al., 2012, p. 403). The whole process of gradient learning is completed when gradient descent reaches the minimum of the loss function, which is ready and optimized for classification.

2.5.3 Recurrent Neural Network

Based upon the Neural Network architecture, researchers in the 1980s have been proposing adding recurrent connections between nodes in order to solve problems involving sequential data, which is now called the Recurrent Neural Network (RNN) (Rumelhart et al. 1986; Pearlmutter, 1989). While this type of network can solve sequential recognition, Bengio et al. found that it is difficult to solve problems where the sequences are getting longer and prediction depends on input presented at an earlier time (Bengio et al., 1994). This is due to vanishing gradients where the error gradient propagating back tends to vanish in time (Hochreiter et al., 1998). Hochreiter et al. also proposed the Long Short-Term Memory (LSTM) approach, a particular type of RNN architecture, which further improved the problem involving long data sequences. LSTM overcomes this problem by adding gates in RNN nodes to regulate the flow of information (Hochreiter et al., 1998). Researchers suggested that LSTM can greatly improve the accuracy of sequence learning, such as offline handwriting recognition (Grave et al. 2011), as well as text classification problems that involve word order (Liu et al., 2016). An LSTM-RNN model can build the relations between sentences in semantic meaning on text classification, which can increase the model accuracy over that of the traditional methods (Tang et al., 2015; Li et al., 2015).

2.6 Summary and Conclusion

This literature review section provided a comprehensive examination of journal papers, conference papers, and books on product defect management, the usage of customer text reviews by companies and engineering teams, and the tools for analyzing textual data with a focus on RNN and LDA. Based on the literature review, manufacturing companies are still facing huge product recall problems even with modern quality assurance technologies and tools, and online customer opinions often get overlooked. Although the literature review has identified a few methods of using OCRs to identify product defect information, these methods are either highly domain-specific, accepting only certain negative OCRs, or identifying defect information only at the OCRs level. This study introduces the newly mature RNN and LDA method to provide solutions for identifying defective product insights and bridging the knowledge gap between product defect management, customer feedback, and neural networking.

Chapter 3—Methodology

3.1 Introduction

The goal of this study is to develop and evaluate how the recurrent neural network (RNN) classifiers and Latent Dirichlet Allocation (LDA) topic model can extract product defect information from online customer reviews (OCRs). To achieve this, multiple data processing steps and models were developed. The chapter begins with a discussion of the raw data collection process, along with the manual data labeling process for a supervised learning dataset. The chapter continues with the data preprocessing methods, which includes cleaning the OCRs' textual data and turning textual data into a digital, computer-readable format. Next, the chapter examines the development of recurrent neural network (RNN) classifiers and the Latent Dirichlet Allocation (LDA) topic model for extracting defect information. The section concludes with the analysis methods used for testing the hypothesis. Figure 3-1, shows the end-to-end process as to how OCRs from a single product is processed to extract product defect insight using the fully trained and built models.

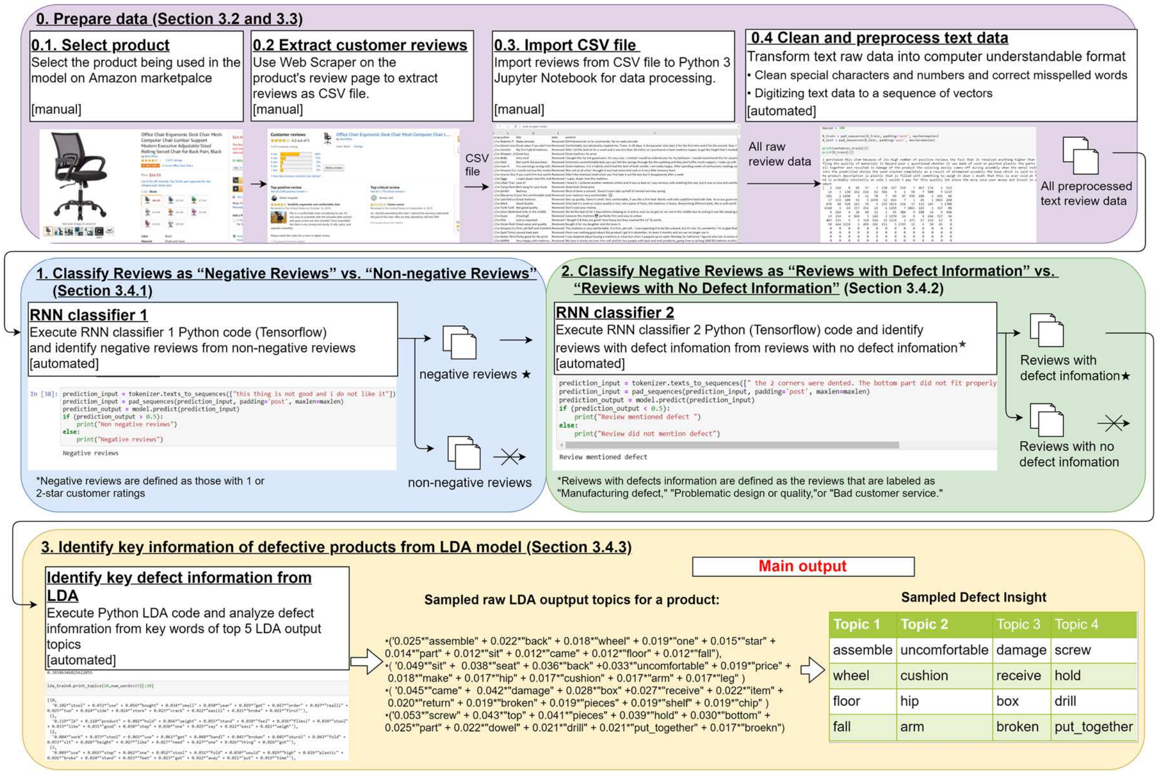


Figure 3-1. End-to-end Process

3.2 Data Collection and Data Labeling

The primary data source of this study is the OCRs and their associated metadata from the Amazom.com marketplace. This dataset contains customer text reviews, product information, star ratings, review dates, and other relevant information. The customer reviews dataset will be used as input for sentiment analysis to identify negative customer reviews, determine their usefulness on providing information about defects, classify the types of defects found in the OCRs, and extract product defect topics within the context of this research.

The Amazon's Customer Reviews Public Dataset is an organized version of OCR data in a number of tab-separated values (TSV) files for researchers in the fields of Natural Language Processing (NLP), Information Retrieval (IR), and Machine Learning

(ML). This data set contains more than 130 million OCRs and associated metadata in 43 product categories in the U.S. marketplace from 1995 until 2015.

3.2.1 Furniture Customer Reviews and Data Format

The furniture section subset of this dataset was used in this study. This data subset contains 792,114 OCRs and associated metadata of customer options on Amazon furniture products. Due to the constraint of manual labor in creating a supervised dataset, this study randomly selected 3000 Amazon OCRs with a rating of three or more stars and 6000 Amazon OCRs with a rating of one or two stars for model building, training, and testing. Each row of data contains the data columns shown in Table 3-1. A sample data of this dataset is shown in Figure 3-2. The focus of this study is the textual analysis in the "review_body" column of the dataset.

Data Column	Data Column Information
marketplace	Two letter country code of the marketplace where the review was written.
customer_id	Random identifier that can be used to aggregate reviews written by a single author.
review_id	The unique ID of the OCR.
product_id	The unique Product ID the OCR pertains to. In the multilingual dataset, the OCRs for the same product in different countries can be grouped by the same product_id.
product_parent	Random identifier that can be used to aggregate OCRs for the same product.

product_title	Title of the product.
product_category	Broad product category that can be used to group OCRs (also used to group the dataset into coherent parts).
star_rating	The 1-5 star rating of the OCR. (with 1 being the lowest, 5 the highest)
helpful_votes	Number of helpful votes.
total_votes	Number of total votes the OCR received.
vine	The OCR was written as part of the Vine program.
verified_purchase	The OCR is on a verified purchase.
review_headline	The title of the OCR .
review_body	The OCR text.
review_date	The date that OCR was written.

Table 3-1. Amazon Reviews Data Columns (Amazon, 2015)


```
{   "marketplace":      "US",
    "customer_id":      "52703681",
    "review_id":        "RZ0Y9U30658TB",
    "product_id"        "B00000ITPY",
    "product_parent":   "676928760",
    "product_title":    "Edutiles Foam Playmats Uppercase Alphabet 26 Pieces -
    Complete Alphabet Set",
    "product_category": "Furniture",
    "star_rating":      "5",
    "helpful_votes"     "5",
    "total_votes":      "6",
    "vine":             "N",
    "verified_purchase":"N",
    "review_headline":  "Educational AND fun!",
    "review_body":      "My son and I BOTH loves these!  This toy is extremely
    versatile.  One minute they are a floor mat, the next minute a puzzle, the
    next he makes them into blocks to stack or he's lining up the letters and
    practicing the  alphabet.  Somewhat expensive, but worth the investment.",
    "review_date":      "2000-03-17"}
```


Figure 3-2. Simple Amazon Raw Data

3.2.2 Data Labeling

Successful machine learning models are built on large volumes of high-quality training data. In order to build RNN models that can extract defective information embedded in each OCR, each OCR needs to be manually labeled by human labelers for RNN model training and testing purposes. The "Amazon Web Services (AWS) SageMaker Ground Truth" data labeling service was procured to manually label OCRs to build the required supervised dataset. AWS SageMaker Ground Truth provides a platform for independent labelers to label machine learning tasks, and each of the 9000 OCRs was reviewed by three human labelers to ensure the accuracy of the data. Human-labeled results are also generated with a confidence score for each label to ensure high-quality data.

Each of the OCRs was labeled with one of the following four labels: "Manufacturing defect," "Problematic design or quality," "Bad customer service," and "No defect information." The labels of "Manufacturing defect" and "Problematic design or quality" are identified as the two main reasons for a company to initiate product recalls (Beamish et al., 2008; Lyles et al., 2008). "Bad customer service" is also added as one of the labels because a customer service issue is also where customers often report issues, especially while shopping on an online platform. Labelers were provided with a detailed definition of each label with an example (as shown in Table 3-2) to categorize each OCR on the AWS SageMaker Ground Truth interface in Figure 3-3. The sample data output of AWS SageMaker Ground Truth is also given in Figure 3-4.

Label	Definitions of labels	Example

Manufacturing defect	OCRs that mention products that have manufacturing defects and/or are improperly manufactured with physical parts, apart from its intended design.	I was very happy with the purchase for the first 3 or 4 days. Then the bearing dropped out of the tube in the middle of it. (review_id: R238K8EITCNRZZ)
Problematic design or quality	OCRs that mention products that have a problematic design or quality issue, with no mention of physical parts falling off.	The knee cushion is not comfortable for sitting any longer than about 10 minutes. The chair is clunky and hard to move on the floor without picking it up. (review_id: R31BYJESH8F2DO)
Bad customer service	OCRs related to the frustration of delivery of the product or customer support process, while not related to the physical products themselves.	I have since called twice more with no returned call. This is the worst customer service I have ever received (review_id: R3VG1CFNR60ED)

No defect information provided	OCRs did not mention any of the defect information from the last three categories	We bought four of these, they are just some real cheap chairs that are overpriced. (review_id: R2F8RCR0LFI7SS)

Table 3-2. Definition of AWS SageMaker Ground Truth Labels

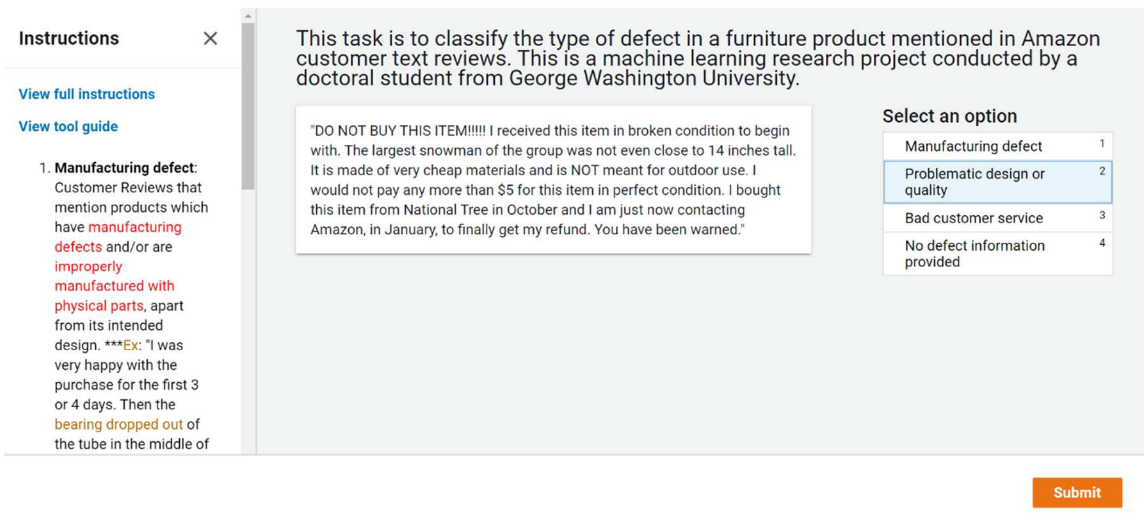


Figure 3-3. AWS SageMaker Interface for Labelers


```
{   "source":"The sliding keyboard drawer was delivered with 3 stripped screws
unable to fix the drawer and the part is not available for re order  until 3
months later. The product is beautiful but unfortunately the execution of the
construction failed.",
    "research-classify-defect-type-3":0,
    "research-classify-defect-type-3-metadata":{
            "confidence":0.95e0,
            "job-name":"labeling-job/research-classify-defect-type-3",
            "class-name":"Manufacturing defect",
            "human-annotated":"yes",
            "creation-date":"2020-01-31T06:32:47.504438",
            "type":"groundtruth/text-classification"
            }
}
```


Figure 3-4. Simple Output Data from AWS SageMaker

3.3 Data Preprocessing

Data preprocessing is an essential step in turning raw text data from OCRs into useful information that the computer can read, and machine learning can process. After the OCRs have been extracted from the review body, the OCRs were cleaned and preprocessed, and then turned into digitized text representation vectors.

3.3.1 Text Preprocessing and Cleaning

As mentioned in section 2.4.3, human-created text data often contains inconsistent wording, special characters, and contractions, which can contribute to inaccurate data analysis and affect model performance. (Han et al., 2000, p. 84; Agarwal et al., 2008; Tang et al., 2005). For this reason, text preprocessing and cleaning is an essential step in ensuring input data quality with normalizing words and removing unnecessary characters. The following three steps are taken in this study to improving text data quality: 1) Stemming, 2) Stop-word removal, and 3) Special character, numeric, and empty text removal.

Stemming is an NLP technique to group and reduce different words with the same root and linguistic meaning into the same word stem or root form. This study employed the Python Natural Language Toolkit (NLTK) package algorithms for the stemming process. Take the word "like" as an example. After this process, terms such as "likes," "liked," "likely," and "liking" would be reduced and unified into the word "like." In the later text representation steps, the computer would treat these different words with the same word stem as an equal text vector representation. This can increase machine learning model performance by treating words such as "likes" and "like" with the same word meaning.

Stop-word removal is an NLP technique to remove commonly used words in English language such as "the," "a," "at," and "is." These Stop-words give no or very little linguistic meaning to the overall context of the given text. To avoid losing the textual message in translation, this study used a custom-written Stop-word removal function. This process can reduce processing time and prevent machine learning models from leaning the training processing toward these frequently appearing Stop-words.

Special character, numeric, and empty text removal is an NLP technique to remove the non-text characters to improve data quality. Special characters include characters such as "!", "@", "#", and "$". Numeric characters include characteristics such as "0"-"9". Empty text includes tab, space, and Next-Line characters. This study employed a custom-written function to finish this process. This increases the machine learning model's performance by only focusing on real textual data.

The following two figures, Figure 3-5 and Figure 3-6, show an OCR before and after these three steps were done.

```
4 MDX boards were broken on arrival, which ostensibly will be replaced after
filling out a form on their website.  This indicates either bad packaging or
inferior quality of the MDX,.  It is a hassle as now I have to store the parts
while waiting for the proper parts to be replaced, and retrieve the old dresser
which was gotten rid of, to store the clothes,  in preparation of this new one
to be put together.  Cheap is what you get.
```

Figure 3-5. Simple OCR Before Text Preprocessing Step

```
mdx board broken arriv ostens replac fill form websit indic either bad packag
inferior qualiti mdx hassl store part wait proper part replac retriev old
dresser gotten rid of store cloth prepar new one put togeth cheap get
```

Figure 3-6. Simple OCR After Text Preprocessing Step

3.3.2 Text Representation

As mentioned in section 2.4.3, OCRs textual data have to be turned into text representation in a digitalized format inputting to machine learning models for computers to recognize the information. This study used the word embedding NLP technique for text representation.

This encoding scheme transforms each word into a set of meaningful, real-valued vectors to represent each word in a given text (Rezaeinia et al., 2019). Instead of assigning random numbers to the text representation vectors for each word, this study uses Word2Vector pre-trained datasets for mapping real number vectors to words. This pre-trained dataset maps words with similar linguistic meaning to a closer vector scale (Mikolov et al., 2013). This increases the efficiency of the training process for the machine learning model.

Text padding is the last step of the data preprocessing before the OCRs are fed to the machine learning model. While different OCRs have different word lengths, the word length of each input OCRs' text representation has to be unified for the RNN model to work properly. In this study, the padding step fills up text representations to a length of 500 vectors by adding '0's after the end of each text representation.

The following two figures, Figure 3-7 and Figure 3-8, show an OCR before and after word embedding and text padding.

```
mdx board broken arriv ostens replac fill form websit indic either bad packag
inferior qualiti mdx hassl store part wait proper part replac retriev old
dresser gotten rid of store cloth prepar new one put togeth cheap get
```

Figure 3-7. Simple OCR Before Text Representation Step

```
array([[ 367,  375, 1153, 1361,  480,  297, 2941,  286,  269,  710, 1660,
         269,  192,  688,  983, 1594,   10,  286, 1301,  162,   29,   37,
         193,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0]])
```

Figure 3-8. Simple OCR After Text Representation Step

3.4 Model Development and Testing Procedure

There were three targeted statistical models used in the research in order to accomplish the thesis statement of providing engineers with an early view of product defects. They include two quantitative RNN classifiers and one Latent Dirichlet Allocation (LDA) topic model. The first RNN classifier categorizes negative OCRs from non-negative OCRs. The second RNN classifier categorizes OCRs that mention defects from those that do not. The LDA topic model provides engineers a view on groups of word items or topics that best describe the type of defects found in a single product.

3.4.1 Model One: RNN classifier for classifying negative reviews from non-negative reviews

Table 3-3 shows the definitions of negative and non-negative OCRs. Defective information about a product is often found in negative OCRs as compared to non-negative OCRs.

Labels	Definition of Labels	Number of OCRs
Negative OCRs	OCRs with 1 or 2-star customer ratings	6000/9000
Non-negative	OCRs with 3 or more-star customer ratings	3000/9000

Table 3-3. RNN Model Labels for Negative and Non- negative OCRs Classification

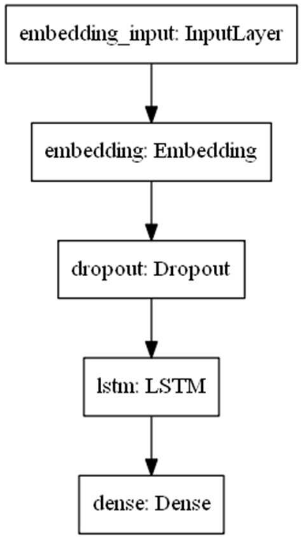
OCRs with negative sentiment have a much higher chance to include complaints and defective issues about products as compared to OCRs with a non-negative sentiment. In order to support this hypothesis, 6000 negative OCRs and 3000 non-negative OCRs were tagged using one of the following labels: “Manufacturing defect," “Problematic design or quality,” Bad customer service," or “No defect information provided”. OCRs with the label of "Manufacturing defect", “Problematic design or quality", and “Bad customer service” were considered as OCRs that provide defect information. OCRs with label of “No defect information provided” do not provide defect information. Total of 5817 out of 6000 negative OCRs provided defect information. Total of 476 out of 3000 non-negative OCRs provided defect information. By using a One-tailed Two-proportion Z-test for the difference of proportions with a p-value of 0, this suggested that negative OCRs provide more defect information then non-negative OCRs (see Table 3-4). A significance level of alpha equal to 0.05 was used throughout this praxis.

Prediction Statistics

Negative OCRs		Non-negative OCRs	
Sampled negative OCRs	6000	Sampled non-negative OCRs	3000
Provided defect information	5817	Provided defect information	476
Not provided defect information	183	Not provided defect information	2524
Sample $p_{\text{non-negative OCRs}}$	0.9695000	Sample $p_{\text{negative OCRs}}$	0.158667
One-tailed Two-proportion Z-test			
$H_0 : p_{\text{non-negative OCRs}} - p_{\text{negative OCRs}} = 0$ $H_1 : p_{\text{non-negative OCRs}} - p_{\text{negative OCRs}} < 0$			
Z-value	-115.33	P-value	0.000

Table 3-4. Two-proportion Z-test for Negative and Non-negative OCRs on Providing Defect Information

Since negative OCRs have a higher probability of having defective product information, a classifier built to distinguish negative OCRs from non-negative OCRs is useful for giving engineers insight on the evidence of a product defect. This RNN model contains five layers, as shown in Fig 3-9, which is similar to the one presented in section 2.5.1.



Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 300, 128)	1280000
dropout (Dropout)	(None, 300, 128)	0
lstm (LSTM)	(None, 70)	55720
dense (Dense)	(None, 1)	71

Total params: 1,335,791
Trainable params: 1,335,791
Non-trainable params: 0

Figure 3-9. RNN Model Architecture and Params for Negative and Non-negative OCRs Classification

The first layer is the input layer that takes input vectors. The second layer is for word embedding, as was described in section 3.3.2. The third layer is a dropout layer. The dropout layer randomly sets the neuron's output to 0 during each iteration of training to avoid overfitting. The fourth layer is an LSTM layer with 128 LSTM neurons chained in sequence for recurrently processing information during the training step. This study uses LSTM units with Keras neural-network library implementation built according to Hochreiter's paper "Long Short-Term Memory layer", with the architecture shown in Figure 3-10 (Hochreiter, 1997).

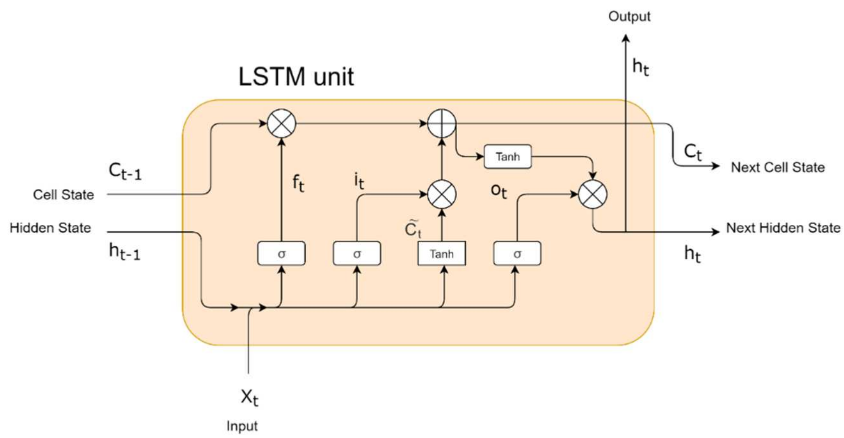


Figure 3-10. LSTM Unit Architecture

The LSTM unit takes the text representation vector X_t, as well as C_{t-1} previous cell memory information and h_{t-1} previous LSTM unit output information, as input to the unit. The first function of the unit is the forget gate function f_t, which decides if the input X_t and h_{t-1} should be kept or not using a sigmoid function (Equation 2-1) with weight W_f and bias b_f. In a textual data model, this function can forget word-level information, such as forgetting the gender of a word that applies earlier in the sentence but is no longer in use, so that the current pronouns can be updated later in the cell state. The function is written as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f)$$

Equation 3-1. Equation of Forget Gate

After this, the input information flow to the i_t sigmoid function and the $\tilde{C}_t$ Tanh function to create the candidate of information to be stored in the memory cell state for later use. This memory cell state is useful for making a decision later in the sequence. In a textual data model, this cell state can store information, such as gender of the present word, and can be used to predict gendered pronouns in future occurrences. The $\tilde{C}_t$ Tanh function, with an output range of [-1, 1], can decide the next candidate of the cell state. The i_t sigmoid function, with an output range of [0, 1], can decide if the input candidate is important to the unit. The $\tilde{C}_t$ Tanh function and i_t sigmoid function both have a weight $W_{\tilde{c}}$/ W_i and a bias $b_{\tilde{c}}$/ b_i to control the influence of the output. The $C_t$function combines both functions above to decide what to update and what to forget for the memory cell state. The functions are written as the following:

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i)$$

Equation 3-2. Equation of Input Gate

$$\tilde{C}_t = \tanh(W_{\tilde{c}} \cdot [h_{t-1}, X_t] + b_{\tilde{c}})$$

Equation 3-3. Equation of Candidate

$$C_t = f_t \cdot C_{t-1} + \tilde{C}_t \cdot i_t$$

Equation 3-4. Equation of Cell State

The final step is to output the information both as output prediction and move to the next hidden state. The input information would first pass to output Sigmoid function o_t with its weight W_o and bias b_o for determining what the next hidden state should be. This output function is multiplied with the tanh of the current cell state function C_t for output to determine the next hidden state of the LSTM unit as follows:

$$o_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o)$$

Equation 3-5. Equation of Output Gate

$$h_t = o_t \cdot \tanh(C_t)$$

Equation 3-6. Equation of Hidden State

After the information is processed with the chained 128 LSTM units, information is then sent to the output layer for classification. The output layer uses one Sigmoid to separate the output into two classes, either near 1 or near 0. This identifies the OCRs as either negative OCRs or non-negative OCRs. This model uses the Adaptive Moment Estimation (Adam) optimizer for backpropagation to update the hidden LSTM layer with a learning rate of 0.01. The model is then trained with 8100 OCRs, which is 90% of the total selected data, with one-third non-negative OCRs and two-thirds negative OCRs, with 50 epochs and a batch size of 32 OCRs in each batch. A successfully trained RNN model is able to automatically identify negative and non-negative OCRs with no given label. This allows engineers to identify negative OCRs from online forums or markets that only contain text reviews but no ratings.

To test the accuracy of the RNN model on classifying negative OCRs from non-negative OCRs, also known as hypothesis one, a set of 900 OCRs are separated out for testing, which is 10% of the total selected data, with one-third non-negative OCRs and two-thirds negative OCRs. These data have not been seen and trained by the RNN model. The RNN would then predict the labels of the testing data and compare it against the actual label. The output of the test would be a set of testing parameters, including precision, recall, f1-score, along with the ROC curve. This result is also tested with a one-tailed z-test for hypothesis testing.

3.4.2 Model Two: RNN classifier for classifying reviews that mention product defects from reviews that do not mention product defects

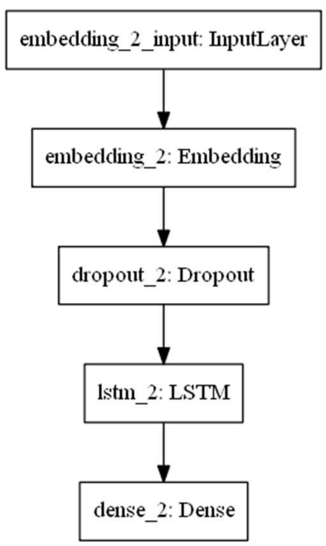
To further investigate which negative OCRs contain product defect or issue information, a second RNN classifier is built to identify OCRs that provide defective product information from OCRs that do not. Some negative OCRs do not contain defective product information. For example, review id # R2F8RCR0LFI7SS states: “We bought four of these, they are just some real cheap chairs that are overpriced.” This OCR only mentions the product is overpriced with no other information about the product issue. The second example is review id # R3W0KKHC5LK7K5 with the review title as “One Star," and a two-word review in the text body as "Pure junk." These do not give engineers any information about the product itself. The RNN model is trained with the two labels: "Defect information provided" and No defect information provided”. The OCRs that are manually labeled as "Manufacturing defect," "Problematic design or quality," or "Bad customer service" are considered with defective product information provided. Otherwise, OCRs with human labels of “No defect information” are considered as no defect information provided. This is presented in the following table:

Labels	Definition of Labels	Number of OCRs
Defect information provided	OCRs with labels of one of the following defect type categories: “Manufacturing defect," "Problematic design or quality," "Bad customer service."	6293

No defect information provided	OCRs with labels of "No defect information."	2707

Table 3-5. RNN Model Labels for Classifying OCRs that Mention Product Defects vs. Do Not Mention Product Defects

This model was trained with 8100 OCRs, which is 90% of total OCRs, with 50 epochs and a batch size of 32 OCRs in each batch. The architecture of this RNN is the same as the one in model one (Figure 3-9), displayed as the following:




```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 300, 128)          1280000
_________________________________________________________________
dropout_2 (Dropout)          (None, 300, 128)          0
_________________________________________________________________
lstm_2 (LSTM)                (None, 70)                55720
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 71
=================================================================
Total params: 1,335,791
Trainable params: 1,335,791
Non-trainable params: 0
_________________________________________________________________
```

Figure 3-11. RNN Model Architecture and Params for Classifying OCRs that Mention Product Defects vs. Do Not Mention Product Defects

A successfully-trained RNN model is able to automatically identify OCRs that provide defect information from OCRs that provide no defect information without a given label. This allows engineers to identify and filter out OCRs that are of value for their engineering design and product correction.

To test the accuracy of the RNN model on classifying OCRs that mention product defects from OCRs that do not mention product defects, also known as hypothesis two, a set of 900 negative OCRs, which is 10% of total, was used for testing. The data was not seen and trained by the RNN model. The RNN would then predict the labels of the testing data and compare it against the actual label. The output of the test was a set of testing parameters, including precision, recall, f1-score, along with the ROC curve. This result was also tested with a one-tailed z-test for hypothesis testing.

3.4.3 Model Three: LDA topic model for providing product defect insight

The Latent Dirichlet Allocation (LDA) topic model is able to automatically offer engineers a view on groups of words, items, or topics, that best describe OCRs with defect information. The LDA topic model builds a probabilistic text model by viewing a document, or OCR, as a mixture of topics, each with its own distribution of topics (Russell et al., 2011). Zhang et al. have suggested that by using the words from LDA discovered topics, they were able to discover the keywords of a defect issue and resolution among a group of complaints from the automobile industry (Zhang et al., 2019). As an example, Zhang et al. suggested that in the component's topic, it discovered the keywords of "ignition," "cylinder," "key," and "lock," which suggested the topic among these complaints is about an ignition component problem related to the car key in the lock. This allows engineers to have an early view of where the problem is.

The LDA topic model assumes that OCRs are represented as random mixtures over k latent topics, where each topic is characterized by a distribution over words (Blei et al., 2003). The model first randomly distributes k latent topics across OCRs by assigning each word a topic denoted as α. This random assignment of topics to words creates α (the

Dirichlet prior on the per-OCR topic distributions θ on M documents, or OCRs) and β (the Dirichlet prior on the per-topic word distribution φ on K topics). The θ and φ are based on Dirichlet prior with the following:

$$\theta_i \sim \mathrm{Dir}(\alpha), \text{where } i \ \in \{1, \ldots, M\}$$

Equation 3-7. Equation of OCRs - Topic Distributions θ

$$\varphi_i \sim \mathrm{Dir}(\beta), \text{where } k \ \in \{1, \ldots, K\}$$

Equation 3-8. Equation of Topics- Word Distribution φ

To improve the word distributions on topics, LDA iteratively goes through each word $w_{i,j}$ in each OCR (assuming word $w_{i,j}$ to is incorrectly assigned but the others are correctly assigned) and probabilistically reassigns word $w_{i,j}$ to topics $\boldsymbol{z}$. This reassignment is based on: 1) θ_i the proportion of words in OCR m that are currently assigned to topic $\boldsymbol{z_{i,j}}$, and 2) φ_k the proportion of this current word $w_{i,j}$ that is assigned to topic $\boldsymbol{z}$ over all OCRs. Given the parameter α and β, the joint distribution of a topic mixture θ, a set of *N* topics **z**, and a set of *N* words **w** is given by Blei et al. (2003).

$$p(\theta, \boldsymbol{z}, \boldsymbol{w} |\ \alpha, \beta) = p(\theta|\alpha) \prod_{k=1}^{N} p(z_n|\theta)\ p(w_n|z_n, \beta)$$

Equation 3-9. Equation of LDA Join Distribution

Figure 3-14 shows the plate notation of the LDA model. This shows the Bayesian network relationship of how OCRs and words are topically related.

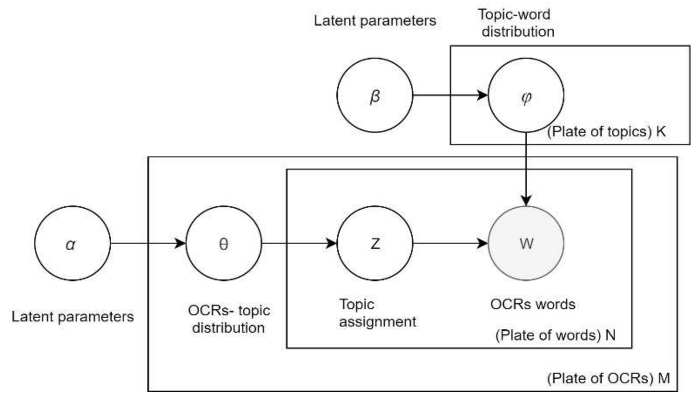


Figure 3-12. Plate Notation of LDA Model

The following variables are used in the LDA model:

- θ_i is the topic distribution for OCRs i
- α is the parameter of the Dirichlet prior distribution on each OCRs - topic
- M is the number of OCRs
- φ_k is the word distribution for topic k
- β is the parameter of the Dirichlet prior distribution on each topic-word
- K is the number of Latent topics
- N is the number of words in a given OCRs
- $w_{i,j}$ is the current word in the j^{th} word in the i^{th} OCRs
- $\boldsymbol{z_{i,j}}$ is the current topic for $w_{i,j}$

After a large number of the iterative training process of reassigning words to topics according to the multinomial distribution, the model is converged with the K numbers of topics each with word distribution φ_k that best describe the set of OCRs. The words that makeup the word distribution φ_k is able to tell the information about the topics among the group of selected OCRs. This allows engineers to have an early view on the OCRs that provided defective information, before manually looking at each one of them.

While LDA models are able to give a view of a list of topics that a group of OCRs is mentioning, for its unsupervised learning property, there is no actual label for the model to test against. The most common evaluation metric that is used on LDA models is topic coherence. Topic coherence measures semantic similarity among the top words that appear in the word distribution φ_k for a single topic. This method might not include the actual meaning of the words that tie back to the defective information. Chang et al. suggested these traditional topic coherence metrics are negatively correlated with the measures of topic quality developed, and they agree that human judgments and manual determinations remain a better way to determine if the LDA model is giving out informative topics among a set of documents or OCRs (Change et al., 2009).

To test out this LDA approach along with RNN models that filter out OCRs that mention defective product information, a set of OCRs of the 8 top-selling home and furniture products on Amazon were selected for testing. This process developed 8 test cases. The OCRs from each product were first filtered using the first RNN model (Figure 3-10) for selecting negative OCRs and then filtered using the second RNN model (Figure 3-11) for selecting OCRs that mention defective product information. These OCRs were then inputted into LDA for discovering essential topic information on defective product information. The 10 highest-scoring words that build upon the 5 LDA topics were used as testing purposes. Amazon SageMaker labelers were then asked to identify if these topics, or the group of words, are related back to the defectiveness of a product or the details of a product itself. This method verifies whether the output topic words are linked back to the product or the defect itself.

3.5 Summary

This chapter reviewed the methods of collecting, labeling, and preprocessing the raw data, as well as giving the analysis methods used for hypothesis testing. It reviewed order a set of three probability models that were developed to help engineers filter out OCRs with valuable defective information as well as to aggregate this information into a set of topics. These methods allow the engineers to have a fast and automatically-generated view on these OCRs related to defect information without manually reading them one by one.

Chapter 4—Results

4.1 Introduction

This chapter provides comprehensive results of the three presented models conducted during this study with the goal of giving engineers an early view to product defects and issues. The first two models are recurrent neural network (RNN) classifiers that categorize OCRs that contain product defect information. This section also demonstrates the results of these models against the human labels of both the model training and the model testing stages. The last model is the Latent Dirichlet Allocation (LDA) topic model that can extract product defect information from OCRs. A total of eight test cases for the LDA model using the eight bestselling products on the Amazon.com home furniture section were used. This section presents the output of LDA model along with their relevance to the defective product information.

4.2 Model One: RNN classifier for classifying negative reviews from non-negative reviews

The first model is the RNN classifier for sorting negative OCRs from non-negative OCRs. Negative OCRs are defined as those OCRs with 1 or 2-star customer ratings. Non-negative OCRs are those defined as OCRs with 3 or more stars customer ratings. The model was trained with 8100 OCRs and was validated with 900 OCRs, with one-third non-negative OCRs and two-thirds negative OCRs. The 8100 OCRs were the training dataset reserved for training the RNN model. The 900 OCRs, which had never been trained by the model, were the testing dataset. The model was executed for 50 epochs, with a batch size of 32 OCRs in each batch, at a learning rate of 0.01. The training results are displayed in the following paragraph.

Figure 4-1 is the training and testing accuracy and loss over the 50 epochs. Loss, or cost, function is the difference of predicted output values with respect to the representation of the provided correct labels to measure how well the model is performing, as stated in Section 2.5.2. Table 4-1 is the training and testing output metrics per 10 epochs over the 50 epochs. The model converged at about epoch 20 with a testing accuracy at about 85%.

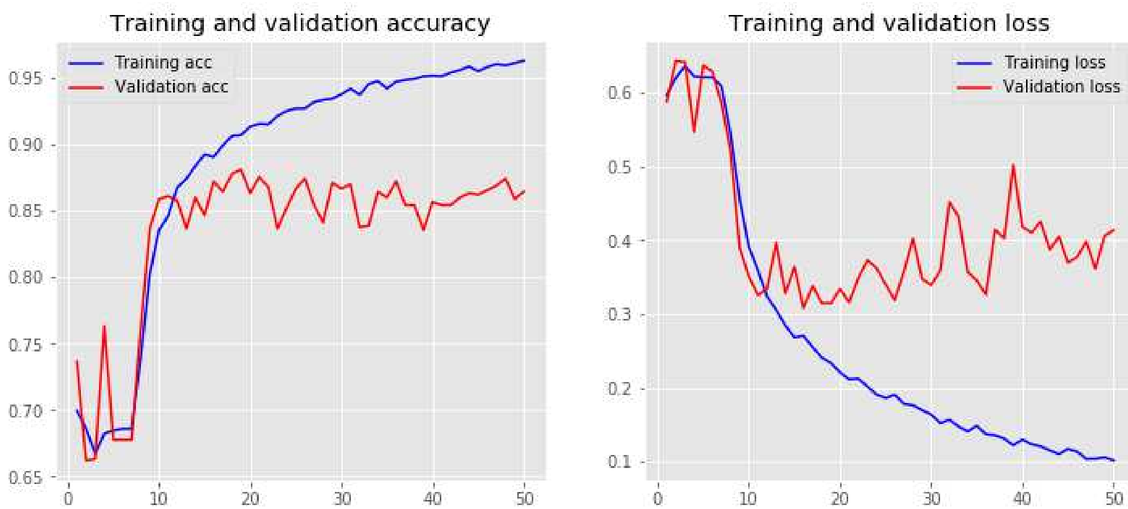


Figure 4-1. Training and Testing (validation) Accuracy and Loss for RNN Model Classifier for Classifying Negative Reviews from Non-negative OCRs

Epochs	Training Loss	Training Accuracy	Testing Loss	Testing Accuracy
1	0.5928	0.7012	0.7229	0.4988
10	0.3169	0.8681	0.3744	0.8278
20	0.1955	0.9253	0.3130	0.8589
30	0.1358	0.9468	0.3416	0.8778
40	0.1187	0.9558	0.3763	0.8744

50	0.0979	0.9648	0.4508	0.8500

Table 4-1. Training Output Metrics of RNN Model for Classifying Negative OCRs from Non-negative OCRs

4.2.1 Hypothesis 1

H1: A predictive model can identify OCRs containing negative sentiments with 70% accuracy.

Hypothesis 1 evaluates the performance of the RNN Model for classifying negative reviews from non-negative reviews. The test data that was not trained by the model was predicted after the 50 epochs compared with the output against the actual label. Table 4-2 shows the prediction statistics and hypothesis validity of this model. This table shows the hypothesis test using one sample Z-tests for a proportion at alpha 0.05 with 70% accuracy. Figure 4-2 shows the receiver operating characteristic (ROC) curve of this model.

Prediction Statistics			
True Positives	509	Testing Accuracy	0.8500
False Positives	48	Testing Precision	0.9156
True Negative	256	Testing F1 Score	0.8851
False Negative	87	Testing Recall	0.8593
One sample Z-tests for a proportion			
$H_0 : p = 0.7$ $H_1 : p > 0.7$			

Sample proportion	0.850000	95% confidence interval for proportion	(0.826672, 0.873328)
Z-value	9,82	P-value	0.000

Table 4-2. Prediction Statistics and Hypothesis Test of for RNN Model Classifier for Classifying Negative OCRs from Non-negative OCRs

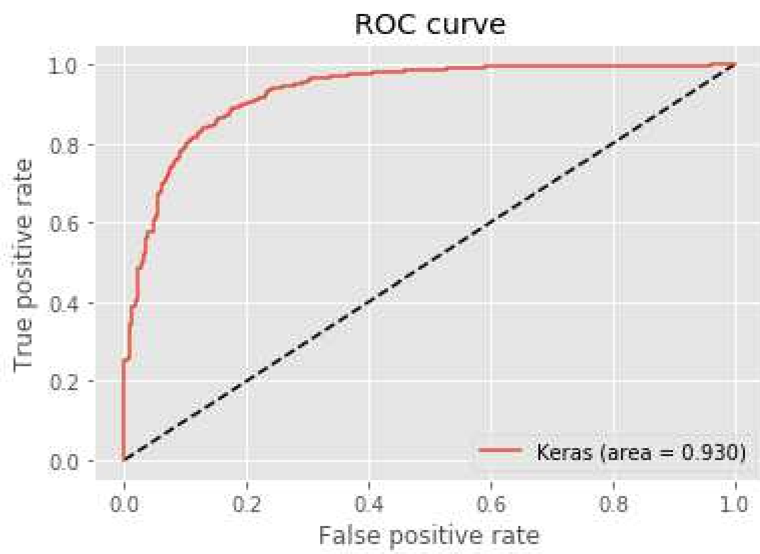


Figure 4-2. ROC Curve for RNN Model for Classifying Negative OCRs from Non-negative OCRs

A significance level, alpha, of 0.05 was used throughout this praxis. Table 4-2 shows that the model correctly predicted 765 OCRs out of 900 OCRs. Hypothesis 1 was tested with a one-sample Z-test for a proportion to evaluate if the model has an accuracy of 70% accuracy. With the area under a ROC Curve at 0.930, the model has good predictive power. With the P-value at 0, the hypothesis test successfully rejected the null hypothesis, and thus this RNN classifier is sufficient in classifying negative reviews from non-negative reviews.

4.3 Model Two: RNN classifier for classifying reviews that mention product defects from reviews that do not mention product defects

The second model is the RNN classifier for categorizing OCRs that mention product defects from OCRs that do not mention product defects. OCRs that mention product defects are defined as the OCRs that are manually labeled as "Manufacturing defect," "Problematic design or quality," or "Bad customer service." OCRs that do not mention product defects are defined as the OCRs that are manually labeled as “No defect information.” This model was trained with 8100 OCRs and was validated with 900 OCRs. The training data is the data used for training and fitting to the RNN model. The testing data is the data used for testing the model, which was not adjusted by the model. This model executed for 50 epochs, with a batch size of 32 OCRs in each batch, at a learning rate of 0.01. The training result is displayed in the following paragraph.

Figure 4-3 is the training and testing accuracy and loss over the 50 epochs. Table 4-3 is the training and testing output metrics per 5 epochs over the 50 epochs. The model converged at about epoch 20 with a testing accuracy at about 84%.

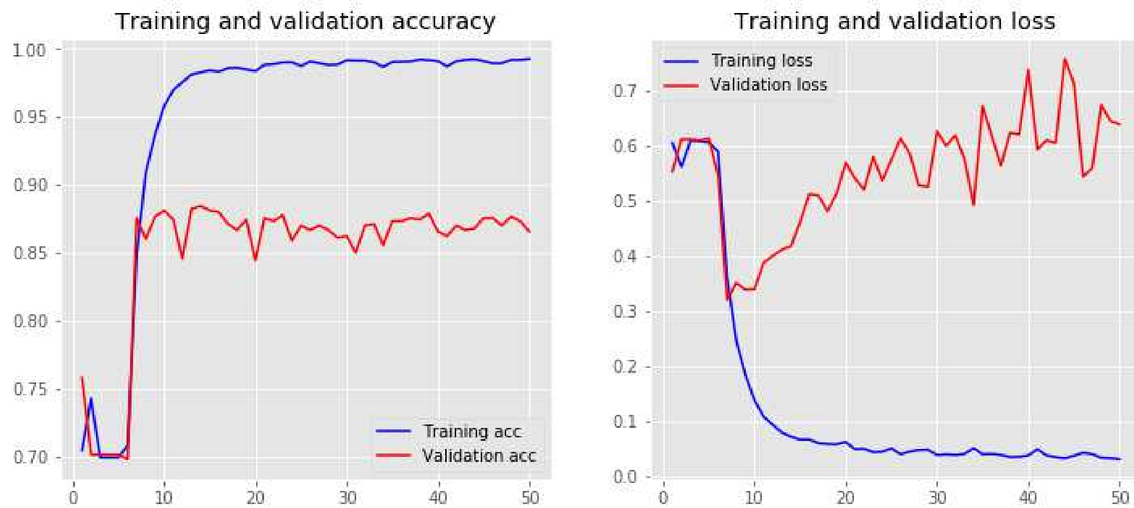


Figure 4-3. Training and Testing (validation) Accuracy and Loss for RNN Model for Classifying OCRs That Mention Product Defects from OCRs that Do Not Mention Product Defects

Epochs	Training Loss	Training Accuracy	Testing Loss	Testing Accuracy
1	0.6049	0.7043	0.5533	0.7578
10	0.1373	0.9680	0.3396	0.8811
20	0.0616	0.9838	0.5697	0.8444
30	0.0378	0.9919	0.6263	0.8622
40	0.0370	0.9912	0.7384	0.8656
50	0.0301	0.9928	0.6391	0.8656

Table 4-3. Training Output Metrics of RNN Model for Classifying OCRs that Mention Product Defects from OCRs that Do Not Mention Product Defects

4.3.1 Hypothesis 2

H2: A predictive model can identify OCRs containing defect information with 70% accuracy.

Hypothesis 2 evaluates the performance of the RNN classifier for classifying OCRs that mention product defects from OCRs that do not mention product defects. The test data that was not fitted by the model was predicted after the 40 epochs and results were compared against the actual label. Table 4-4 shows the prediction metrics and hypothesis test of this model. This table shows the hypothesis test of the one sample Z-tests for a proportion at alpha 0.05 with 70% accuracy. Figure 4-4 shows the ROC curve of this model.

Prediction Statistics			
True Positives	579	Testing Accuracy	0.8659
False Positives	69	Testing Precision	0.8985
True Negative	200	Testing F1 Score	0.9208
False Negative	52	Testing Recall	0.9208
One sample Z-tests for a proportion			
$H_0 : p = 0.7$ $H_1 : p > 0.7$			
Sample proportion	0.865556	95% confidence interval for proportion	(0.843269, 0.887842)

Z-value	10.84	P-value	0.000

Table 4-4. Prediction Metrics and Hypothesis Test of RNN Model for Dlassifying OCRs that Mention Product Defects from OCRs that Do Dot Mention Product Defects

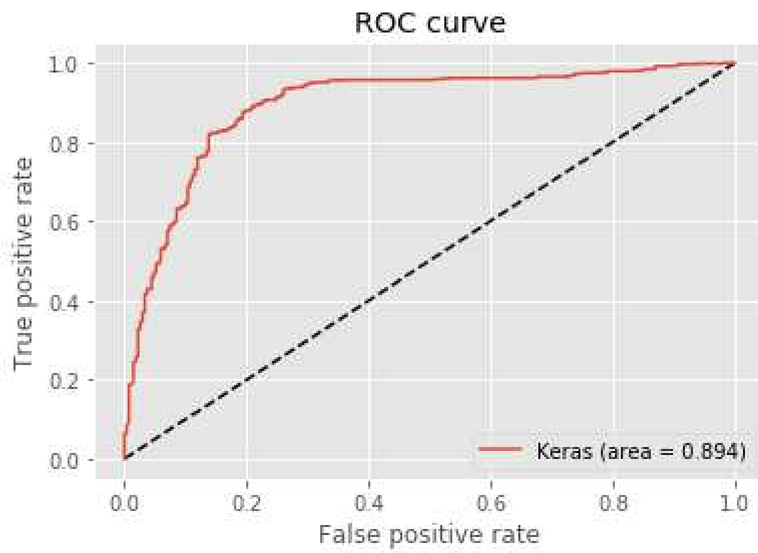


Figure 4-4. ROC Curve for RNN Model for Classifying OCRs that Mention Product Defects from OCRs that Do Not Mention Product Defects

A significance level, alpha, of 0.05 is used overall this praxis. Table 4-4 shows that the model correctly predicted 779 OCRs out of 900 OCRs. The predictive power of this model is good with the area under a ROC Curve at 0.894. Hypothesis 2 was tested with one-sample Z-tests for a proportion to evaluate if the model has an accuracy of 70%. With the P-value at 0, the hypothesis test successfully rejected the null hypothesis, and thus this RNN classifier is sufficient in classifying reviews that mention product defects from reviews that do not mention product defects.

4.4 Model Three: LDA topic model for providing product defect insight

The LDA topic model was used in this study to identify groups of words, or topics, that best describe OCRs that have already been identified as negative and embedded with defect information by previous models. The selection provides the aggregated LDA topic modeling result of the top eight furniture products in Amazon.com as of 12th April 2020. Each product was provided with 5 topics with 10 words each. Those groups of words were validated by Amazon SageMaker human labelers to verify their relevance to the product. Amazon SageMaker human labelers were asked to evaluate if at least half of the group of words retrieved by the LDA topics are relevant to detail, buying process, usage, or defect of a furniture product.

4.4.1 Test Case One: Linenspa 2 Inch Gel Infused Memory Foam Mattress Topper, Twin

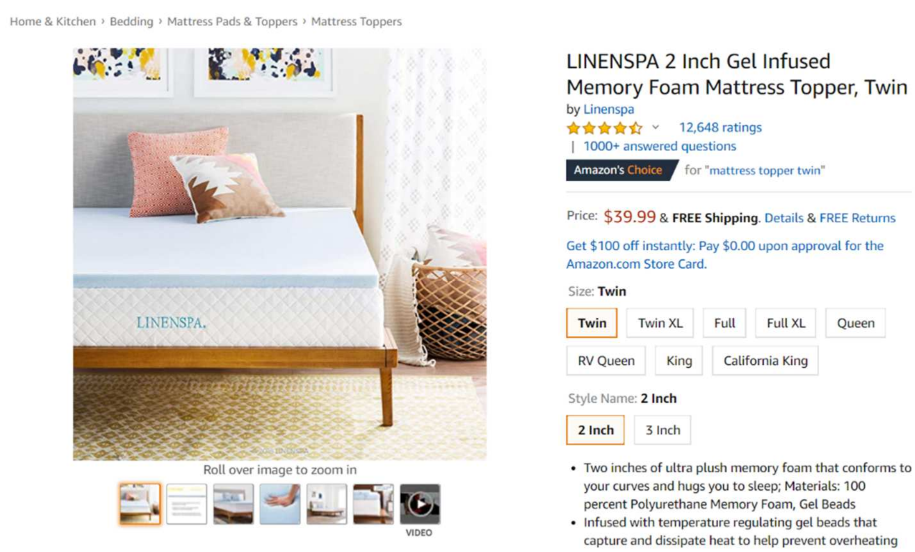


Figure 4-5. Linenspa Gel Infused Memory Foam Mattress on Amazon.com

Linenspa 2 Inch Gel Infused Memory Foam Mattress Topper (Figure 4-5) is a polyurethane memory foam mattress topper that adds softness on top of mattresses to enhance the sleeping experience. This product received 12,684-star ratings at an average of 4.7 out of 5 stars and 7202 valid OCRs. The RNN classifier for classifying negative OCRs from non-negative OCRs identified 1658 negative OCRs among the total valid OCRs. The second RNN classifier for classifying reviews that mention product defects from reviews that do not mention product defects identified 1363 OCRs that mention product defects.

The LDA topic model was used to identify the 5 topics with 10 words each that best describe the OCRs which mention product defects and generated a topic Coherence Value (CV) of 0.4930. The following table shows the topics, their supporting words, and their support weight. The topics were also given to Amazon SageMaker human labelers for their relevance to a furniture product or a product defect, along with its corresponding confidence level.

Topic	Words supporting the topic (support weight)						Is this topic relevant to the product or a defect? (Confidence)
	Words	Weight	Words	Weight	Words	Weight	Yes (0.95)
1	sleep	0.029	pain	0.028	bed	0.027	
	pad	0.027	night	0.021	use	0.016	
	topper	0.014	back_ pain	0.014	hip	0.013	

	help	0.013					
2	topper	0.035	sleep	0.027	bed	0.023	Yes (0.95)
	night	0.017	get	0.016	Cool	0.016	
	hot	0.014	purchase	0.013	bought	0.012	
	memory _foam	0.012					
3	bed	0.035	topper	0.026	smell	0.023	Yes (0.94)
	hour	0.021	like	0.018	inch	0.017	
	size	0.016	open	0.016	air	0.016	
	order	0.015					
4	back	0.036	return	0.027	box	0.022	Yes (0.95)
	topper	0.020	would	0.019	one	0.019	
	try	0.015	review	0.011	amazon	0.010	
	disappoi nt	0.010					
5	soft	0.036	topper	0.026	bed	0.022	Yes (0.95)
	sink	0.020	support	0.020	like	0.019	
	foam	0.018	firm	0.014	really	0.014	
	body	0.014					

Table 4-5. Topic Model Words for LINENSPA Mattress Topper

Amazon SageMaker human judgment indicated 5 out of 5 topics show words related to furniture products or product issues. The 5 topics summarized the terms used in OCRs that mention defects. Words that support topic number 5 might indicate a body support problem with the topper being too soft and the foam sinks. The following list shows three sampled OCRs that indicate body support problems using the word search function with the word “support” in the data.

- “Six months later the foam is squashed and offers no support at all. She weighs less than 110 lbs. I will not buy this pad again.” (review ID # R2MV2APVP8P9CX)
- “this topper goes completely flat once you lay on it and offers zero support.” (review ID # RYTH07FT8W7XS)
- “This thing provided no support, it crushes down to nothing anywhere there is weight (I'm 5'9" 155lbs... Shouldn't be an issue)” (review ID # RRPFIQ56HYMVU)

4.4.2 Test Case Two: Homall Gaming Chair

Figure 4-6. Homall Gaming Chair on Amazon.com

Homall Gaming Chair (Figure 4-6) is a gaming chair with seat height adjustment, 360 Degree Swivel, and Multi-direction Wheels. This product received 11,027-star ratings at an average of 4.3 out of 5 stars and 6,381 valid OCRs. The RNN classifier for classifying negative OCRs from non-negative OCRs identified 1,732 negative OCRs among the total valid OCRs. And the second RNN classifier for classifying reviews that mention product defects from reviews that do not mention product defects identified 1,559 OCRs that mention product defects.

The LDA topic model was used to identify the 5 topics with 10 words each that best describe OCRs which mention product defects and generated a topic CV of 0.3583. The following table shows the topics, their supporting words, and their support weight. The topics are also given to Amazon SageMaker human labelers for their relevance to a furniture product or a product defect, along with its corresponding confidence level.

Topic	Words supporting the topic (support weight)						Is this topic relevant to the product or a defect? (Confidence)
	Words	Weight	Words	Weight	Words	Weight	Yes (0.95)
1	seat	0.028	start	0.020	month	0.028	
	arm_rest	0.016	arm	0.013	use	0.012	
	rock	0.011	leather	0.010	bolt	0.010	
	uncomfortable	0.010					
2	screw	0.035	assemble	0.015	hole	0.018	Yes (0.95)
	instruction	0.017	one	0.016	get	0.013	
	part	0.011	bolt	0.011	pieces	0.011	
	came	0.010					
3	sit	0.024	back	0.021	seat	0.019	Yes (0.95)
	use	0.015	cushion	0.013	really	0.013	
	uncomfortable	0.012	game	0.012	adjust	0.010	

	feel	0.009					
4	like	0.027	buy	0.021	would	0.017	Yes (0.95)
	uncomfortable	0.017	seat	0.016	get	0.016	
	back	0.016	cheap	0.015	sit	0.014	
	pad	0.013					
5	month	0.026	one	0.023	broke	0.019	No (0.95)
	review	0.015	use	0.015	broken	0.013	
	bought	0.013	replace	0.011	back	0.010	
	refund	0.010					

Table 4-6. Topic Model Words for Homall Gaming Chair

Amazon SageMaker human judgment indicated 4 out of 5 topics show words related to furniture products or product issues. The 5 topics summarized the terms used in OCRs that mention defects. Words supporting topic number 2 might indicate an assembly problem, with words such as "screw," "bolt," and "hole" mentioned. The following list shows three sampled OCRs that indicate assembly problems using the word search function with the word "screw" in the data.

- "One of the pre drilled screw ports was misthreaded and cause the screw to strip on one of the arm rests." (review ID # R2DB0Q8O15GIM9)
- "One of the side brackets on two of the chairs has a screw hole with no threads, which means no bracket cover." (review ID # R1FXZTBM0XKDU)

- “I have tried to disassemble and reassemble numerous times, I can only imagine that the holes drilled for the screws is uneven and causes the chair to screw in unevenly.” (review ID # RZH9BO806AFMY)

4.4.3 Test Case Three: Furinno 3-Tier Open Shelf Bookcase

Figure 4-7. Furinno 3-Tier Open Shelf Bookcase on Amazon.com

Furinno 3-Tier Open Shelf Bookcase (Figure 4-7) is a storage cabinet manufactured with engineered Particle Board. This product received 4,006-star ratings at an average of 4.1 out of 5 stars and 1,917 valid OCRs. The RNN classifier for classifying negative OCRs from non-negative OCRs identified 682 negative OCRs among the total valid OCRs. The second RNN classifier for classifying reviews that mention product defects from reviews that do not mention product defects identified 619 OCRs that mention product defects.

The LDA topic model was used to identify the 5 topics with 10 words each that best described OCRs which mention product defects and generated a topic CV of 0.5162. The following table shows the topics, their supporting words, and their support weight. The topics were also given to Amazon SageMaker human labelers for their relevance to a furniture product or a product defect, along with its corresponding confidence level.

Topic	Words supporting the topic (supporting weight)						Is this topic relevant to the product or a defect? (Confidence)
	Words	Weight	Words	Weight	Words	Weight	Yes (0.95)
1	put_together	0.034	cheap	0.030	quality	0.027	
	get	0.021	price	0.018	assemble	0.016	
	disappoint	0.016	return	0.015	screw	0.015	
	miss	0.014					
2	came	0.045	damage	0.042	box	0.028	Yes (0.95)
	receive	0.027	item	0.022	return	0.022	

	broken	0.020	pieces	0.019	shelf	0.019	
	chip	0.019					
3	small	0.068	book	0.033	cube	0.033	Yes (0.94)
	fit	0.031	shelf	0.031	size	0.020	
	space	0.016	look	0.015	storage	0.014	
	return	0.014					
4	screw	0.053	top	0.043	pieces	0.041	Yes (0.95)
	hold	0.039	bottom	0.030	part	0.025	
	dowel	0.022	drill	0.021	put_together	0.017	
	broken	0.017					
5	screw	0.047	side	0.038	sticker	0.032	Yes (0.95)
	cover	0.025	back	0.024	shelf	0.024	
	look	0.018	panel	0.018	unit	0.016	
	assemble	0.016					

Table 4-7. Topic Model Words for Furinno Open Shelf Bookcase

Amazon SageMaker human judgment indicated 5 out of 5 topics had words related to furniture products or product issues. The 5 topics summarized the terms used in OCRs that mention defects. Words that support topic number 2 might indicate product may ship with damaged pieces, with words such as “damage,” “box,” and “came” mentioned. The following list shows three sampled OCRs that indicate assembly problems using the word search function with the word "damage" in the data.

- "The packaging needs to be sturdier. The box had bumps on edges and the edges of the plyboard inside were damaged when it arrived." (review ID # R3FOMSDTKNBUXR)
- "The corners were bent and damaged on almost all the pieces as well, so it's not even in good shape for the price." (review ID # RNS89OL7MK10D)
- "The corners to the box were not protected, and it obviously got dropped on one corner damaging all pieces at that corner. See photos." (review ID # RH8QQIM1RWATX)

4.4.4 Test Case Four: Furinno Simplistic Study Table

Figure 4-8. Furinno Simplistic Study Table on Amazon.com

Furinno Simplistic Study Table (Figure 4-8) is a desk manufactured from high quality durable composite wood. This product received 2,137-star ratings at an average of 3.8 out of 5 stars and 1,291 valid OCRs. The RNN classifier for classifying negative OCRs from non-negative OCRs identified 420 negative OCRs among the total valid OCRs. The second RNN classifier for classifying reviews that mention product defects from reviews that do not mention product defects identified 386 OCRs that mention product defects.

The LDA topic model was used to identify the 5 topics with 10 words each that best describe OCRs which mention product defects and generated a topic CV of 0.4256. The following table shows the topics, their supporting words, and their support weight. The topics were provided to Amazon SageMaker human labelers for their relevance to a furniture product or a product defect, along with its corresponding confidence level.

Topic	Words supporting the topic (supporting weight)						Is this topic relevant to the product or a defect? (Confidence)
	Words	Weight	Words	Weight	Words	Weight	Yes (0.95)
1	screw	0.092	table	0.061	one	0.046	

	top	0.043	hole	0.040	came	0.034	
	pieces	0.033	side	0.032	put_ together	0.031	
	board	0.025					
2	cheap	0.099	quality	0.063	assemble	0.038	Yes (0.95)
	time	0.038	poor	0.033	material	0.031	
	wast_ money	0.028	easy	0.027	flimsy	0.024	
	apart	0.021					
3	desk	0.058	flimsy	0.052	bad	0.046	Yes (0.95)
	damage	0.045	worth	0.038	wobble	0.038	
	cheap	0.029	table	0.028	quality	0.028	
	terrible	0.027					
4	sturdy	0.074	need	0.044	desk	0.042	Yes (0.95)
	table	0.039	expect	0.033	put	0.033	
	much	0.032	buy	0.028	something	0.022	
	better	0.021					
5	table	0.044	return	0.035	assemble	0.033	Yes (0.95)
	miss	0.033	desk	0.031	product	0.030	
	damage	0.026	scratch	0.022	arrive	0.022	

	edge	0.018					

Table 4-8. Topic Model Words for FURINNO Study Table

Amazon SageMaker human judgment indicated 5 out of 5 topics show words related to furniture products or product issues. The 5 topics summarized the terms used in OCRs that mention defects. Words that support topic number 2 and 3 might indicate a quality issue, with words such as “flimsy,” “wobble,” and “quality” mentioned. The following list shows three sampled OCRs that indicate assembly problems using the word search function with the word “wobble” in the data.

- “Don't waste your money! It may be easy to put up, but it wobbles when you stand it up - so not safe as a desk.” (review ID # R1DRFJ3OF0OE2L)
- “This is very cheap. Extremely wobbly. The cam fits would not turn all the way to secure the top to the base.” (review ID # R21WWX5JK4ZEBB)
- “All we have is a laptop on it and just getting up and down it wobbles/could almost fall over. Would not buy again.” (review ID # R1ZF70XHTY4NWV)

4.4.5 Test Case Five: Ashley Furniture Signature Design 8 Inch Innerspring Queen Mattress

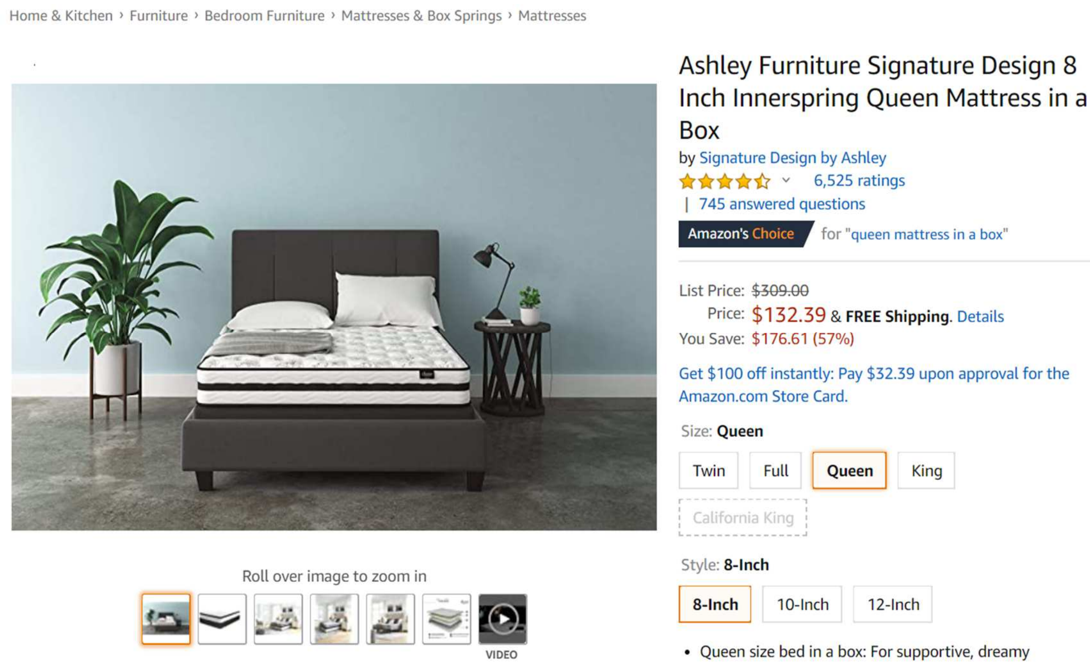


Figure 4-9. Ashley Furniture Signature Design 8 Inch Innerspring Queen Mattress on Amazon.com

Ashley Furniture Signature Design 8 Inch Innerspring Queen Mattress (Figure 4-9) is a mattress designed with layers of quilt foam, high-density padding, and high-quality 13-gauge Bonnell coil units. This product received 6,525-star ratings at an average of 4.3 out of 5 stars and 3,673 valid OCRs. The RNN classifier for classifying negative OCRs from non-negative OCRs identified 879 negative OCRs among the total valid OCRs. The second RNN classifier for classifying reviews that mention product defects from reviews that do not mention product defects identified 752 OCRs that mention product defects.

The LDA topic model was used to identify the 5 topics with 10 words each that best describe OCRs which mention product defects and generated a topic CV of 0.3959. The following table shows the topics, their supporting words, and their support weight. The topics were also given to Amazon SageMaker human labelers for their relevance to a furniture product or a product defect, along with its corresponding confidence level.

Topic	Words supporting the topic (supporting weight)						Is this topic relevant to the product or a defect? (Confidence)
	Words	Weight	Words	Weight	Words	Weight	Yes (0.95)
1	sleep	0.031	first	0.023	get	0.021	
	month	0.019	year	0.017	much	0.017	
	bought	0.016	memory _foam	0.014	put _complete	0.014	
	feel	0.013					
2	one	0.036	bed	0.029	firm	0.025	Yes (0.93)
	spring	0.023	bad	0.022	support	0.022	
	size	0.019	terrible	0.018	enough	0.016	

	coil	0.015					
3	firm	0.045	like	0.024	soft	0.022	Yes (0.94)
	would	0.020	box	0.019	bed	0.017	
	look	0.012	night	0.012	bought	0.011	
	purchase	0.010					
4	pain	0.046	sleep	0.045	back	0.039	Yes (0.95)
	month	0.037	very	0.025	wake	0.023	
	worst	0.017	hip	0.017	hurt	0.016	
	side	0.015					
5	bed	0.045	inch	0.028	inflate	0.017	Yes (0.95)
	comfort	0.015	smell	0.013	top	0.013	
	frame	0.012	spring	0.012	plush	0.011	
	plastic	0.011					

Table 4-9. Topic model words for Ashley Mattress

Amazon SageMaker human judgment indicated 5 out of 5 topics showed words related to furniture products or product issues. The 5 topics summarized the terms used in OCRs that mention defects. Words that support topic number 2 might indicate a product issue, with its spring and coils with words such as "firm," "spring," and "coil" mentioned. The following list shows three sampled OCRs that indicate assembly problems using the word search function with the word "coil" in the data.

- “There’s almost no padding on top of the coils, so I can feel every one of them. I purchased the 12-inch version and it only inflated to 10 inches.” (review ID # R10IYP9UXQR44P)
- “The coils move (and creak) with you. The only buffer is the thin leaf of fabric (alleged to be memory foam) between my back and this neatly arranged pile of rocks.” (review ID # R1EKZ80MX14WGF)
- “When I lay down I sink so far I can feel the coils and see them protruding from the sides of the bed. This is not a firm bed. I have never woken up with more pain and stiffness in my life.” (review ID # R2JU2VDDCOPOUF)

4.4.6 Test Case Six: Lifetime Folding Laptop Table TV Tray

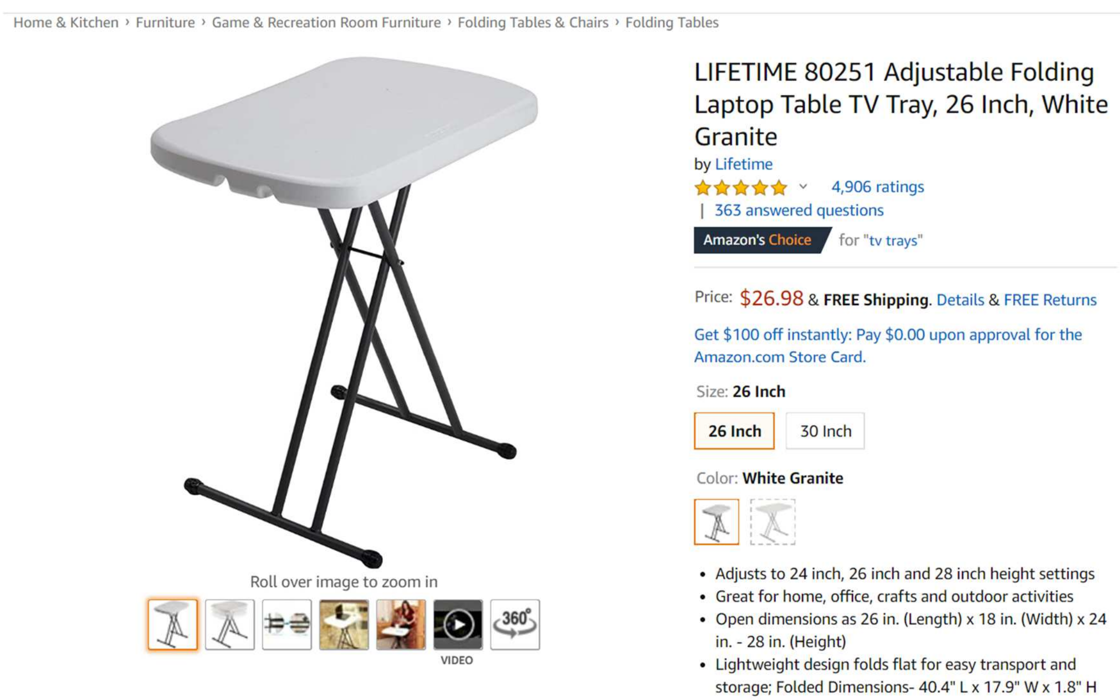


Figure 4-10. Lifetime Folding Laptop Table TV Tray on Amazon.com

Lifetime Folding Laptop Table TV Tray (Figure 4-10) is a small folding table constructed of high-density polyethylene. This product received 4,906-star ratings at an average of 4.8 out of 5 stars and 3,717 valid OCRs. The RNN classifier for classifying negative OCRs from non-negative OCRs identified 300 negative OCRs among the total valid OCRs. The second RNN classifier for classifying reviews that mention product defects from reviews that do not mention product defects identified 229 OCRs that mention product defects.

The LDA topic model was used to identify the 5 topics with 10 words each that best describe OCRs which mention product defects and generated a topic CV of 0.3503. The following table shows the topics, their supporting words, and their support weight. The topics were also given to Amazon SageMaker human labelers for their relevance to a furniture product or a product defect, along with its corresponding confidence level.

Topic	Words supporting the topic (supporting weight)						Is this topic relevant to the product or a defect? (Confidence)
	Words	Weight	Words	Weight	Words	Weight	Yes (0.95)
1	leg	0.063	much	0.043	open	0.030	
	better	0.026	close	0.025	problem	0.019	
	adjust	0.019	hard	0.019	use	0.018	
	try	0.016					

2	one	0.070	height	0.040	small	0.031	No (0.94)
	work	0.028	return	0.025	amazon	0.023	
	bought	0.021	would	0.020	say	0.020	
	got	0.019					
3	use	0.077	sturdy	0.039	laptop	0.037	No (0.95)
	fold	0.032	room	0.028	little	0.027	
	bought	0.025	surface	0.020	put	0.020	
	size	0.019					
4	plastic	0.059	set	0.054	like	0.041	Yes (0.94)
	leg	0.040	fold	0.035	probable	0.023	
	sturdy	0.022	squeeze	0.019	change	0.019	
	height	0.018					
5	leg	0.078	plastic	0.041	use	0.034	Yes (0.95)
	bar	0.034	hole	0.029	collapse	0.028	
	squeeze	0.028	place	0.025	adjust	0.024	
	fit	0.023					

Table 4-10. Topic Model Words for Lifetime Folding Laptop Table

Amazon SageMaker human judgment indicated 3 out of 5 topics shows words related to furniture products or product issues. The 5 topics summarized the terms used in OCRs that mention defects. Words that support topic number 5 might indicate the product has a design problem with the leg, with words such as "plastic," "leg," and "squeeze" mentioned. The following list shows three sampled OCRs that indicate assembly problems using the word search function with the word “pain” in the data.

- “You're supposed to squeeze this plastic trigger to partially-collapse one of the metal tube ends of the legs so that they fit into one of several holes to set the table's height. However, it appears the unit I got, which was a used version, has the tubes cut at the factory far too long, and as a result, you cannot collapse the tube far enough to insert it into, or remove it from, the holes.” (review ID # R3Q45LQKYAJKXN)
- “I don't recommend this table. It was challenging to set the legs in place. I thought I would need to donate the table to charity so that someone else would have better luck, but after many tries, I was able to get the legs in place” (review ID # R1ENEEJAQZDUCD)
- “The table is wably and uneven, it seems that one leg is shorter. The tubes (legs) are much smaller in diameter.” (review ID # R18GHHQAKW8XKQ)

4.4.7 Test Case Seven: BestOffice Office Chair

Figure 4-11. BestOffice Office Chair on Amazon.com

BestOffice Office Chair (Figure 4-11) is an armchair with adjustable stool and rolling swivel. This product received 5,975-star ratings at an average of 4.2 out of 5 stars and 3,508 valid OCRs. The RNN classifier for classifying negative OCRs from non-negative OCRs identified 787 negative OCRs among the total valid OCRs. The second RNN classifier for classifying reviews that mention product defects from reviews that do not mention product defects identified 689 OCRs that mention product defects.

The LDA topic model was used to identify the 5 topics with 10 words each that best describe OCRs which mention product defects and generated a topic CV of 0.3698. The following shows the topics, their supporting words, and their support. The topics were also given to Amazon SageMaker human labelers for their relevance to a furniture product or a product defect, along with its corresponding confidence level.

Topic	Words supporting the topic (supporting weight)						Is this topic relevant to the product or a defect? (Confidence)
	Words	Weight	Words	Weight	Words	Weight	No (0.90)
1	back	0.035	use	0.034	broke	0.020	
	return	0.018	month	0.018	day	0.018	
	seller	0.016	replace	0.016	purchase	0.016	
	plastic	0.015					
2	sit	0.049	seat	0.038	back	0.036	Yes (0.95)
	uncomfortable	0.033	price	0.019	make	0.018	
	hip	0.018	cushion	0.017	arm	0.017	
	leg	0.017					
3	assemble	0.025	back	0.022	wheel	0.018	Yes (0.90)
	one	0.019	star	0.015	part	0.014	

	sit	0.012	came	0.012	floor	0.011	
	fall	0.011					
4	screw	0.063	assemble	0.035	hour	0.022	Yes (0.95)
	arm	0.022	back	0.019	arm_rest	0.018	
	hard	0.018	put_toget her	0.017	line	0.017	
	bolt	0.016					
5	small	0.042	size	0.029	feel	0.022	Yes (0.95)
	back	0.020	uncomfort able	0.018	sit	0.017	
	good	0.016	fine	0.015	fit	0.015	
	arm	0.015					

Table 4-11. Topic Model Words for BestOffice Office Chair

Amazon SageMaker human judgment indicated 4 out of 5 topics shows words related to furniture products or product issues. The 5 topics summarized the terms used in OCRs that mention defects. Words that support topic number 3 might indicate the product has a design problem with the wheels, with words such as “wheel,” “fall,” and “floor” mentioned. The following list shows three sampled OCRs that indicate assembly problems using the word search function with the word “pain” in the data.

- “I never advanced to any stage of assembly as I could not attach the wheels in any manner that they did not immediately fall off when the base was turned right side up. This indicates extremely poor manufacturing quality.” (review ID # R29WRWMILSIW5H)

- “The wheelbase is small the legs do not extend out from under the seat portion of this chair. If you do not sit down all the way to the back of the chair it will flip an you will land on the floor wondering *** just happened.” (review ID # RNWH1QPS6O7TE)
- “The wheel base is not long enough for the size of the seat. If you lean forward to get something while seated the chair dumps you on the floor and then I got hit right at the base of my neck by back of the chair causing a painful bruise.” (review ID # R1NF7I9TE5GGQX)

4.4.8 Test Case Eight: AmazomBasics Office Chair

Figure 4-12. AmazomBasics Office Chair on Amazon.com

AmazomBasics Office Chair (Figure 4-12) is an adjustable height armchair with tilting control. This product received 7,034-star ratings at an average of 4.2 out of 5 stars and 4,612 valid OCRs. The RNN classifier for classifying negative OCRs from non-negative OCRs identified 1,063 negative OCRs among the total valid OCRs. The second RNN classifier for classifying reviews that mention product defects from reviews that do not mention product defects identified 950 OCRs that mention product defects.

The LDA topic model was used to identify the 5 topics with 10 words each that best describe OCRs which mention product defects and generated a topic CV of 0.3942. The following table shows the topics, their supporting words, and their support weight. The topics were also given to Amazon SageMaker human labelers for their relevance to a furniture product or a product defect, along with its corresponding confidence level.

Topic	Words supporting the topic (supporting weight)						Is this topic relevant to the product or a defect? (Confidence)
	Words	Weight	Words	Weight	Words	Weight	

1	assemble	0.041	back	0.039	screw	0.033	Yes (0.95)
	bolt	0.024	one	0.023	hole	0.018	
	part	0.017	put	0.016	seat	0.012	
	hard	0.012					
2	use	0.029	year	0.029	like	0.022	Yes (0.95)
	last	0.018	seem	0.016	leather	0.016	
	purchase	0.015	first	0.014	cushion	0.013	
	time	0.012					
3	back	0.062	support	0.033	sit	0.024	Yes (0.90)
	seat	0.024	return	0.016	height	0.016	
	uncomfor table	0.015	adjust	0.014	cushion	0.014	
	tilt	0.014					
4	return	0.026	box	0.018	screw	0.017	Yes (0.95)
	apart	0.016	product	0.016	amazon	0.015	
	seat	0.014	lean	0.012	cushion	0.011	
	put _together	0.011					
5	seat	0.038	use	0.032	cushion	0.029	Yes (0.95)
	arm	0.018	pad	0.015	lean	0.015	

					_back		
	broke	0.014	arm_rest	0.013	side	0.013	
	feel	0.012					

Table 4-12. Topic Model Words for AmazonBasics Office Chair

Amazon SageMaker human judgment indicated 5 out of 5 topics shows words related to furniture products or product issues. The 5 topics summarized the terms used in OCRs that mention defects. Words that support topic 5 might indicate the product has a design problem with the armrest, with terms such as "arm_rest," "broke," and "uncomfortable" mentioned. The following list shows three sampled OCRs that indicate assembly problem using the word search function with the word "pain" in the data.

- "Got this chair for my new apartment, and it was great for one week until the arm broke. I'm not a big person, 5'8"/130, so I was surprised when the arm just snapped when I pushed on it getting up out of the seat." (review ID # RY4GBDIK3M8LV)
- "Barely 2 days after assembling the chair, I managed to crack the right side armrest as I was shifting my weight to correct my posture; I barely weigh 155lbs and the armrest didn't even bear my full weight before buckling. It's such a shame." (review ID # R3IZU22TEHUQ7W)
- "The arms also wobble and you feel like they might tear out of the seat at any time. Sitting in this chair for long periods is an absolute no go." (review ID # R3UTXEA0YHA09X)

4.4.9 Hypothesis 3

H3: A predictive model can identify 70% of key product defect topics for a single product using OCRs.

Hypothesis 3 evaluated the performance of the LDA topic model, which combines the above two RNN classifier, for retrieving key product defect topics. Amazon SageMaker human labelers were asked to evaluate if at least half of the group of words retrieved by the LDA topics are relevant to detail, buying process, usage, or defect of a furniture product. Since there are eight test cases demonstrated in this paper, one sample sign test was used for the hypothesis test. Table 4-13 shows the number of relevant topics generated by each of the eight products. The following table 4-14 shows the result of the one-sample sign test for the median.

Summary of test cases			
Test cases	Relevance	Topic CV	Average confidence
Test case one	5 out of 5 topics are related to a furniture product.	0.4930	0.948
Test case two	4 out of 5 topics are related to a furniture product.	0.3583	0.950
Test case three	5 out of 5 topics are related to a furniture product.	0.5162	0.948
Test case four	5 out of 5 topics are related to a furniture product.	0.4256	0.950

Test case five	5 out of 5 topics are related to a furniture product.	0.3959	0.944
Test case six	3 out of 5 topics are related to a furniture product.	0.3503	0.946
Test case seven	4 out of 5 topics are related to a furniture product.	0.3698	0.930
Test case eight	5 out of 5 topics are related to a furniture product.	0.3942	0.950

Table 4-13. Summary of Test Cases

One sample sign test for median			
H_0 : η = 0.7 H_1 : η > 0.7			
Sample (N)	8	Median	5
95% confidence interval for proportion	(3.936, 5)	P-value	0.035

Table 4-14. Hypothesis Test for LDA Topic Model

The test products have a mean of 4.375 out of 5 topics and a median of 5 out of 5 topics evaluated as relevant to a furniture product or problematic product information. They generated a mean topic CV of 0.4129 with all of them more than 0.3, which indicates the words in topics are somewhat coherent. Some test products generated a topic CV of higher than 0.5, which indicates that they have a good coherence between topics. A significance level, alpha, of 0.05 was used over all this praxis. Hypothesis 3 was tested with a one sample sign test to evaluate if the model can retrieve 70% of the topic relevant to a furniture product or problematic product information. With the P-value at 0.035, the hypothesis test successfully rejected the null hypothesis and thus, the LDA model can successfully retrieve key product defect topics.

4.5 Summary

Three quantitative models for OCR sentiment analysis were built, and the prediction data was collected and analyzed in this research. A summary of the results is provided in the following table 4-15.

Model	Test Type	Result
RNN model for classifying negative OCRs from non-negative OCRs	One sample Z-tests for a proportion (Hypothesis 1)	Null hypothesis Rejected
RNN classifier for classifying OCRs that mention product defects from OCRs that do not mention product defects	One sample Z-tests for a proportion (Hypothesis 2)	Null hypothesis Rejected

LDA topic model for providing product defect insight	One sample sign test	Null hypothesis Rejected

Table 4-15. Summary of Results

The first and second models successfully rejected the null hypothesis. This means that the RNN model for classification of negative OCRs from non-negative OCRs and for classification of OCRs that mention product defects from OCRs that do not mention product defects are working as expected. The LDA topic model, which combines the above two RNN classifiers, can retrieve key product defect topics from OCRs. Eight test cases were provided using the combined models of the first (RNN model for classifying negative OCRs), the second (RNN model for classifying OCRs that mention product defects), and the third (LDA topic model for providing product defect insight) model in this section. Those results confirmed that these combined models can pinpoint product issues as mentioned in OCRs. This means the combined models can provide engineers and customers with a fast and automatic way to have insights into product defects. A detailed discussion of this combined model is provided in chapter 5.

Chapter 5—Discussion and Conclusions

5.1 Discussion

The purpose of this research was to investigate the possibility of using probabilistic models for online customer reviews to retrieve product defect information and provide engineers with insight about the defects. This research used two recurrent neural network (RNN) classifiers and one Latent Dirichlet Allocation (LDA) topic to determine if these models can retrieve product defect information. The first RNN model (classifying negative online customer reviews (OCRs)) and the second RNN model (classifying OCRs that mention product defects) were successfully trained and tested. The LDA model also successfully combined the first and the second RNN models to retrieve key information on OCRs that are negative and mention product defects. This section will provide a discussion on the other RNN model that this research attempted to build and the insight about product defectiveness.

To further investigate the defective information embedded in the OCRs that were classified with defective information in model two, an RNN model was built and it attempted to identify what kind of defect type ("Manufacturing defect," "Problematic design or quality," and "Bad customer service.") was mentioned in the OCRs. Due to the similarity of the use of words among the OCRs mentioning product defects, that RNN model was not able to classify OCRs down to the defect type. It was overfitted with the classification type that has the heaviest weight. The assumption is that the root cause is because the words used in all three classes of labels were so similar that the model was unable to separate out the probability space. The initial assumption was that the model could identify the difference among these words, but that was proven not to be the case.

This limitation with this RNN approach provided an excellent space to use the LDA model. The LDA model, combined with the first and the second RNN models, successfully retrieved key information on OCRs that are negative and mention product defects. Instead of spending valuable time to read through all the OCRs to find out where the problem is, this combined model provides engineers with a fast and easy way to locate the problems that customers mention most frequently. As an example, in test case number eight (AmazomBasics Office Chair), the words “arm,” “side,” and “broke” were mentioned in topic 5, as well as the bigram “arm_rest”. These words pointed to a problem with the arm rest that may easily break. With a word search of ORCs with the word “arm,” the search indicated 305 OCRs consisting of the word “arm,” and a lot of these OCRs mentioned the broken arm rest, as seen in Section 4.4.8. Engineers can therefore use this model to locate and take immediate actions on this issue.

5.2 Conclusions

This research explored the best method to provide engineers with an early view into product defects. In current product development, OCRs have become an important way to explore customer opinions and complaints toward product defects. While there is a large number of OCRs available on social media and e-commerce sites, it is difficult for engineers to manually inspect those OCRs for defects information, which will delay product recall. This research has demonstrated a novel predictive model using RNN and LDA topic models to extract product defects from OCRs in a fast and automatic way. As the time to recall action increases during a product defect event, the recovery will be more challenging (Hora et al., 2011). This new predictive model provides the engineers with an early view into product defects, which enables the team to take required mitigation actions earlier and to proactively stop the spread of the detective products.

In summary, this research has proposed and constructed two RNN classifiers and one LDA topic to determine if these models can retrieve product defect information. The first RNN model was able to identify negative OCRs, where most of these OCRs consist of complaints and issues about the products. The second RNN model was able to identify if the OCRs consist of one of the following information labels about the products' defect: "Manufacturing defect," "Problematic design or quality," and "Bad customer service." The Latent Dirichlet Allocation (LDA) model successfully combined the first and the second RNN models to retrieve key defect information on OCRs that were negative and mentioned product defect. This combined model was able to locate the key words of the problems and issues that customers mentioned the most in their OCRs, which in turn achieved the goal of providing the engineers with an early view into potential product defect events.

5.3 Contributions to Body of Knowledge

This praxis adds to the body of knowledge in two major areas. The first area is in defect and recall management where there have been publications using different kinds of sentiment analysis to identify specific OCRs and gain product defect information such as using k-nearest neighbors (k-NN) algorithm, Random Forest (Decision Tree) (Bleaney et al., 2018), and term frequency (Abrahams et al., 2012). The literature review did not find any publication using RNN models on OCRs to gain defect insight. This praxis used

RNN to find negative OCRs and OCRs that mentioned defects, which were subsequently fed into an LDA model to provide engineers with product defect insight.

The second contribution is the combined RNN and LDA models to use OCRs from a single product to provide defect insights using a set of keywords. In the area of product defect discovery, there has been one publication that used the LDA topic model to aggregate product defects and solutions information among online customer complaints in recent years (Zhang et al., 2019). The literature review did not find any publication that was able to analyze all OCRs, regardless of whether the OCRs were complaints or not. A lot of e-commerce websites, such as Amazon.com and Google Shopping, allow customers to leave constructive comments on products, including positive ones. This praxis adds to the body of knowledge with this novel, predictive model using RNN and LDA that examines all OCRs on a product to provide engineers with product defect insight.

5.4 Recommendations for Future Research

Future research may consider improving and refining the accuracy of both the RNN classifiers and the LDA topic model, and addressing the main outstanding problem identified in this research: the RNN classifier model for identifying defect types.

RNN classifiers may be improved by adding multiple LSTM layers to construct a deeper neural network architecture. The accuracy may be improved this way, although the training time may increase. RNN classifiers may also be improved by training on a larger supervised dataset, while this would involve more manual labor work. The other neural network architectures such as Convolutional Neural Network layers can also be explored in future research for increasing the performance of identifying defective products.

The deficiency for the RNN classifier that was attempted to identify defect types could be solved by constructing term frequency word lists for each defect type. Abrahams et al. have developed a set of smoke words, specifically for auto defect and auto safety issues, to identify whether OCRs mentioned defect or safety issues (Abrahams et al., 2012). This method may be harder for particular classifiers to differentiate "Manufacturing defect" from "Problematic design or quality," since they have a very similar use of words in the OCRs.

LDA topic models may be enhanced by constructing a word dictionary for some known defect keywords and assigning them to specific topic before putting the OCRs into the LDA topics. Zhang el al. have done similar work on assigning known defect keywords belonging to a specific component, symptom, and resolution topics (Zhang el al., 2019). This may increase the topic coherence with a higher coherence value, as well as decreasing the irrelevant words aggregated by the LDA topic model.

In summary, this research has demonstrated a fast and practical approach for extracting product defect information from thousands of OCRs that belong to a particular product. This novel approach to defect management may help product engineers to locate product defects and shorten product recall and recovery time. While this praxis only collected data and tested on the home furnishing industry, this approach can be expanded easily to other industries for future research and further study.

References

Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A., & Arshad, H. (2018) State-of-the-art in artificial neural network applications: A survey. Heliyon 4(11)

Abney S., Schapire R., & Singer Y. (1999) Boosting applied to tagging and PP Attachment. In Proceedings of EMNLP/VLC-99, pages 38--45

Abrahams, A. S., Jiao, J., Wang, G. A., & Fan, W. (2012). Vehicle defect discovery from social media. *Decision Support Systems*, *54*(1), 87–97. doi: 10.1016/j.dss.2012.04.005

Ahsan, K. (2013). Trend Analysis of Car Recalls: Evidence from the US Market. *International Journal of Managing Value and Supply Chains*, 4(4), 1–16. doi: 10.5121/ijmvsc.2013.4401

Agarwal, S., Godbole, S., Punjani, D., & Roy, S. (2007). How Much Noise Is Too Much: A Study in Automatic Text Classification. *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. doi: 10.1109/icdm.2007.21

Aggarwal, C. C. (2015). Data classification: algorithms and applications. Boca Raton: CRC.

Barton, B. (2006). Ratings, Reviews & ROI. *Journal of Interactive Advertising*, *7*(1), 5–50. doi: 10.1080/15252019.2006.10722125

Beck, J. E., & Woolf, B. P. (2000). High-Level Student Modeling with Machine Learning. *Intelligent Tutoring Systems Lecture Notes in Computer Science*, 584–593. doi: 10.1007/3-540-45108-0_62

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166. doi: 10.1109/72.279181

Berman, B. (1999). Planning for the inevitable product recall. *Business Horizons*, *42*(2), 69–78. doi: 10.1016/s0007-6813(99)80011-1

Bleaney, G., Kuzyk, M., Man, J., Mayanloo, H., & Tizhoosh, H. R. (2018). Auto-detection of Safety Issues in Baby Products. *Lecture Notes in Computer Science Recent Trends and Future Technology in Applied Intelligence*, 505–516. doi: 10.1007/978-3-319-92058-0_49

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3, 993–1022. doi: 993–1022

Boben, G. (2019). Analyzing Amazon TV reviews with Latent Dirichlet Allocation, Retrieved from https://medium.com/analytics-vidhya/analyzing-amazon-tv-reviews-with-latent-dirichlet-allocation-97d19c3bb878

Bridle, J. S. (1990). Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition. *Neurocomputing*, 227–236. doi: 10.1007/978-3-642-76153-9_28

Chaklader, R., & Parkinson, M. B. (2017). Data-Driven Sizing Specification Utilizing Consumer Text Reviews. *Journal of Mechanical Design*, *139*(11). doi: 10.1115/1.4037476

Chan, N. L., & Guillet, B. D. (2011). Investigation of Social Media Marketing: How Does the Hotel Industry in Hong Kong Perform in Marketing on Social Media

Websites? *Journal of Travel & Tourism Marketing*, *28*(4), 345–368. doi: 10.1080/10548408.2011.571571

CPSC (2019, December 20). About CPSC. Bethesda, MD: U.S. Consumer Product Safety Commission Retrieved from https://www.cpsc.gov/About-CPSC

Dellarocas, C., Zhang, X., & Awad, N. F. (2007). Exploring the value of online product reviews in forecasting sales: The case of motion pictures. *Journal of Interactive Marketing*, *21*(4), 23–45. doi: 10.1002/dir.20087

Dickinson, M., & Meurers, W. D. (2003). Detecting errors in part-of-speech annotation. *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - EACL 03*. doi: 10.3115/1067807.1067823

DoD. (2015) Risk, Issue, and Opportunity Management Guide for Defense Acquisition Programs. Washington, DC, USA: Office of the Deputy Assistant Secretary of Defense for Systems Engineering/Department of Defense.

El-Dehaibi, N., & Macdonald, E. F. (2019). Extracting Customer Perceptions of Product Sustainability From Online Reviews. *Volume 2B: 45th Design Automation Conference*. doi: 10.1115/detc2019-98233

Eskin, E. (2000). Detecting Errors within a Corpus using Anomaly Detection. *NAACL 2000: Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, 148–153. Retrieved from https://dl.acm.org/doi/10.5555/974305.974325

Graves, A. (2012). Offline Arabic Handwriting Recognition with Multidimensional Recurrent Neural Networks. *Guide to OCR for Arabic Scripts*, 297–313. doi: 10.1007/978-1-4471-4072-6_12

Golding, A.R., & Roth, D. (1999) A Winnow-Based Approach to Context-Sensitive Spelling Correction. Machine Learning 34, 107–130. doi: 10.1023/A:1007545901558

Han, J., Kamber, M., & Pei, J. (2012). *Data mining: concepts and techniques*. Waltham, MA: Morgan Kaufmann.

Hochreiter, S. (1998). The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *06*(02), 107–116. doi: 10.1142/s0218488598000094

Hora, M., Bapuji, H., & Roth, A. V. (2011). *Safety hazard and time to recall: The role of recall strategy, product defect type, and supply chain player in the U.S. toy industry. Journal of Operations Management,29(7-8), 766-777.* doi:10.1016/j.jom.2011.06.006

Ho-Dac, N. N., Carson, S. J., & Moore, W. L. (2013). The Effects of Positive and Negative Online Customer Reviews: Do Brand Strength and Category Maturity Matter? *Journal of Marketing*, *77*(6), 37–53. doi: 10.1509/jm.11.0011

Hubbard, D. W. (2009). The failure of risk management: why its broken and how to fix it. J. Wiley & Sons.

IEEE (2006) ISO/IEC 16085:2006, Standard for Software Engineering - Software Life Cycle Processes - Risk Management. doi: 10.1109/ieeestd.2006.288594

Jarrell, G., & Peltzman, S. (1985). The Impact of Product Recalls on the Wealth of Sellers. Journal of Political Economy,93(3), 512-536. doi:10.1086/261313

Jarrett, K., Kavukcuoglu, K., Ranzato, M. A., & Lecun, Y. (2009). What is the best multi-stage architecture for object recognition? *2009 IEEE 12th International Conference on Computer Vision*. doi: 10.1109/iccv.2009.5459469

Jeong, E.-Y. (2017) Samsung's Conclusions on Galaxy Note 7 Recall Backed Up by Regulator. *The Wall Street Journal*. Retrieved from https://www.wsj.com/articles/samsungs-conclusions-on-galaxy-note-7-recall-backed-up-by-regulator-1486348636

Jeong, E. (2019). Samsung to Recall Galaxy Note 7 Smartphone Over Reports of Fires. Retrieved from https://www.wsj.com/articles/samsung-to-recall-galaxy-note-7-smartphone-1472805076

Jiang, H., Kwong, C. K., & Yung, K. L. (2017). Predicting Future Importance of Product Features Based on Online Customer Reviews. *Journal of Mechanical Design*, *139*(11). doi: 10.1115/1.4037348

Keras Special Interest Group (2020). Keras. Retrieved from https://keras.io/

Karakaya, F., & Barnes, N. G. (2010). Impact of online reviews of customer care experience on brand or company selection. *Journal of Consumer Marketing*, *27*(5), 447–457. doi: 10.1108/07363761011063349

Kreek, R. A., & Apostolova, E. (2018). Training and Prediction Data Discrepancies: Challenges of Text Classification with Noisy, Historical Data. *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-Generated Text*. doi: 10.18653/v1/w18-6114

Landgrebe, J., & Smith, B. (2019). There is no Artificial General Intelligence. Retrieved from https://arxiv.org/abs/1906.05833

Lee, T., & Bradlow, E. (2010). Automated Marketing Research Using Online Customer Reviews. *SSRN Electronic Journal*. doi: 10.2139/ssrn.1726055

Li, J., Luong, T., Jurafsky, D., & Hovy, E. (2015). When Are Tree Structures Necessary for Deep Learning of Representations? *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. doi: 10.18653/v1/d15-1278

Liu P., Qiu X., & Huang X., (2016). Recurrent neural network for text classification with multi-task learning. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16). AAAI Press, 2873–2879. doi: 10.5555/3060832.3061023

Lyles, M. A., Flynn, B. B., & Frohlich, M. T. (2008). All Supply Chains Dont Flow Through: Understanding Supply Chain Issues in Product Recalls. *Management and Organization Review*, *4*(2), 167–182. doi: 10.1111/j.1740-8784.2008.00106.x

Mack R. (2016, September 15). Another Samsung Galaxy Note 7 up in smoke as users ignore recalls. CNET . Retrieved from https://www.cnet.com/news/another-samsung-galaxy-note-7-up-in-smoke-recalled-phone-not-put-down-data-shows/

Magno, F. (2012). Managing Product Recalls: The Effects of Time, Responsible vs. Opportunistic Recall Management and Blame on Consumers' Attitudes. Procedia - Social and Behavioral Sciences, 58, 1309–1315. doi: 10.1016/j.sbspro.2012.09.1114

Matos, C. A. D., & Rossi, C. A. V. (2007). Consumer reaction to product recalls: factors influencing product judgement and behavioural intentions. *International Journal of Consumer Studies*, *31*(1). doi: 10.1111/j.1470-6431.2006.00499.x

McColloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. doi: 10.1007/bf02478259

Mikolov, T., Chen, K., Corrado, G., & Dean, J.: Efficient Estimation of Word Representations in Vector Space. In: ICLR: Proceeding of the International Conference on Learning Representations Workshop Track, Arizona, USA, pp. 1301–3781 (2013) https://arxiv.org/abs/1301.3781

Mikolov T., Sutskever I., Chen K., Corrado G., & Dean J. (2013). Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13). Curran Associates Inc., Red Hook, NY, USA, 3111–3119. doi: 10.5555/2999792.2999959

Nair V., Hinton G. (2010). Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML'10). Omnipress, Madison, WI, USA, 807–814. doi: 10.5555/3104322.3104425

Nasukawa, T., & Yi, J. (2003). Sentiment analysis: Capturing favorability using natural language processing. *Proceedings of the International Conference on Knowledge Capture - K-CAP 03*. doi: 10.1145/945645.945658

Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. doi: 10.1561/9781601981516

Pearlmutter. (1989). Learning state space trajectories in recurrent neural networks. *International Joint Conference on Neural Networks*. doi: 10.1109/ijcnn.1989.118724

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. doi: 10.3115/v1/d14-1162

Ren, Y., Wang, R., & Ji, D. (2016). A topic-enhanced word embedding for Twitter sentiment classification. *Information Sciences*, *369*, 188–198. doi: 10.1016/j.ins.2016.06.040

Rezaeinia, S. M., Rahmani, R., Ghodsi, A., & Veisi, H. (2019). Sentiment analysis based on improved pre-trained word embeddings. *Expert Systems with Applications*, *117*, 139–147. doi: 10.1016/j.eswa.2018.08.044

Rose, S., Hair, N., & Clark, M. (2011). Online Customer Experience: A Review of the Business-to-Consumer Online Purchase Context. *International Journal of Management Reviews, 13(1), 24-39.* doi:10.1111/j.1468-2370.2010.00280.x

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. doi: 10.1038/323533a0

Russell, S. J., & Norvig, P. (2011). Artificial intelligence: a modern approach. Hoboken: Pearson.

Sanchanta, M., & Takahashi, Y. (2010, March 09). The Wall Street journal. Toyota's Recall Costs Could Top $5 Billion. *Retrieved from* https://www.wsj.com/articles/SB10001424052748704145904575111341893725992

Shahin, A., & Chan, J. F. L. (2006). Customer requirements segmentation (CRS): A prerequisite technique for quality function deployment (QFD). *Total Quality Management & Business Excellence*, *17*(5), 567–587. doi: 10.1080/14783360600588117

Siomkos, G. J., & Kurzbard, G. (1994). The Hidden Crisis in Product□harm Crisis Management. *European Journal of Marketing*, *28*(2), 30–41. doi: 10.1108/03090569410055265

Souiden, N., & Pons, F. (2009). Product recall crisis management: the impact on manufacturers image, consumer loyalty and purchase intention. *Journal of Product & Brand Management*, *18*(2), 106–114. doi: 10.1108/10610420910949004

Steffes, E. M., & Burgee, L. E. (2009). Social ties and online word of mouth. *Internet Research*, *19*(1), 42–59. doi: 10.1108/10662240910927812

Suryadi, D., & Kim, H. M. (2019). A Data-Driven Approach to Product Usage Context Identification From Online Customer Reviews. *Journal of Mechanical Design*, *141*(12). doi: 10.1115/1.4044523

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., & Qin, B. (2014). Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. doi: 10.3115/v1/p14-1146

Tang, D., Qin, B., & Liu, T. (2015). Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.* doi:10.18653/v1/d15-1167

Tang, J., Li, H., Cao, Y., & Tang, Z. (2005). Email data cleaning. Proceeding of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining - KDD 05. doi: 10.1145/1081870.1081926

Tibken, S., & Cheng, R. (2017, January 23). Samsung answers burning Note 7 questions, vows better batteries. CNET . Retrieved from https://www.cnet.com/news/samsung-answers-burning-note-7-questions-vows-better-batteries/

Turing, A. M. (1950). I.—Computing Machinery And Intelligence. *Mind*, *LIX*(236), 433–460. doi: 10.1093/mind/lix.236.433

Walden, D. D., Roedler, G. J., Forsberg, K., Hamelin, R. D., & Shortell, T. M. (2015). *Systems engineering handbook: a guide for system life cycle processes and activities*. Wiley.

Wei, P.-S., Lu, H.-P. (2013). An examination of the celebrity endorsements and online customer reviews influence female consumers' shopping behavior. *Computers in Human Behavior*, *29*(1), 193–201. doi: 10.1016/j.chb.2012.08.005

Werbos, P. J. (1975). Beyond regression: new tools for prediction and analysis in the behavioral sciences. *Ph. D. Thesis, Harvard University,* Cambridge, MA

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2017). *Data mining: practical machine learning tools and techniques*. Amsterdam: Morgan Kaufmann.

Witty, R. (2015). Use KPI and KRI Mapping to Make the Business Case for Business Resilience. *Gartner*. Retrieved from https://www.gartner.com/document/3135625

Zhai, C. X., & Massung, S. (2016). Text data management and analysis: a practical introduction to information retrieval and text mining. New York, NY: ACM Books.

Zhang, W., Du, T., & Wang, J. (2016). Deep Learning over Multi-field Categorical Data. *Lecture Notes in Computer Science Advances in Information Retrieval*, 45–57. doi: 10.1007/978-3-319-30671-1_4

Zhang, X., Qiao, Z., Ahuja, A., Fan, W., Fox, E. A., & Reddy, C. K. (2019). Discovering Product Defects and Solutions from Online User Generated Contents. The World Wide Web Conference on - WWW 19. doi: 10.1145/3308558.3313732

Zhang, Z., Guo, C., & Goes, P. (2013). Product Comparison Networks for Competitive Analysis of Online Word-of-Mouth. *ACM Transactions on Management Information Systems*, *3*(4), 1–22. doi: 10.1145/2407740.2407744

Zhou, F., Ayoub, J., Xu, Q., & Yang, X. J. (2019). A Machine Learning Approach to Customer Needs Analysis for Product Ecosystems. *Journal of Mechanical Design*, *142*(1). doi: 10.1115/1.4044435

Appendix A: Sampled OCRs Input Data with AWS SageMaker Annotation

This appendix provides the 10 sampled raw data with AWS SageMaker Annotation. This is used in training and testing the RNN models.

class-name	confidence	creation-date	human-annotated	job-name	source	type	review-id	customer-id	product_id
No defect information provided	0.95	2020-01-31T07:55:00.528476	yes	labeling-job/research-classify-defect	Great buy! We upgraded our 6 year old from a toddler bed and this was a good affordable bed frame.	groundtruth/text-classification	R138YASBONWZ7R	11188064	B002TSAE70
Problematic design or quality	0.94	2019-12-10T07:06:54.525607	yes	labeling-job/research-classify-defect	"Packaged fine, easy to put together. Once I put it together I noticed I was tilting to the right side. Another reviewer on here had the same problem. Returning for refund. Disappointing."	groundtruth/text-classification	R1HUPYFUX1IZVM	24288830	B00G4LP9RI
Bad customer service	0.92	2019-12-10T05:47:05.374013	yes	labeling-job/research-classify-defect	"I have called - I have e-mailed only to be told that I need to return the netting to the store (only it was purchased more than 90 days ago) as a Xmas gift, which target doesn't not take returns past 90 days. Who would of thought that the mosquito netting would'nt fit. It is about 4inches to short, so being the domestic engineer that I'am, tried it on the inside of the Gazebo instead of the outside (it fits) Who engineered this??? I called the company after nearly a week they finely called me back; they stated that they are sending me a replacement - after I threated to file a complaint with the MN States Attorney's Office. We'll see if this one fits - the only reason I wanted it was because of the netting and all the mosquito that we get here in Minnesota!!"	groundtruth/text-classification	R185ZW73D0H2AF	17733203	B0000E667T
Problematic design or quality	0.95	2020-01-31T07:34:05.651894	yes	labeling-job/research-classify-defect	"The bed has been in use just 11 days by my 7 and 9 year old grandkids and the bottom trundle is already unusable , it will be taken down boxed up and returned this weekend.The slats have split at 3 places she is now back on the floor."	groundtruth/text-classification	R2EIGZZNOJJ80T	1477363	B001B3RET0

Problematic design or quality	0.95	2019-12-10T06:20:05.385374	yes	labeling-job/research-classify-defect	I would not advise anyone to buy this item. It was the Hardest thing to put together I had ever seen. Nothing fit. We had to put boards behind it to hold it together. The holes did not line up and the hard ware did not work we had to buy wooden dowels and glue them in to try to make them work. The headboard looks like crap. I would rate it as a minus 5.	groundtruth/text-classification	R1Z8JTBUHE8F6S	5376379	B00CUH0DN0
No defect information provided	0.94	2020-01-31T06:50:01.800415	yes	labeling-job/research-classify-defect	"Very elegant,i received many complements about this table."	groundtruth/text-classification	R2ATFFSKABK2G8	10182744	B00JIKOFHC
Manufacturing defect	0.76	2020-01-31T07:08:45.687478	yes	labeling-job/research-classify-defect	"Poorly made, and will not last. Minor pieces bent early in ownership and just after 1 year the base snapped at the weld rendering product a complete waste. The company refused to provide any customer service despite multiple requests for assistance, replacement of part etc... Photos were provided of the broken base. The chair completely collapsed. BTW I weigh155 lbs... the chair is dangerous and company did not care. I'm very dissatisfied. Fortunately I was not injured when the base broke. Save yourself the headache and wasted money. AVOID this awful chair & company."	groundtruth/text-classification	R3BF0Y0MNK5E91	12160987	B006B81068
No defect information provided	0.95	2020-01-31T07:53:28.338287	yes	labeling-job/research-classify-defect	This product has worked out perfectly for us. We use it as a seperate working station in our kitchen and it has accomplished what we purchased it for. It has met all of our expectations. I would really recommend this table.	groundtruth/text-classification	labeling-job/research-classify-defect		
Problematic design or quality	0.77	2020-01-31T06:24:22.187210	yes	labeling-job/research-classify-defect	"We like the looks of the piece of furniture. Weren't so pleased with some of the little construction/shipping flaws but they are easily fixed. My wife and I had the stand out of the box and assembled in a couple of hours - and we are definitely senior citizens. Solid, well designed and fits right in with our other living room furniture."	groundtruth/text-classification	R2ZEC22H83EAXY	19160333	B000NPTUUI

Manufacturing defect	0.91	2020-01-31T06:24:51.334721	yes	labeling-job/research-classify-defect	"The hardware mainly the allen head screews are junk, once they go it you will not get them out, thus leaving it impossible to move the bed later or correct assymboly errors I had to jimmy rig several steps, one allen head was lacking the allen cut out and not a single spare screew in the hardware kit."	groundtruth/text-classification	R2RIYWOPU92HZC	28270448	B002WRGEK4

Appendix B: Computational Python Code

This appendix provides the computational python code to both testing and training the RNN models, as well as the LDA topic model for outputting the product defect LDA topics. The code is based upon Keras.io for the RNN models (Keras, 2020) and Boben's LDA topic model for Amazon TV shows (Boben, 2019).

```
import tensorflow as tf
import tensorflow_datasets as tfds

import numpy as np
import pandas as pd
from pandas.io.json import json_normalize


from string import punctuation
from collections import Counter

import re
import os
import json

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import roc_curve
from sklearn.metrics import f1_score, precision_score, recall_score,
confusion_matrix
from sklearn.metrics import auc

import string
from string import punctuation
```

```
from nltk.corpus import stopwords
from nltk.stem import SnowballStemmer
from nltk.stem import PorterStemmer, WordNetLemmatizer


from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Embedding
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.layers import LSTM
from tensorflow.keras.layers import Conv1D, MaxPooling1D

import gensim
import pyLDAvis
import pyLDAvis.gensim
import pandas as pd
import gensim.corpora as corpora
from gensim.utils import simple_preprocess
from gensim.models import CoherenceModel
import warnings

Model_input = pd.read_json('C:/Users/inputRawData.json', lines=True)
Model_input['useful'] = np.where(Model_input['class-name'] == "No defect
information provided", 0, 1)

Model_input["source"] = Model_input["source"].replace({""':"}, regex=True)
Model_input["source"] = Model_input["source"].replace({'br':"}, regex=True)
Model_input["source"] = Model_input["source"].str.lower()
```

```
Model_input["source"] = Model_input["source"].str.replace(r"[^a-zA-Z ]+", "
").str.strip()
Model_input["source"] = Model_input["source"].str.replace(r" +", " ").str.strip()

## RNN Model 1
Model_input['source'] = Model_input['source'].fillna(' ')
sentences = Model_input['source'].values
y = Model_input['posneg'].values
sentences_train, sentences_test, y_train, y_test = train_test_split(sentences, y,
test_size=0.1, train_size = 0.9, random_state=1000)

tokenizer = Tokenizer(num_words=5000, filters='!"#$%&()*+,-
./:;<=>?@[\\]^_`{|}~\t\n', lower=True, split=' ', char_level=False)
tokenizer.fit_on_texts(sentences_train)

X_train = tokenizer.texts_to_sequences(sentences_train)
X_test = tokenizer.texts_to_sequences(sentences_test)

vocab_size = len(tokenizer.word_index) + 1  # Adding 1 because of reserved 0 index

maxlen = 300

X_train = pad_sequences(X_train, padding='post', maxlen=maxlen)
X_test = pad_sequences(X_test, padding='post', maxlen=maxlen)

BASE_DIR = 'C:/Users/Downloads/'
GLOVE_DIR = os.path.join(BASE_DIR, 'glove.6B')

embeddings_index = {}
f = open(os.path.join(GLOVE_DIR, 'glove.6B.100d.txt'), encoding='utf-8')
for line in f:
```

```
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype='float32')
    embeddings_index[word] = coefs
f.close()
word_index = tokenizer.word_index
EMBEDDING_DIM = 100
embedding_matrix = np.zeros((len(word_index) + 1, EMBEDDING_DIM))
for word, i in word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        # words not found in embedding index will be all-zeros.
        embedding_matrix[i] = embedding_vector

with open('tokenizer.pickle', 'wb') as handle:
    pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST_PROTOCOL)

    from keras import backend as K

def recall_m(y_true, y_pred):
        true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
        possible_positives = K.sum(K.round(K.clip(y_true, 0, 1)))
        recall = true_positives / (possible_positives + K.epsilon())
        return recall

def precision_m(y_true, y_pred):
        true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
        predicted_positives = K.sum(K.round(K.clip(y_pred, 0, 1)))
        precision = true_positives / (predicted_positives + K.epsilon())
        return precision
```

```
def f1_m(y_true, y_pred):
    precision = precision_m(y_true, y_pred)
    recall = recall_m(y_true, y_pred)
    return 2*((precision*recall)/(precision+recall+K.epsilon()))

checkpoint_path = "training_model1/cp.ckpt"
checkpoint_dir = os.path.dirname(checkpoint_path)
cp_callback = tf.keras.callbacks.ModelCheckpoint(filepath=checkpoint_path,
                                                 save_weights_only=True,
                                                 verbose=1)

max_features = 10000
batch_size = 32
embedding_size = 128
kernel_size = 5
filters = 64
pool_size = 4
lstm_output_size = 70
print('Build model...')
model = Sequential()
model.add(Embedding(len(word_index) + 1,
                    EMBEDDING_DIM,
                    weights=[embedding_matrix],
                    input_length=maxlen,
                    trainable=False))
model.add(Dropout(0.35))
model.add(Conv1D(filters,
                 kernel_size,
                 padding='valid',
                 activation='relu',
                 strides=1))
```

```
model.add(MaxPooling1D(pool_size=pool_size))
model.add(LSTM(lstm_output_size))
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer='adam',
            loss='binary_crossentropy',
            metrics=['accuracy',f1_m,precision_m, recall_m])
model.summary()

history = model.fit(X_train, y_train,
                epochs=50,
                verbose=2,
                validation_data=(X_test, y_test),
                batch_size=batch_size)

                import matplotlib.pyplot as plt
plt.style.use('ggplot')
def plot_history(history):
    acc = history.history['accuracy']
    val_acc = history.history['val_accuracy']
    loss = history.history['loss']
    val_loss = history.history['val_loss']
    x = range(1, len(acc) + 1)

    plt.figure(figsize=(12, 5))
    plt.subplot(1, 2, 1)
    plt.plot(x, acc, 'b', label='Training acc')
    plt.plot(x, val_acc, 'r', label='Validation acc')
    plt.title('Training and validation accuracy')
    plt.legend()
    plt.subplot(1, 2, 2)
```

```
        plt.plot(x, loss, 'b', label='Training loss')
        plt.plot(x, val_loss, 'r', label='Validation loss')
        plt.title('Training and validation loss')
        plt.legend()

    loss, accuracy, f1_score, precision, recall = model.evaluate(X_train, y_train,
verbose=False)
    print("--Classify negative and non-negative customer reviews, Model 1--")
    print("Training Accuracy: {:.4f} Training F1 Score: {:.4f}".format(accuracy,
f1_score))
    print("Training Precision: {:.4f} Training Recall: {:.4f}".format(precision, recall))
    print("------------------------------------------------------------------------------")
    loss, accuracy, f1_score, precision, recall = model.evaluate(X_test, y_test,
verbose=False)
    print("Testing Accuracy: {:.4f} Testing F1 Score: {:.4f}".format(accuracy, f1_score))
    print("Testing Precision: {:.4f} Testing Recall: {:.4f}".format(precision, recall))
    plot_history(history)

    y_pred_keras = model.predict(X_test).ravel()
    fpr_keras, tpr_keras, thresholds_keras = roc_curve(y_test, y_pred_keras)
    auc_keras = auc(fpr_keras, tpr_keras)
    plt.figure(1)
    plt.plot([0, 1], [0, 1], 'k--')
    plt.plot(fpr_keras, tpr_keras, label='Keras (area = {:.3f})'.format(auc_keras))
    plt.xlabel('False positive rate')
    plt.ylabel('True positive rate')
    plt.title('ROC curve')
    plt.legend(loc='best')
    plt.show()

    ## RNN Model 2
```

```
Model_input['source'] = Model_input['source'].fillna(' ')
sentences = Model_input['source'].values
y = Model_input['useful'].values
sentences_train, sentences_test, y_train, y_test = train_test_split(sentences, y,
test_size=0.1, train_size = 0.9, random_state=1000)

checkpoint_path_2 = "training_model2/cp.ckpt"
cp_callback_2 = tf.keras.callbacks.ModelCheckpoint(filepath=checkpoint_path_2,
                                  save_weights_only=True,
                                  verbose=1)
max_features = 10000
batch_size = 32
embedding_size = 128
kernel_size = 5
filters = 64
pool_size = 4
lstm_output_size = 70
print('Build model...')
modelTwo = Sequential()
modelTwo.add(Embedding(max_features, 128))

modelTwo.add(Dropout(0.2))
modelTwo.add(Conv1D(filters,
           kernel_size,
           padding='valid',
           activation='relu',
           strides=1))
modelTwo.add(MaxPooling1D(pool_size=pool_size))
modelTwo.add(LSTM(lstm_output_size))
modelTwo.add(Dense(1, activation='sigmoid'))
```

```
modelTwo.compile(optimizer='adam',
            loss='binary_crossentropy',
            metrics=['accuracy',f1_m,precision_m, recall_m])
modelTwo.summary()

history = modelTwo.fit(X_train, y_train,
                epochs=50,
                verbose=2,
                validation_data=(X_test, y_test),
                batch_size=batch_size,
                callbacks=[cp_callback_2])

loss, accuracy, f1_score, precision, recall = modelTwo.evaluate(X_train, y_train,
verbose=False)
print("--Classify useful and useless customer reviews, Model 2--")
print("Training Accuracy: {:.4f} Training F1 Score: {:.4f}".format(accuracy,
f1_score))
print("Training Precision: {:.4f} Training Recall: {:.4f}".format(precision, recall))
print("-----------------------------------------------------------------------------")
loss, accuracy, f1_score, precision, recall = modelTwo.evaluate(X_test, y_test,
verbose=False)
print("Training Accuracy: {:.4f} Training F1 Score: {:.4f}".format(accuracy,
f1_score))
print("Training Precision: {:.4f} Training Recall: {:.4f}".format(precision, recall))
plot_history(history) #Plot traning History
plt.show() #Plot ROC Curve

## RNN Model for further classify the defect type
Model_3_input = Model_input[Model_input.useful != 0]
Model_3_input['source'] = Model_3_input['source'].fillna(' ')
```

```
sentences = Model_3_input['source'].values
y = pd.get_dummies(Model_3_input['class-name']).values
sentences_train, sentences_test, y_train, y_test = train_test_split(sentences, y,
test_size=0.1, train_size = 0.9, random_state=1000)

checkpoint_path_3 = "training_model3/cp.ckpt"
cp_callback_3 = tf.keras.callbacks.ModelCheckpoint(filepath=checkpoint_path_3,
                                                   save_weights_only=True,
                                                   verbose=1)
max_features = 10000
batch_size = 128
embedding_size = 128
kernel_size = 5
filters = 64
pool_size = 4
lstm_output_size = 70
print('Build model...')
adam = Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.999, amsgrad=False)
modelThree = Sequential()
modelThree.add(Embedding(max_features, embedding_size))
model.add(Conv1D(filters,
                 kernel_size,
                 padding='valid',
                 activation='relu',
                 strides=1))
modelThree.add(LSTM(128,return_sequences=True, dropout=0.2,
recurrent_dropout=0.2))
modelThree.add(Dense(3, activation='softmax'))
modelThree.compile(optimizer=adam, loss='categorical_crossentropy',
metrics=['accuracy'])
modelThree.summary()
```

```
history = modelThree.fit(X_train, y_train,
                    epochs=50,
                    verbose=2,
                    validation_data=(X_test, y_test),
                    batch_size=batch_size,
                    callbacks=[cp_callback_3])

loss, accuracy = modelThree.evaluate(X_train, y_train, verbose=False)
print("Training Accuracy: {:.4f}".format(accuracy))
loss, accuracy = modelThree.evaluate(X_test, y_test, verbose=False)
print("Testing Accuracy:  {:.4f}".format(accuracy))
plot_history(history)

#LDA Topic Model
InputDf = pd.read_csv('C:/Users/office_chair_2.csv') #OCRs input of a product

InputDf["content"] = InputDf["content"].str.lower()
InputDf["content"] = InputDf["content"].str.replace(r"[^a-zA-Z ]+", " ").str.strip()
InputDf["content"] = InputDf["content"].str.replace(r" +", " ").str.strip()

X_test = tokenizer.texts_to_sequences(sentences)
X_test = pad_sequences(X_test, padding='post', maxlen=maxlen)

#Run Model one and drop negative OCRs
checkpoint_path = "training_model1_v2/cp.ckpt"
checkpoint_dir = os.path.dirname(checkpoint_path)
model.load_weights(checkpoint_path)
prediction  = model.predict(X_test)
InputDf['posneg'] = np.where(prediction >=0.5, 0, 1)
InputDfTwo = InputDf.drop(InputDf[InputDf.posneg == 0].index)
```

```
#Run Model one and drop useless OCRs
checkpoint_path_2 = "training_model2/cp.ckpt"
checkpoint_dir_2 = os.path.dirname(checkpoint_path_2)
modelTwo.load_weights(checkpoint_path_2)
predictTwo  = modelTwo.predict(X_test_Two)

#Text data clean up for LDA model
InputDfTwo['useful'] = np.where(predictTwo >=0.5, 1, 0)
stopwords = set(stopwords.words('english'))
InputDfTwoLDA['content'] = InputDfTwoLDA['content'].apply(lambda x: '
'.join([item for item in x.split() if item not in stopwords]))
porter_stemmer = PorterStemmer()
def stem_sentences(sentence):
    tokens = sentence.split()
    stemmed_tokens = [porter_stemmer.stem(token) for token in tokens]
    return ' '.join(stemmed_tokens)
InputDfTwoLDA['content'] = InputDfTwoLDA['content'].apply(stem_sentences)
def sent_to_words(sentences):
    for sentence in sentences:
        yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))
def bigrams(words, bi_min=15, tri_min=10):
    bigram = gensim.models.Phrases(words, min_count = bi_min)
    bigram_mod = gensim.models.phrases.Phraser(bigram)
    return bigram_mod
def get_corpus(df):
    words = list(sent_to_words(df.content))
    bigram = bigrams(words)
    bigram = [bigram[review] for review in words]
    id2word = gensim.corpora.Dictionary(bigram)
    id2word.filter_extremes(no_below=10, no_above=0.35)
```

```
    id2word.compactify()
    corpus = [id2word.doc2bow(text) for text in bigram]
    return corpus, id2word, bigram

train_corpus4, train_id2word4, bigram_train4 = get_corpus(InputDfTwoLDA)

with warnings.catch_warnings():
    warnings.simplefilter('ignore')
    lda_train4 = gensim.models.ldamulticore.LdaMulticore(
                        corpus=train_corpus4,
                        num_topics=5,
                        id2word=train_id2word4,
                        chunksize=100,
                        workers=7, # Num. Processing Cores - 1
                        passes=100,
                        eval_every = 1,
                        per_word_topics=True)
    lda_train4.save('lda_train4.model')

coherence_model_lda = CoherenceModel(model=lda_train4, texts=bigram_train4, 
dictionary=train_id2word4, coherence='c_v')
print(coherence_model_lda.get_coherence())

lda_train4.print_topics(20,num_words=10)[:10]
```

els (Keras, 2020) and Boben’s LDA topic model for Amazon TV shows (Boben, 2019).

```
import tensorflow as tf
import tensorflow_datasets as tfds
```