

**Discovering and Understanding Community
Opinions of Neighbourhoods Expressed in Question
Answering Platforms**

Marzieh Saeidi

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Department of Computer Science
University College London

May 9, 2017

تقدیم بہ مادرم و پدرم، بتول و صالح

I, Marzieh Saeidi, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Humans value the opinions of others. In recent years, people have been using social media platforms to both voice and gather opinions. Looking for relevant pieces of information through the huge amount of expressed opinions across several platforms is an overwhelming task. This is why automatically extracting information from such sources has received a great deal of attention in both academia and industry. However, little work in this field has been dedicated to the domain of city neighbourhoods. One reason is that unlike for many products and services, there are no dedicated review platforms for collecting opinions regarding the neighbourhoods.

In the absence of dedicated review sites, a great amount of expressed opinions on neighbourhoods and other domains can be found on community question answering (QA) platforms. So far, this data has not been used. This raises a question as to what the strengths and limitations of QA data are and what challenges does it bring for extracting opinion information expressed about neighbourhoods. In this thesis, we comprehensively investigate these questions, using data from Yahoo! Answers for neighbourhoods of London.

First, we investigate how well QA discussions reflect the demographic attributes of neighbourhoods present in census (e.g. *age*, *religion*, etc.). Our results show that significant, strong and meaningful correlations exist between text features from QA data and many demographic attributes. For instance, the terms “poverty”, “drug”, and “rundown” are amongst the top correlated terms with the attribute *deprivation*. We further demonstrate that text features based on Yahoo! Answers discussions can achieve a very good accuracy in predicting a wide

range of demographic attributes for neighbourhoods. These predictions outperform predictions that are made using Twitter data, a platform that has been used widely in the past for predicting many real-world attributes.

Demographics data provides objective statistics related to the population of neighbourhoods. Many attributes of interest are not reflected in those statistics. For instance, census data does not record statistics regarding whether a neighbourhood is *posh*, *quiet* or good for *nightlife*. Knowing these aspects is complementary to the demographic attributes in forming an understanding of neighbourhoods. We investigate whether text features from QA data can predict such aspects. To do this, we create a dataset of neighbourhoods labeled with these aspects. Our prediction results show that QA data can predict such aspects with a higher performance compared to Twitter data in the presence of these labels.

Predicting a single value for a characteristic of a neighbourhood cannot provide a complete picture of people's opinions. To provide a fine-grained summary, a popular approach is to extract the sentiments towards different aspects of a given entity from each expressed opinion. Aspect-based sentiment analysis has been studied extensively, but research has always utilised the text from dedicated review platforms where a user usually writes opinions on a single specified entity. In the absence of a review platform for neighbourhoods, we extend the task to process the text from QA platforms where fewer assumptions can be made and the data is noisy. We construct a human-annotated dataset based on text from Yahoo! Answers discussions with a high inter-annotator agreements of over 70%, a suitable level for this task. To address this task, we propose methods based on representations of text that are learned sequentially using recurrent neural models or representations that are defined using the traditional bag of n-grams features. Our proposed methods can achieve prediction accuracies on similar levels to the less challenging sentiment analysis tasks.

In summary, the study in this thesis demonstrates the strengths of QA data in predicting the values of real-world entities and for extracting information from opinions, specifically for the domain of city neighbourhoods.

Acknowledgements

The dissertation marks the end of a long and eventful journey. It goes without saying that there are many people that I would like to acknowledge for their support along the way.

Without equal are the tremendous sacrifices that my parents made to ensure that I had both a good education and the highest appreciation for the value of education. Throughout my life their selfless focus on family and the attainment of their children has been unrelenting. It is to them that I dedicate this dissertation.

I am indebted to Sebastian Riedel, my supervisor, for accepting the challenge of supervising me as his first UCL PhD student. He provided me with invaluable guidance on this research at every turn and found vital funding for me to continue through the last year. Without his constant encouragement, guidance and good humour the pages that follow would not have come to fruition.

I would like to thank Philip Treleaven for providing me with the opportunity to start this PhD and for providing the financial support that allows nascent ideas and raw potential to become academic research and contributions to this body of work. I would like to express my gratitude to my second supervisor Licia Capra on her advice on urban side of my research. I also like to thank Maria Liakata for guiding me through the last part of my work.

A special thanks is due to my mentor, collaborator and friend, Guillaume Bouchard. He made me believe in myself and showed me that research can be fun.

No expressions that I know in English can adequately express the depth and breadth of thanks I have to my friends Evi Pliota and Kayvan Mojarrad. Saying

that they are my best friends is merely the start. They have been a family to me when I was far from my own family in Iran. They always lent me an ear for my complaints and provided unconditional support whenever I needed it.

A special thanks to my dear friend Noemi Nava, my 'partner-in-crime' during the PhD. During these years that sometimes seemed never-ending, we have laughed together, we have cried, we have danced, we have travelled, and we have dreamed. You have made the memories of these years much more fun and for that I am grateful.

I also want to thank my PhD lab mates that were with me during this journey, Ivan Sanchez, Matko Bošnjak, Manisha Verma, and Jason Narad. Because of you, I was looking forward to going to the lab every day, even on the days when my experiments were not producing satisfactory results.

Last but not least, I would like to thank my partner Jacob Coy for his endless love, encouragement and his positive attitude throughout this journey. Without whom I would have struggled to find the inspiration and motivation needed to complete this dissertation.

Contents

1	Introduction	27
1.1	Motivation	27
1.2	Research Problem	29
1.3	Research Tasks	30
1.3.1	Opinion Aggregation	31
1.3.2	Opinion Mining	34
1.4	Scope and Assumptions	35
1.5	Contributions	37
1.6	Structure of Thesis	38
1.7	Published Papers	39
2	Literature Review	41
2.1	Social Media Data for Text Prediction	41
2.2	Urban Data Mining Using Crowd-Generated Data	45
2.3	Opinion Mining	47
2.3.1	Levels of Analysis	48
2.3.2	Data Sources	50
2.3.3	Domain	51
2.3.4	Approach	53
I	Opinion Aggregation For Neighbourhoods	59
3	Predicting Population Demographics	61
3.1	Research Questions	61

3.2	Technical Background	62
3.2.1	Regression	62
3.2.2	Linear Regression	63
3.2.3	Non-linear Regression	65
3.2.4	Gaussian Process Regression	66
3.3	Approach	68
3.3.1	Domain Entities and Concepts	68
3.3.2	Unit of Analysis	69
3.3.3	Document Representation	69
3.3.4	Correlation Analysis	73
3.3.5	Prediction	74
3.4	Dataset	77
3.4.1	Neighbourhoods	77
3.4.2	Yahoo! Answers Data	78
3.4.3	Twitter Data	80
3.4.4	Population Demographics Data	81
3.4.5	Unification of Geographical Units	82
3.5	Experiments	83
3.5.1	Scope	83
3.5.2	Demographic Attributes	84
3.5.3	Evaluation Setup	84
3.5.4	Implementation	85
3.6	Results	85
3.6.1	Correlation	85
3.6.2	Prediction	89
3.7	Limitations	97
3.7.1	Yahoo! Answers	97
3.7.2	Twitter	99
3.8	Discussion	100

4	Predicting Perceived Characteristics of Neighbourhoods	103
4.1	Research Questions	104
4.2	Technical Background	106
4.2.1	Classification	106
4.2.2	Logistic Regression	107
4.2.3	Classification without Labeled Instances	108
4.3	Approach	111
4.3.1	Domain Entities and Concepts	111
4.3.2	Unit of Analysis	111
4.3.3	Correlation Analysis	112
4.3.4	Prediction	112
4.4	Dataset	115
4.4.1	Aspects and Labeled Instances	116
4.4.2	Labeled Features	117
4.5	Experiments	117
4.5.1	Scope	117
4.5.2	Aspects	117
4.5.3	Labeled Features	117
4.5.4	Reference Distribution	118
4.5.5	Evaluation Setup	118
4.5.6	Implementation	119
4.6	Results	119
4.6.1	Correlation	119
4.6.2	Prediction	122
4.7	Discussion	130
II	Opinion Mining For Neighbourhoods	135
5	Fine-grained Opinion Mining from Social Media Data	137
5.1	Research Questions	138
5.2	Task	139

5.2.1	Existing Tasks	139
5.2.2	Targeted Aspect-Based Sentiment Analysis	140
5.2.3	Formal Definition	141
5.2.4	Evaluation Metric	143
5.3	Dataset	143
5.3.1	Preprocessing	143
5.3.2	Annotation	146
5.3.3	Procedure	148
5.3.4	SentiHood	150
5.4	Discussion	155
6	Targeted Aspect-Based Sentiment Analysis	161
6.1	Research Questions	162
6.2	Technical Background	163
6.2.1	Part-of-Speech Information	164
6.2.2	Word Embeddings	164
6.2.3	Neural Networks	166
6.2.4	Recurrent Neural Networks	167
6.2.5	Long-Short Term Memory Networks	168
6.2.6	Bidirectional LSTMs	168
6.3	Model	169
6.3.1	Training	170
6.4	Representations	170
6.4.1	Bag of N-grams Representations	171
6.4.2	Sequential Representations	176
6.5	Experiments	178
6.6	Results	182
6.6.1	Synthetic Evaluation Set	190
6.7	Data Augmentation	196
6.7.1	Automatic Augmentation	196
6.7.2	User-Assisted Augmentation	198

6.8 Discussion	205
7 Conclusion	209
7.1 Critical Evaluation	210
7.2 Future Work	211
7.2.1 Opinion Aggregation	211
7.2.2 Opinion Mining	212
7.3 Research Vision	213
Appendices	215
Appendix A Perceived Characteristics	215
A.1 Learning from Labeled Instances	215
A.2 Learning from Labeled Features	217
A.3 Learning From Labeled Instances and Features	220
A.4 Beyond London	225
Appendix B Guidelines for SentiHood Annotations	228
Appendix C Targeted Aspect-Based Sentiment Analysis	230
C.1 Synthetic Evaluation Test - Predictions	230
C.2 Synthetic Dataset with Augmented Data	236
Bibliography	240

List of Figures

2.1	Comparison of the current literature in text prediction to our research. The horizontal axis indicates the source of text that is used for prediction. The vertical axis indicates the number of target values for prediction. References to existing work are provided.	45
2.2	Comparison of the current literature in urban data mining using crowd-generated data to our work in this thesis. The horizontal axis indicates the nature of the data (text vs. non-text) that is used for prediction. The vertical axis indicates the number of target values for prediction. References to existing work are provided.	48
2.3	Comparison of the current literature in the field of sentiment analysis (SA) and our work in this thesis in opinion mining. The horizontal axis indicates the granularity level of the extracted information and the vertical axis indicates the number of entities that can be handled in a unit of text. Data sources are highlighted in bold and references to existing work are provided.	53
3.1	Input data and output values.	63
3.2	Linear regression fits a linear function (line) through the observed data.	64
3.3	A GP can fit a non-linear function (line) through the observed data and estimate the uncertainty.	66
3.4	The normalised tf-idf representation of word uni-grams of d^*	70
3.5	The normalised tf-idf of word uni-grams and word bi-grams of d^*	71
3.6	Binary representation of word uni-grams of the document d^*	71

3.7	The normalised tf-idf representation of the context around Norbury for the document d^*	72
3.8	The value of the RBF function as the distance between two locations increases.	76
3.9	Histogram of the number of QAs per areas of London.	79
3.10	Histogram of the number of sentences per areas of London.	79
3.11	Histogram of the number of tweets per areas of London.	81
3.12	Figure shows the division of London by LSOAs. It also shows the neighbourhoods of London. Centroid points of LSOAs are marked with dark dots and neighbourhoods are marked with green circles.	83
3.13	Maps of selected demographic attributes over LSOAs of London. Darker regions indicate higher values for an attribute.	91
4.1	Aspects that present geographical clustering. The points on map indicate the centre points of neighbourhoods that have been annotated positively for each given aspect.	125
4.2	Aspects that do not present clustering in geographical space. The points on map indicate the centre points of neighbourhoods that have been annotated positively for each given aspect.	126
4.3	Learning curves of performances in terms of AUC for selected aspects using Yahoo! Answers text features when we train a model using labeled instances. Red lines represents the standard deviation of the performance (\pm std)	127
4.4	Learning curves of the classification performance in terms of AUC for the selected aspects using Twitter data and the frequency score method.	131
4.5	Learning curves of the classification performance in terms of AUC for the selected aspects using Twitter data and the GE model.	131
5.1	Histogram of the number of location mentions per sentence.	145

5.2	Histogram of the length of sentences in our dataset in terms of the number of tokens..	146
5.3	Examples of annotated sentences in BRAT.	149
5.4	The number of opinions per sentence for SentiHood and SemEval datasets.	154
5.5	The number of labeled aspect categories and their sentiment distributions for SentiHood and SemEval datasets.	156
5.6	Number of unique expressions for each aspect in the SentiHood dataset.	157
5.7	Word clouds for aspect expressions of two aspects with a high and a low number of distinct expressions.	157
6.1	POS categories of words in a sentence.	164
6.2	Examples of how “dog” and “cat” can be represented using one-hot vector representation.	165
6.3	Word embeddings in a 3 dimensional space. Words that are semantically similar are closer to each other in this space.	165
6.4	The architecture of a one-layer neural network.	166
6.5	An RRN architecture.	167
6.6	Long short term memory cell.	169
6.7	Examples of sentences with multiple locations. The context of each location is marked with the same colour as the location name. . . .	171
6.8	Masked target entity representations of two locations using word uni-grams and bi-grams.	172
6.9	Left-right context representations of two locations using word uni-grams.	173
6.10	Context window representation of two locations using word uni-grams.	173
6.11	Distance-bucketed representation for location1 using word uni-grams.	174
6.12	Sum of embeddings representation of the sentence for location1 . . .	175

6.13 Sum of left-right embeddings representation of the sentence for location1	175
6.14 Pooling of left-right embeddings representation of the sentence for location1	176
6.15 Bidirectional LSTM outputs a representation for each token in the sentence. The output at the index of each location is then fed into a softmax layer to identify the sentiment class for the corresponding aspect. In this figure, LSTM is trained to identify the sentiment of the aspect <i>price</i> . The model should predict Positive for location1 and Negative for location2	177
6.16 Word embeddings are extended by two cells to represent two tokens of “target_loc” and “location1”.	180
6.17 Performances of the best sequential (SEQ) and bag of n-grams (BoNgrams) representations for each aspect and different categories of sentences.	186
6.18 The ratio of the correct predictions as the length of sentences increases. The size of each circle indicates the number of sentences in the length range.	187
6.19 Performances of SEQ and BoNgrams representations on the <i>Lexical Variation</i> synthetic category on aspect detection, sentiment detection and on average.	191
6.20 Performances of SEQ and BoNgrams representations on the <i>Negation</i> synthetic category.	192
6.21 Performances of SEQ and BoNgrams representations on the <i>Noise</i> synthetic category.	193
6.22 Performances of SEQ and BoNgrams representations on the <i>Multi-Agree</i> synthetic category.	194
6.23 Performances of SEQ and BoNgrams representations on the <i>Multi-Disagree</i> synthetic category.	195

6.24	The overall performances of SEQ and BoNgrams representations on the test set when different categories of augmented data is added for training. AUC is averaged over sentiment and aspect classification over all the aspects. Note that the y-axis starts with 0.5.	203
6.25	Performances of SEQ and BoNgrams representations on different categories of sentences in the test set when different categories of augmented data are added to the training data. Results are averaged over all aspects and over aspect and sentiment classifications. Note that the y-axis starts with 0.5.	204
7.1	Learning curves of the performance in terms of AUC for selected aspects using Twitter data when we train a model using labeled instances.	215
7.2	Learning curves of the performance in terms of AUC for selected aspects using Twitter data when we train a model using labeled instances (cont.)	216
7.3	Learning curves of the performance in terms of AUC using Yahoo! Answers data and the frequency score (Test set only).	217
7.4	Learning curves of the performance in terms of AUC using Yahoo! Answers data and the GE model (Test set only).	218
7.5	Representation of a pseudo-instance using a labeled feature.	220
7.6	Contours of performance in terms of AUC for selected aspects using Yahoo! Answers data when we train a model using labeled features and instances.	222
7.7	Contours of performance in terms of AUC for selected aspects using Twitter data when we train a model using labeled features and instances.	223
7.8	Results in terms of the average AUC (aspect and sentiment) on the synthetic set of <i>Lexical Variation</i> using SEQ and BoNgrams representations.	236

7.9	Results in terms of the average AUC (aspect and sentiment) on the synthetic set of <i>Negation</i> using SEQ and BoNgrams representations.	237
7.10	Results in terms of the average AUC (aspect and sentiment) on the synthetic set of <i>Noise</i> using SEQ and BoNgrams representations. . .	238
7.11	Results in terms of the average AUC (aspect and sentiment) on the synthetic set of <i>Multi-Agree</i> using SEQ and BoNgrams representations.	238
7.12	Results in terms of the average AUC (aspect and sentiment) on the synthetic set of <i>Multi-Disagree</i> using SEQ and BoNgrams representations.	239

List of Tables

1.1	Examples of data on Twitter about Camden . Tweets are filtered using the name of the area.	28
1.2	Examples of questions and their answers (QAs) on aspects of neighbourhoods. Names of areas are in bold.	29
1.3	An example of a QA where the discussion does not provide information on the current aspects of the neighbourhood Bow	32
1.4	Examples of sentences that have been identified as having positive and negative sentiments for the aspect safety of Camden Town . . .	35
3.1	Examples of QA threads where more than one area is discussed. . .	78
3.2	The number of significantly correlated terms (p-value < 0.001 and adjusted using Bonferroni correction) from both Yahoo! Answers and Twitter. “Y! A” is used in place of Yahoo! Answers due to the space limit. To see examples of the correlated terms, refer to Table 3.3 and 3.4.	86
3.3	Significantly correlated (p-value < 0.001) terms with the highest correlation coefficients for the selected demographic attributes using the normalised tf-idf features of Yahoo! Answers data.	87
3.4	Significantly correlated (p-value < 0.001) terms with the highest correlation coefficients for selected demographic attributes using the normalised tf-idf features of Twitter data.	88

3.5	Prediction results in terms of ρ using different feature representations of Yahoo! Answers data. Results are averaged over 10 folds and standard deviations are shown in parenthesis. All correlations are statistically significant with a p-value < 0.01 . Results having a * superscript have at least 2 folds with a p-value > 0.01	90
3.6	Prediction results on a selected set of attributes using spatial regression and Yahoo! Answers data. <i>W</i> and <i>C</i> indicate whether text features (context-binary representation) or coordinates are used as features. All correlations are statistically significant with a p-value < 0.01 . Results having a * superscript have at least 2 folds with a p-value > 0.01 . An upward arrow indicates an increase of performance in comparison with the results obtained using a linear regression model and text features presented in Table 3.5.	92
3.7	Prediction results on a wide range of attributes in terms of ρ using <i>context-binary</i> representation for Yahoo! Answers and binary representation for Twitter . Results are averaged over 10 folds and standard deviations are shown in parenthesis. All correlations are statistically significant (p-value < 0.01). Results having a * superscript have at least 2 folds with a p-value > 0.01	95
3.8	cont.	96
3.9	Number of QAs discussing neighbourhoods of Birmingham, Manchester and London.	98
3.10	Number of tweets collected for the areas of Birmingham, Manchester and London.	99
4.1	Values of the labeled feature “dance” for the label <i>Nightlife</i> and the probabilities of the predicted class under the model.	110
4.2	Aspects provided by Spareroom and the number of areas that are labeled for each aspect.	116
4.3	The selected aspects and the number of unique labeled features provided by annotators collectively for each aspect.	118

4.4	The number of significantly correlated terms (p-value < 0.01) from both Yahoo! Answers and Twitter with the selected aspects. “ <i>Y!A</i> ” is used in place of Yahoo! Answers due to the space limit. Examples of top correlated terms from both Yahoo! Answers and Twitter are provided in Table 4.5 and 4.6.	120
4.5	Top correlated terms from Yahoo! Answers with the selected aspects.	121
4.6	Top correlated terms from Twitter with the selected aspects.	121
4.7	Aspect prediction results in terms of AUC using Twitter and Yahoo! Answers data. Classifiers for aspects are trained using labeled instances. These results are based on the best performing representations of Yahoo! Answers (context PMI) and Twitter (binary) for these tasks. Two of the words with the highest coefficients that are common amongst most of the folds are displayed for each aspect.	123
4.8	Results of <i>spatial</i> prediction of aspects in terms of AUC using Twitter and Yahoo! Answers data when classifiers are trained using labeled instances. Upward arrows indicate an increase over the performance of each Yahoo! Answers and Twitter when spatial information is not incorporated into the features (Table 4.7).	124
4.9	Prediction performances using labeled features applied on data from Yahoo! Answers and Twitter using the frequency score method and the GE model. The AUC is reported on the test set. Upward arrows indicate whether the GE model improves upon the performance of the frequency score method for the given aspect and the source.	128
4.10	Examples of neighbourhoods that contain the term “quiet” in their tweets and that are labeled negative and positive for the aspect <i>Quiet</i> respectively. The word “quiet” is a labeled feature for the aspect <i>Quiet</i>	130
5.1	Example of an input sentence and the output labels.	142
5.2	Statistics of the QA dataset	143

5.3	SentiHood dataset statistics.	153
5.4	SemEval dataset statistics.	153
6.1	Statistics for train, dev and test sets.	180
6.2	Aspect and sentiment classification results using different types of representations. Results are shown both in terms of F_1 and accuracy and AUCs for aspect and sentiment classification.	182
6.3	Performances of the best sequential (SEQ) and bag of n-grams (BoNgrams) representations on each aspect. AUC scores are averaged over aspect and sentiment detection.	185
6.4	Performances of SEQ and BoNgrams representations on different categories of sentences. AUC scores are averaged over aspect and sentiment, for all the aspects.	185
6.5	Correlations ($r < 0.01$) between the length of a sentence and the prediction in terms of point-biserial correlation coefficient.	188
6.6	Examples of input sentences and their predicted labels using BoNgrams (top) and SEQ (bottom) representations.	189
7.1	Prediction performances using labeled features applied on data from Yahoo! Answers and Twitter . The AUC is reported on <i>all areas</i> . 219	
7.2	The results of the predictions using Yahoo! Answers and Twitter data when both labeled instances and features are used for training.	221
7.3	Predicting aspects using metadata from Twitter	224
7.4	Shared aspects of AirBnB and Spareroom and the number of areas that are labeled with each aspect in AirBnB dataset.	225
7.5	Prediction performance in terms of AUC, cross validated over areas of several cities around the world.	226
7.6	Prediction performance in terms of AUC for areas of a new city (zero-shot learning), averaged over all cities.	226
7.7	Prediction performance in terms of AUC for aspects of areas of <i>Tokyo</i> . 227	

7.8 Examples of labeled sentences in the *Lexical Variation* synthetic test. 230

7.9 Examples of labeled sentences in the *Negation* synthetic test. . . . 231

7.10 Examples of labeled sentences in the *Noise* synthetic test. 232

7.11 Examples of labeled sentences in the *Multi-Agree* synthetic test. . . 234

7.12 Examples of labeled sentences in the *Multi-Disagree* synthetic test. 235

Chapter 1

Introduction

1.1 Motivation

Understanding the public's opinion is very important in many domains and scenarios. The opinion of others provides a useful source of information for an individual in decision making. It helps business owners in analysing the benefits and shortcomings of their products and services to their users. Policymakers also benefit from public's opinion when designing policies.

Currently, online sources and especially social media sites are popular platforms for people to voice their opinions. Many products and services have dedicated review platforms on which users can provide their feedback and opinions. For instance, retailer sites such as Amazon provide facilities for their users to leave feedback on the products they have purchased. Directory sites such as Yelp¹ publish crowd-sourced reviews about local businesses, e.g. restaurants. Unfortunately, not all the domains and entities of interest have their own review platforms, even though knowing the publics' opinion about them can be valuable.

Take for instance neighbourhoods of cities.² Knowing about different char-

¹<https://www.yelp.co.uk/>

²There exists sites that allow users to express their opinions about different elements present in neighbourhoods (e.g. restaurants, parks, etc), such as Foursquare (<https://foursquare.com/>) and Yelp (<https://www.yelp.co>). Users do not often express their opinions about characteristics of the neighbourhoods on these sites. Spareroom (<https://www.spareroom.co.uk/>) has recently added a feature that allows users to leave feedback on the areas that they know. However, the amount of data available is currently very limited.

Table 1.1: Examples of data on Twitter about **Camden**. Tweets are filtered using the name of the area.

Reports are that the Electric Ballroom in Camden , London has been hit by the senseless riots. Let us know if you're in the area. Stay safe.
Lunch work from home jog in area Camden before #summerinlondon really comes to an end! #london
Meanwhile in Camden , Jewish Museum

acteristics of neighbourhoods is very important in many situations. Examples of these situations include an individual moving to a new city or a new neighbourhood, a business owner opening a business in a new location, and the government allocating resources to different regions or communities. In acquiring information about neighbourhoods, objective statistics are of great value and so are the opinions of others. Opinions of people provide a picture of how a neighbourhood is perceived by the public. This view does not always reflect the objective descriptions or the official statistics and can be complementary to such information. In the absence of dedicated review platforms for neighbourhoods, we need to find alternative sources of opinions.

Twitter data has been used in the past for analysing people's opinions and sentiments regarding different topics [1, 2]. However, Twitter is not usually used for expressing and discussing opinions at length. Perhaps one reason for this is that Twitter imposes a strict limit (140 characters) on the length of each microblog or tweet. Twitter is mainly used on-the-go and from a mobile device for users to share their spontaneous thoughts and observations. When it comes to neighbourhoods, Twitter is often used by users to talk about what they observe or their activities while being in a location. Table 1.1 shows examples of tweets about the London area of Camden. As we can see, it is hard to form an image of an area by reading a few tweets.

Community question answering (QA) platforms, on the other hand, are dedicated to discussions and expressions of opinions in length on many topics including neighbourhoods. Examples of QA platforms are Yahoo! Answers³ and

³<https://answers.yahoo.com/>

Table 1.2: Examples of questions and their answers (QAs) on aspects of neighbourhoods. Names of areas are in bold.

<p>Question: <i>Can anyone please suggest a safe, affordable place to live in greater london?</i></p> <p>Preferably close to victoria station and good school for my 7 year old.</p>
<p>Answer: The Beckenham - Penge area is the only place I can think of. Crystal Palace isn't safe, I used to live there and it is a disgusting, rough area. Try Beckenham!</p>
<p>Answer: best areas (going by safety, accessibility on public transport, and nice places to live) holland park, notting hill, kensington, islington, angel, barrons court. wimbledon is nice too.</p>

Quora⁴ where one can find many discussions on different aspects of neighbourhoods across many cities. A quick search on Yahoo! Answers using names of each well-known area of London such as Camden Town or Brixton returns over 100 discussion threads. Yahoo! Answers also returns over 20 results for less known areas of London such as Streatham or Golders Green. The discussions on QA platforms are not limited in length and can cover several aspects of the topic. Unlike Twitter, the discussions are not necessarily spontaneous and it may take days or months for users to respond to a question. Discussions on QA platforms are less constrained in comparison to the comments made on review platforms, but more focused and comprehensive compared to the opinions expressed on Twitter. Table 1.2 shows examples of discussion threads on Yahoo! Answers regarding some neighbourhoods of London. These discussions provide direct information regarding different neighbourhoods and their aspects such as affordability and safety.

1.2 Research Problem

Reading through over 20,000 of questions and answers for areas of interest and summarising aspects of each neighbourhood is an overwhelming task which needs a lot of time and patience. Therefore, methods are needed to somehow extract information from these discussions automatically. This is not a new problem. As e-commerce and crowd-sourced review sites have become more and

⁴<https://www.quora.com/>

more popular, the number of reviews that a product or a business receives has grown rapidly. For example, for a popular product such as iPhone6, the number of reviews can be in hundreds or even thousands. This makes it difficult for customers to quickly assess the overall user opinion about a product in order to make a purchase decision. Therefore, extracting useful information from customer reviews has created a lot of interest in academia and in industry, both for interesting challenges it brings to the field of Natural Language Processing (NLP) and also for its value in commercial applications. However, most of the focus so far has been on the analysis of customer reviews that are expressed on review-specific platforms [3, 4, 5].

The text generated on QA platforms is less constrained and more generic in comparison with the extensively studied review data where more assumptions can be made. Extracting information from opinions expressed on QA platforms can present new challenges that have not been investigated in the past. Most of the existing work that utilises QA data aims to improve the way QA platforms operate [6, 7, 8, 9]. The text from QA discussions has not been used in the past for extracting opinion information.

In this thesis, we raise the question as to why the opinions expressed on QA platforms have not yet got much attention. To answer this question, we investigate the strengths and the limitations of text coming from QA platforms and the challenges it brings for extracting information from opinions expressed for neighbourhoods of a city. We use the data from the Yahoo! Answers question answering platform about neighbourhoods of London.

1.3 Research Tasks

The objective of this thesis is to investigate the challenges that using QA data brings for extracting opinion information about neighbourhoods of cities. In particular, we are interested in discovering people's opinions about characteristics of neighbourhoods such as their safety, price, trendiness or deprivation.

We investigate extracting opinion information in two levels of granularity.

On a coarser level, we predict an overall value for each characteristic of a neighbourhood. For instance, we predict whether a neighbourhood is known to be safe, trendy or dominated by people from a specific religion or ethnic background. We refer to this as opinion aggregation in this thesis. Opinion aggregation means aggregating all the opinions into a single value. This is to validate whether the discussions on QA platforms reflect the true characteristics of neighbourhoods. On a finer level, we identify all the different sentiments that have been expressed for a neighbourhood about a characteristic. For example, we identify all the negative and positive opinions or statements that have been made about the safety of a neighbourhood. This is referred to as fine-grained opinion mining in the literature. Providing fine-grained opinion information can help users to form a better understanding of people's opinions about a neighbourhood.

1.3.1 Opinion Aggregation

The aim of the opinion aggregation task, in this thesis, is to validate whether the discussions on QA platforms reflect the true characteristics of neighbourhoods. Moreover, it is useful to know values of different characteristics of neighbourhoods, especially those that are not available in the official statistics. For this task, we study whether we can predict a value indicating the extent of which a neighbourhood is known for having a characteristic from the QA data. Examples of such characteristics are *safety*, *trendiness*, and *quietness*. To compare the strengths and limitations of QA data in opinion aggregation for neighbourhoods with other social media sources, we also apply our methods to the data from Twitter.

QA discussions can contain opinions on many characteristics of neighbourhoods as we have seen in Table 1.2. However, we cannot expect for all the characteristics of interest to be discussed for all the neighbourhoods on a QA platform. Furthermore, not every discussion on QA platforms about neighbourhoods provide useful or relevant information. An example is provided in Table 1.3.

To show that the collective opinions expressed for neighbourhoods on QA

Table 1.3: An example of a QA where the discussion does not provide information on the current aspects of the neighbourhood **Bow**.

<p>Question: <i>Bow bells??</i> what does it mean when someone says they were born under the sounds of Bow bells??</p>
<p>Answer: they are the bells of st mary le bow church in the city of london. everyone who was born within the range of these bells is said to be a real londoner, other people aren't ...</p>

platforms reflect their true characteristics, we first investigate whether the population demographic attributes of neighbourhoods available in the census data can be predicted using the discussions on QA platforms. We then aim to predict characteristics of interest that are not available in the official statistics but are useful and essential in forming an opinion about a neighbourhood. We call this group of characteristics the “perceived characteristics” of neighbourhoods. These two sub-tasks are explained further below.

1.3.1.1 Predicting Population Demographics

Population demographic attributes are objective statistics available through the census records. The values for the demographic attributes can be measured numerically, often by a population count. In this thesis, we refer to the population demographic attributes as “*attributes*”, in short. Examples of such attributes are the percentage population of *Muslim* living in an area or the *Unemployment Rate*. Census data also includes attributes that are not calculated by population count but can still be measured numerically. An example is the average house prices in an area.

Many QA threads can be found that *explicitly* discuss demographic attributes such as religion or ethnic background. The following is an example taken from Yahoo! Answers where a demographic attribute (i.e. the Jewish religion) has been mentioned. The name of the neighbourhood is in bold and the demographic attribute is underlined.

“... **Golders Green** is pretty good. its safe (its the most Jewish of London sub-

urbs, and they are generally a quiet and peaceful bunch) and it has good transport connections ...”

It is not always the case for all the demographic attributes to be discussed explicitly in QA threads. However, the language people use when talking about specific neighbourhoods and the choice of the vocabulary may still be indicative of the attributes of those neighbourhoods. We define our hypothesis for this task as follows:

Hypothesis 1 *The language used in QA discussions about neighbourhoods reflects the demographic attributes of their population taken from census records.*

1.3.1.2 Predicting Perceived Characteristics of Neighbourhoods

The values for many of the perceived characteristics of neighbourhoods cannot be calculated through a count measure. These characteristics are related to how people perceive an area. We refer to these characteristics as “*aspects*” in this thesis. Examples of such aspects are how *posh*, *trendy* or *quiet* an area is. Some of the aspects of interest are related to the activities of people in a neighbourhood such as whether a neighbourhood is good for *nightlife* or *dining*.

Some of the aspects in this group can be related to the demographic attributes, but the value of such attributes in the objective statistics may not reflect people’s perception. Take *safety* as an example: even though safety is measured in the official statistics by the number of crimes reported in a neighbourhood, those numbers do not always reflect the view that people have of the safety of a neighbourhood. For example, it is common for people to feel uncomfortable to walk in a quiet residential area, but feel safe to be in a crowded place. This is contrary to what crime statistics suggest: there are more crimes in crowded neighbourhoods than in residential ones.⁵

There are many QA threads on Yahoo! Answers platform that discuss these

⁵<http://maps.met.police.uk/>

aspects of neighbourhoods. The example below shows discussions on aspects such as *nightlife*, *shopping*, *safety*, and *quietness*. The names of neighbourhoods are in bold and aspect related terms are underlined.

*“**Islington** is great for shopping and lots of cafes etc ... **Shoreditch/Hoxton** have become the more trendy going out districts in the last 10 years ... I think **Willesden Green** is quiet and fairly safe ...”*

While not all the aspects of interest may have been discussed explicitly in QA discussions, the language people use in the discussions may still reflect the values of these aspects for neighbourhoods. Therefore, we define our hypothesis for this task as follows:

Hypothesis 2 *The language used in QA discussions about neighbourhoods reflects their perceived characteristics.*

1.3.2 Opinion Mining

Aggregating opinions into a single value cannot provide a complete picture of people’s view about a neighbourhood. To provide a fine-grained summary, a very popular approach in opinion mining is to extract sentiments expressed towards different aspects of a given entity in a small unit of text such as a sentence. This means that instead of providing users with a single value indicating whether a neighbourhood is known for having a characteristic or not, we can provide the users with all the positive and negative opinions that people have expressed for each characteristic. This is of great value since it eliminates the need for a user to read through hundreds of QA threads and to categorise (e.g. positive, negative, etc.) the information of interest. Table 1.4 shows examples of the desired output for this task, e.g. positive and negative opinions about the safety of Camden Town extracted from several QA threads.

Table 1.4: Examples of sentences that have been identified as having positive and negative sentiments for the aspect safety of **Camden Town**.

Camden Town: Safety
✓Regarding safety, Camden Town is usually ok!
✓I have lived in Camden Town , never had a problem!
✗Parts of Camden Town are run-down, try to avoid them if possible.

This task is similar to the existing task of aspect-based sentiment analysis. In aspect-based sentiment analysis task, we identify the sentiment that is expressed towards different aspects of an entity. Research on this task, so far, has only utilised the text from dedicated review platforms. Fine-grained opinion mining from social media platforms has not been studied. Unlike the review platforms that are used for reviewing entities, social media platforms such as QAs are used for general discussions. Therefore, the text from these platforms is more prone to noise and is less constrained. However, this text is closer to the natural way that people express themselves. For instance, unlike review platforms where users usually talk about one entity at a time, in QA platforms users can talk about several entities in the same unit of text.⁶

Therefore, mining fine-grained opinion information from QA discussions can raise new challenges when creating a dataset, and also when defining models for extracting fine-grained opinion information for neighbourhoods. Despite the fact that the text from QA discussions is noisy and is not written as reviews, we hypothesise that it can be used for extracting fine-grained opinion information:

***Hypothesis 3** Discussions on QA platforms about neighbourhoods can be used for extracting fine-grained opinion information for neighbourhoods.*

1.4 Scope and Assumptions

In this section, we describe the scope of the thesis and the assumptions that we have made.

⁶Examples of such discussions are provided in Table 1.2.

- In our investigations, we focus mainly on the neighbourhoods of *London*. Many discussions regarding neighbourhoods of London can be found on QA platforms. This can be because London is a big cosmopolitan city and the destination for many tourists and immigrants. The hypotheses that we propose in this thesis may not hold for other cities if not enough data is available for their neighbourhoods.
- In this thesis, we use the discussions from Yahoo! Answers QA platform for proving our hypotheses. The Yahoo! Answers platform was created over a decade ago⁷ and contains many existing discussions about neighbourhoods of London and many other cities. Other popular QA platforms such as Quora have emerged more recently.⁸ Therefore, they may have a fewer number of discussions available for neighbourhoods of cities. While the text coming from different QA platforms might be very similar in nature (less constrained and more generic compared to review data), we have not studied whether our hypotheses hold for other QA platforms.
- This thesis focuses on using *language* for predicting characteristics of neighbourhoods. Metadata from Twitter and Yahoo! Answers such as the number of users, the number of tweets, the number of votes on an answer, the credibility of the QA user, etc. can be incorporated into the prediction models. However, this is out of the scope of this thesis.
- For opinion aggregation, we apply our methods to Yahoo! Answers as well as Twitter. This is because Twitter data has been used in the past for predicting values of real-world entities including neighbourhoods. Therefore, it can provide a reasonable baseline for predictions made using Yahoo! Answers data. For opinion mining, however, we only investigate extracting fine-grained information from *Yahoo! Answers*. Fine-grained opinion mining has not been applied to the data from Twitter in the past and is out of the scope of this thesis.

⁷Yahoo! Answers was launched in 2005.

⁸Quora was launched in 2010.

1.5 Contributions

In summary, in this thesis, we show that in the absence of centralised platforms for opinions, alternative social media sources can be used for extracting useful and necessary information. We specifically demonstrate this for the community question answering platform of Yahoo! Answers in the domain of neighbourhoods. A detailed list of contributions is provided below. These contributions are divided into two categories:

Opinion Aggregation

- We show that the language used on Yahoo! Answers discussions about neighbourhoods reflects the demographic attributes of neighbourhoods. We do this by demonstrating that there are significant, strong and meaningful correlations between many terms in those discussions and many attributes from the demographics data. Further, we show that these attributes can be predicted using the term frequency features of Yahoo! Answers text with an average Pearson correlation coefficient of 0.62, a 4% increase over Twitter.
- We demonstrate that the language used on Yahoo! Answers discussions about neighbourhoods also reflects the aspects of neighbourhoods. We show that such aspects can be predicted using the term frequency features of Yahoo! Answers discussions with an average AUC of 74%. This is 4% higher than the performance achieved using the term frequency features of Twitter data.

Opinion Mining

- We create SentiHood, a benchmark dataset for fine-grained opinion mining for neighbourhoods from Yahoo! Answers discussions, achieving inter-annotator agreements of over 70%, a suitable level for this task. This is the first time that the generic and less constrained text from a social media platform has been used for creating a dataset for fine-grained opinion mining.

- We introduce a new task in the field of sentiment analysis to address extracting fine-grained opinion information from the less constrained text of social media. We call this task Targeted Aspect-Based Sentiment Analysis.
- We propose strong methods based on representations of text that are learned sequentially using recurrent neural models or representations that are defined using the traditional bag of n-grams features. Our results show that, overall, discussions on Yahoo! Answers about neighbourhoods can be used for fine-grained opinion mining. This is despite the fact that these discussions were not written as reviews for neighbourhoods. Our proposed methods for the task of targeted aspect-based sentiment analysis on SentiHood dataset can achieve performances of over 90% in AUC in extracting the correct aspects and sentiments expressed for neighbourhoods.

1.6 Structure of Thesis

In this section, we provide an overview of the structure of the thesis.

In Chapter 2, we investigate the current literature related to this work. We first look at the literature that uses crowdsourced and social media data for making predictions about real-world entities. We also discuss the research work that uses social media data for urban data mining. We finally discuss the current tasks and the literature in the field of opinion mining and critically challenge their abilities to process the text from less constrained platforms such as QA.

Experiments in this thesis are divided into two parts. Each part investigates one of the tasks that we proposed in Section 1.3. In Part I, we investigate predicting characteristics of neighbourhoods through opinion aggregation. This is done both for the demographic attributes and the perceived characteristics in the following two chapters.

In Chapter 3, we discuss in details the creation of datasets from both Yahoo! Answers and Twitter platforms. We also investigate whether strong and meaningful correlations exists between text features from Yahoo! Answers data and the demographic attributes taken from census statistics. Text features are also used

to evaluate how well these attributes can be predicted. Results of our models on text from both QA and Twitter are compared.

In Chapter 4, we first explain the process of creating a dataset for perceived characteristics of neighbourhoods. We then investigate how well text features from Yahoo! Answers data can predict such aspects and how the predictions compare to those made using the data from Twitter.

Part II is dedicated to fine-grained opinion mining for neighbourhoods using QA data. This part is divided into the two following chapters:

In Chapter 5, we investigate the suitability of the existing tasks in the field of fine-grained opinion mining for processing the text based on QA discussions without solving the task. We propose a new task by combining two existing tasks in the field. Further, we describe creating a human-annotated dataset from the text obtained from Yahoo! Answers discussions about neighbourhoods of London for fine-grained opinion mining. We call this dataset SentiHood.

In Chapter 6, we attempt to solve the task proposed in Chapter 6 on SentiHood dataset. We investigate the use of sequential and bag of n-grams representations for extracting fine-grained opinion information for neighbourhoods. We apply our methods to the data available in SentiHood dataset and analyse the results in details.

Finally, in Chapter 7, we evaluate our work and our approach. We also provide suggestions for the future work.

1.7 Published Papers

Information regarding the publications that are based on this thesis is listed below together with the chapter that they correspond to.

Chapters	Paper
3	<i>Lower Dimensional Representations of City Neighbourhoods</i> M Saeidi, S Riedel, L Capra Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence
5 and 6	<i>SentiHood: Targeted Aspect Based Sentiment Analysis Dataset for Urban Neighbourhoods</i> M Saeidi, G Bouchard, M Liakata, S Riedel COLING 2016

Chapter 2

Literature Review

In this thesis, we extract and summarise opinions expressed in community question answering platforms with regards to neighbourhoods of cities. We do this in different levels of granularity. On a higher level, we predict values for attributes and aspects of neighbourhoods. We refer to this task as “opinion aggregation”. A value indicates the extent of which a neighbourhood is known for having an attribute or an aspect. This value can be continuous or binary. Opinion aggregation for neighbourhoods is related to the field of text prediction which is discussed in Section 2.1. Since we are interested in predicting attributes and aspects of neighbourhoods using QA data, we compare our work to the existing research that facilitates crowd-generated data for characterising attributes of urban areas. This is discussed in Section 2.2. On a lower level of granularity, we extract fine-grained information from opinions in QA text. For this, we place our work within the existing field of opinion mining and sentiment analysis. We look at the current tasks, datasets and approaches that exist in this field and discuss their limitations for addressing fine-grained information extraction from QA data.

2.1 Social Media Data for Text Prediction

Text prediction refers to the task of predicting a value using features that represent the meaning of the text. Text regression is a specific case of text prediction in which we predict a numeric value. Data from online sources, especially data generated on social media platforms has been used in the past to predict values

of real-world entities in many domains [10, 11, 12].

Research indicates that many attributes of the real-world can be predicted using different features from text descriptions. This is done by converting the text into numerical regressors. These regressors are features obtained using methods from computational linguistics.

Expert Generated Text

Financial reports are used for predicting the volatility of stock returns [10]. It is shown that the text-only method comes within 5% of the error of a strong historical data baseline. Research shows that house prices can also be predicted using descriptions of houses that are on sale from online estate agent websites [13].

Crowd Generated Text

Many recent studies in this field are inspired by the availability of crowd generated text (e.g. review platforms) or text-based social media platforms (e.g. blogs, Twitter). For instance, reviews from movie review sites are used to predict a movie's opening weekend revenue [11]. It is shown that text features based on movie reviews can improve or even replace features that are based on strong metadata baselines. Metadata includes information such as the genre of the movie, running time, release date, the presence of particular actors or actresses in the cast, etc.

Twitter data, in particular, has been used widely as a social media source to make predictions in many domains. For example, box-office revenues have been predicted using data from Twitter [14]. Predictions using text features from Twitter data are shown to outperform other market-base predictors (i.e. Hollywood Stock Exchange). Also, content analysis performed on tweets having a reference to a political party or a politician shows that the number of mentions of each political party can reflect the election results [12]. Further, it is shown that correlations exist between mood states of collective tweets and the value of Dow Jones Industrial Average (DJIA) [15]. Such sentiment measures can predict changes in DJIA closing values with an accuracy of 87%. Tweets of users have also been studied to find patterns and regularities in Twitter such as bots [16].

Data generated on QA platforms have not been used in the past for predicting attributes of real-world entities. Most research work that utilises QA data aims to increase the performance of QA platforms in analysing question quality [17], question popularity [6], predicting the best answers [9, 8] or the best responder [7].

Domain

Text from online resources and especially social media platforms has been used to make predictions in many domains. Examples of these domains are financial markets [10], housing market [13], movie industry [14], election [12], etc.

Predicting demographic attributes of individual users using their language on social media platforms especially Twitter has been the focus of many research work. For example, text from blogs, telephone conversations, and online forum posts are utilised for predicting author's age [18]. Results show that the age of authors can be predicted where the correlations between the predicted and the observed demographic attributes (Pearson's correlation ρ) can reach almost 0.7. In other work, sociolinguistic associations using geo-tagged Twitter data have been discovered [19] and the results indicate that demographic information such as first language, race, and ethnicity can be predicted using text features of tweets. The predicted values of such demographic information can reach correlations of up to 0.3 with their real values. Other research focuses on user income [20], showing that text features from users' tweets can achieve a good predictive accuracy.

Finally, text from social media platforms has been used to predict characteristics of neighbourhoods. Specifically, it is shown that the collective sentiment of users' tweets can be used to predict the well-being of the communities that users belong to in the real world [21, 22]. In these works, well-being is defined as the deprivation index of a neighbourhood. Deprivation index or more precisely index of multiple deprivation (IMD) is a UK government measure for deprivation in English local councils.¹

¹IMD takes into account the income, employment, health deprivation, education, barriers to housing, living environment, and crime levels of an area or its population.

Features and Methods

Text features that are used as regressors vary from n-grams [10, 11], count of word class types [18, 12] (e.g. percentage of words longer than 6 letters) to more syntactic features such as POS (Part of Speech) tags [10, 11] or dependency relations [11]. An n-gram is a sequence of n items in a text. These items are usually words but other items such as letters or syllables can be also used. POS tags are categories that define the grammatical properties of words. Words with the same POS category have similar syntactic behaviour. Using these features for representations can lead to sparse vectors of high dimensions. In some cases, instead of using sparse representations, dense feature representations are used. These representations are obtained using methods such as LSA [13], PCA, SVD or neural networks [20]. In other cases, especially when the sentiment is utilised, different hand-engineered features are used for quantifying the sentiment in text [21, 22]. Suitability of different representations depends heavily on the nature of the task and the data.

To make predictions using text representations, the majority of the existing work use a linear regression model [19, 10, 18, 15]. Gaussian process regression has also been used for text prediction [20, 23]. Gaussian processes are capable of capturing the non-linear relationships between the features and target values.

Our Work

Figure 2.1 shows the major differences between the current work in text prediction and the work carried in this thesis. Research in text prediction has utilised text from different sources. Expert generated text such as financial reports or house descriptions are generated by the experts in the field and serve a specific purpose. The language used in these sources are formal and the text is not noisy. Here, noise refers to the information irrelevant to the topic of interest. Crowd-generated text refers to the text that is generated by users on constrained platforms such as review platforms or generic social media platforms such as Twitter, QAs and blogs. On these platforms, users tend to adopt an informal language and the text is often noisy. The language people use on Twitter is very informal and

can be different from other social media platforms. For example, many words are shortened by using abbreviations or by removing vowels. One reason for this is the length limit that is applied on each tweet. While sources such as Twitter have received a lot of attention in the existing literature for predicting real-world values, sources such as QA platforms have not been utilised.

As we can see in the figure, existing studies focus on making predictions for one or very few set of real-world values in their respective domains. In this thesis, we predict a wide range of attributes and aspects of neighbourhoods using text from QA discussions.

Comparison with Literature in Text Prediction

		Source			
		Expert Generated	Crowd Generated Data		
Number of Target Variables				Twitter	Other
		Wide Range			
One or Few	Stock Market (Financial Reports [10]) House Prices (Estate Agent Site [13])	Author's Language, Race, Income (Twitter [19,20]) Stock Market (Twitter [15]) Politics (Twitter [12]) Movie Revenue (Twitter [14]) Neighbourhood (Twitter [21,22])	Author's Age (Blogs [18]) Movie Revenue (Review [11])		

Figure 2.1: Comparison of the current literature in text prediction to our research. The horizontal axis indicates the source of text that is used for prediction. The vertical axis indicates the number of target values for prediction. References to existing work are provided.

We use similar feature representations and methods to the existing work in literature for the prediction tasks.

2.2 Urban Data Mining Using Crowd-Generated Data

Many works in urban data mining are inspired by the availability of location-based social networks. These works have tried to characterise geographical regions from various perspectives using social media data. Some of the works use

information on activities of users in these platforms [24, 25] while others take advantage of the textual information [26, 21] that some of these platforms offer. We will describe these in more details below.

Data Sources

Examples of crowd-generated data used in urban data mining are telecommunication data, transport flow, and Twitter data. Telecommunication data together with geo-tagged venues from Foursquare are combined in [25] to predict candidate activities such as nightlife or shopping for a user in an area. Activities of users on Twitter such as the number of tweets, the number of users, and the movement of the crowd has been used to successfully categorise urban areas [24]. These categories include “bedroom”, “office”, “nightlife” and “multi-functional”.

Deprivation Index and Non-Textual Data

One aspect of urban area life that has been the focus of many research work in urban data mining is finding correlations between different sources of data and the well-being index, i.e. IMD, of neighbourhoods across a city or a country [27, 28, 21, 29]. This is because identifying deprived areas is very important for the government and policy makers in order to allocate resources. However, collecting this data is an expensive process for the government. Therefore, the deprivation data cannot be collected frequently. This has inspired many studies to use frequently updated and easy to obtain social media and crowd-sourced data to estimate a proxy to deprivation. For instance, high correlations ($|\rho| > 0.7$) are discovered between features based on aggregated call details of mobile phone users and the deprivation index in Ivory Coast [27]. The physical elements that are present in an area (e.g. the number of car washes or bus stops) are used to predict their deprivation [29] achieving an F-measures as high as 0.74 for some categories of deprivation. These physical elements are obtained from crowd-sourced platforms of OpenStreetMap and Foursquare. Also, the flow of public transport data has been used to find correlations ($\rho = 0.21$) with the deprivation of areas in London [28]. Results show that the levels of transport flow of highly deprived areas are much lower than that of less deprived areas.

Deprivation Index and Text

Features taken from text-based social media platforms such as Twitter are also used to predict the deprivation index of areas. Research shows that there are links between the deprivation of areas and the use of vocabulary or topics of the users in that area on Twitter [26]. For example, certain topics such as celebrity gossips or environmental issues are correlated, positively or negatively, with the deprivation index of areas. Moreover, as discussed previously, correlations up to 0.35 are discovered between the sentiment expressed in tweets of users in a community and the deprivation of that community [21]. To obtain a quantitative measurement for the collective sentiment of users in an area, the sentiment of each tweet is estimated and aggregated over all the users detected for an area. Moreover, using expressive linguistic features such as POS tags are shown to result in a more precise computation of IMD in [22].

Our Work

Figure 2.2 shows the main differences between the existing work in urban data mining and our work in this thesis. Social media data and inexpensive crowd-generated data has been used in many studies to find links to the existing phenomena in urban areas. Most existing research focuses on predicting one single attribute, mainly IMD. While some work has used text-based features to make predictions or to find correlations, the majority of work rely on user activities and other resources such as telecommunication data. Moreover, the text from QA platforms has not been used for prediction in this field. In this thesis, we investigate predicting a wide range of attributes and aspects for neighbourhoods using the text from opinions expressed on the QA platform of Yahoo! Answers about neighbourhoods.

2.3 Opinion Mining

The aim of opinion mining, which is also referred to as sentiment analysis (SA) in literature, is to analyse people's opinions, sentiments, and emotions towards entities and their attributes. An entity can be a product or a service. The early

Comparison with Literature in Urban Data Mining

		Data Source Type	
		Non-text	Text
Number of Target Variables	Wide Range		Neighbourhood Attribute Prediction (QA)
	One or Few	IMD (Cellular[27], Transport[28], POI [29]) Profile Areas (Twitter [24]) Predict Activities (Telecom/Foursquare [25])	IMD (Twitter[21, 22, 26])

Figure 2.2: Comparison of the current literature in urban data mining using crowd-generated data to our work in this thesis. The horizontal axis indicates the nature of the data (text vs. non-text) that is used for prediction. The vertical axis indicates the number of target values for prediction. References to existing work are provided.

research on opinions and sentiment [30, 31, 32, 33, 34] did not use the terms sentiment analysis or opinion mining. The term *sentiment analysis* was first introduced in [35], and the term *opinion mining* first appeared in [36], both in 2003.

Since then, the field has got a lot of attention from both research and industry. The industry interest stems from the fact that sentiment analysis has applications in many domains. These domains include a wide range of products (mobile phones, cameras, books, etc) and services (restaurants, cafes, schools, hotels, etc). This field also created a lot of opportunities for research to solve the challenges that it brings to the field of language processing. Furthermore, the availability of a huge volume of opinionated data on social media platforms has accelerated the work and research in this field.

2.3.1 Levels of Analysis

Opinion mining is mainly performed in four levels: document level, sentence level, entity level, and aspect level. Document level sentiment analysis [32, 34] investigates whether a whole document expresses a positive or a negative senti-

ment. In this level of analysis, we assume that only sentiment towards a single entity is expressed in a document. This analysis cannot be used in documents where multiple entities are discussed.

In sentence level sentiment analysis [4], we identify the overall sentiment of a sentence. Here, we make the same assumption as above. This assumption is limiting. Oftentimes, multiple entities or different aspects of one or more entities can be discussed in a single sentence. To be able to identify the sentiments towards different entities or towards different aspects of a single entity, two recent tasks have been introduced: *aspect-based* sentiment analysis and *target-dependent* sentiment analysis.

In aspect-based sentiment analysis, which is also referred to as fine-grained opinion mining, different sentiments towards different aspects of a *single* entity can be expressed in a sentence. Instead of identifying the overall sentiment towards the entity, we can now extract sentiments towards its different aspects [37, 38, 39]. For instance, consider the following sentence: “The waiter was very rude but the pizza was delicious”. The entity is a restaurant which is implicit and *negative* and *positive* sentiments are expressed towards two of its aspects, *service* and *food*, respectively.

In target-dependent (a.k.a. targeted) sentiment analysis, we investigate the classification of sentiments towards different target entities that are present in a sentence [40, 1, 41, 2, 42]. This task assumes only an *overall* sentiment for each entity. For instance, we can process the sentence “Despite having a bad day, I think Taylor Swift’s new song is amazing.” and identify that the sentiment towards Taylor Swift is positive. Even though this task allows for the presence of more than one entity, so far, the existing corpora for this task have contained sentiment labels towards a single entity in each sentence.

Limitations

Existing tasks in the field of sentiment analysis make specific assumptions about the given unit of text which is limiting. For example, in the task of aspect-based sentiment analysis, it is assumed that opinions towards aspects of one single en-

tity are expressed in a sentence. In targeted sentiment analysis, on the other hand, we assume only an overall sentiment for one or more target entities. There exists many scenarios in which sentiments towards different aspects of several entities are discussed in the same unit of text. The discussions on QA platforms regarding neighbourhoods are examples of such scenarios. To address these limitations, we propose a new task that combines two existing tasks of *targeted* and *aspect-based* sentiment analysis. We call this task *Targeted Aspect-Based Sentiment Analysis*. Not only in this task we allow for extracting sentiments towards several entities (similar to targeted sentiment analysis task), sentiments towards different aspects of different entities can be identified (similar to aspect-based sentiment analysis task). This is particularly useful when the entity of interest does not have a dedicated review platform where one can assume that a user expresses opinions about one entity in a single review (and consequently in all the sentences in the review snippet).

2.3.2 Data Sources

In this section, we describe the data sources that have been used for different tasks in the field of sentiment analysis.

Sentiment Analysis

Many web sources and social media platforms have been used to create datasets for the task of sentiment analysis. Some examples are blogs [43, 44, 45], news [46, 47], suicide notes [48] and Twitter [1, 41].

Aspect-Based Sentiment Analysis

For the task of aspect-based sentiment analysis, current research has mainly focused on text from dedicated review sites. To the best of our knowledge, no current dataset in this task is created based on text from generic social media platforms such as blogs or QA. For example, the dataset based on reviews of five electronic products taken from merchandise sites is used in [4]. Beer and camera reviews are used in [3] which are obtained from reviews on a beer reviewing site²

²<https://www.beeradocate.com>

and Amazon respectively. Reviews of restaurants taken from city guide website³ is another example of work on dedicated review data [5].

The popularity of the data from dedicated review sites for this task is perhaps because review platforms impose implicit constraints. Customers use these platforms to write a review or to search through existing reviews. Each entity has its own page in which users can write their comments. This means that users often express opinions about a single entity at a time. There are no such constraints on QA platforms. Each user can ask a question about one or more entities and get responses with regards to one, some or all the entities in question. Also, people often tend to react to the answers of other respondents or tell stories that are not completely relevant to the main question. Therefore, the data can be noisy and the text spans in which opinions for different entities are expressed cannot always be separated per entity. The current task of aspect-based sentiment analysis does not handle several entities in the same unit of analysis.

Target-Dependent Sentiment Analysis

Twitter data has been used for the task of target-dependent sentiment analysis. Other sources of social media data have not been yet utilised for this task.

Our Work

In this thesis, we extract information from opinions expressed for neighbourhoods using text from QA discussions. One reason for this choice is the lack of availability of a dedicated review platform for neighbourhoods. QA platforms contain valuable and informative discussions and opinions on different aspects of many neighbourhoods across different cities in the world. Social media data and specifically data from QA platforms have not been used in the past for fine-grained opinion mining (a.k.a. aspect-based sentiment analysis). The use of such data introduces new challenges that we will investigate and discuss in this thesis.

2.3.3 Domain

Research on sentiment analysis has been applied and studied in many domains. Here, we provide a brief overview of the work on some of these domains.

³<http://www.citysearch.com/guide/newyork-ny-metro>

Finance

One of the first domains that sentiment analysis has been applied to is *Finance* [30]. More work has been done since then in sentiment analysis for stock price predictions and in defining trading strategies [15, 49, 50, 46]. The relation between the NFL betting line and public opinions in blogs and Twitter is also studied [45].

Hospitality

The *Hotel* [37, 51] and *restaurant* [37, 52, 5, 51, 53] domains have received a lot of attention in the field of sentiment analysis. The benchmark dataset in the SemEval annual shared task of aspect-based sentiment analysis is based on restaurant reviews [54].

Products

Sentiment analysis has been applied extensively for summarising the sentiment of the public towards many different *products*. These products vary in range. Examples are: camera [3, 55, 56, 4], beer [3], laptop [54], MP3 players [55, 4, 51], DVD players [4], cellular phones [31, 4] and PDAs (Personal Digital Assistants) [31]. Moreover, the sentiment of *movie* reviews from forums [43, 44, 32, 57] or Twitter [14] are investigated in several studies.

Our Work

Fine-grained opinion mining has not been applied to the domain of *neighbourhoods*.⁴ This can be due to the lack of review specific data for this domain. In this thesis, we investigate fine-grained opinion mining for neighbourhoods from QA data.

Figure 2.3 shows the differences between the existing work in the field of sentiment analysis and our work in this thesis. These differences are in terms of the number of entities, the granularity of aspect information and the source of data. For simplicity, we do not include “Domain” and “Approach” dimensions in this figure. We describe the approaches that have been taken in the literature

⁴The collective sentiment expressed in tweets of users has been used for predicting deprivation index of neighbourhoods [26]. This is slightly different from analysing sentiment of people towards different neighbourhoods.

for addressing the tasks in this field and our proposed approach in the following section.

Comparison with Literature in Opinion Mining

		Information Granularity	
		Overall	Fine-grained
# Target Entities	Multiple	Targeted SA (Twitter[1,2,40,41,42])	Targeted Aspect-Based SA Neighbourhoods(QA)
	Single	SA (Blogs[43,44,45]) (News [46,47])	Aspect Based SA (Review[5,34,35,36,37,38])

Figure 2.3: Comparison of the current literature in the field of sentiment analysis (SA) and our work in this thesis in opinion mining. The horizontal axis indicates the granularity level of the extracted information and the vertical axis indicates the number of entities that can be handled in a unit of text. Data sources are highlighted in bold and references to existing work are provided.

2.3.4 Approach

From an NLP perspective, even though solving the sentiment analysis problem involves solving many problems of natural language like co-reference resolution and negation, we often do not need to thoroughly understand the context to determine the sentiment [58]. Approaches that address the general task of sentiment analysis can be divided into two general categories: unsupervised and supervised. These approaches are explained further below.

Unsupervised Methods for Sentiment Analysis

Unsupervised methods for sentiment analysis are mainly lexicon-based. Lexicon-based methods [34, 59, 60, 4] rely on sentiment related words that can be obtained using different approaches. Sentiment lexicons are words that are indicative of sentiment, either positive or negative. Although sentiment words and phrases are important for sentiment analysis, relying only on them is far from sufficient. A positive or negative sentiment word can have opposite ori-

entations in different domains of analysis. Additionally, a sentence containing sentiment words may not express any sentiment and many sentences without sentiment words can bear sentiment.

Supervised Methods for Sentiment Analysis

Feature-based supervised methods such as maximum entropy classification and support vector machines have been used for the classification of the sentiment [32]. The performance of these models depend on the features that are used to represent text. Even though term frequency features such as tf-idf have traditionally been important in many NLP tasks, in sentiment analysis task a better performance can be achieved using presence rather than frequency [32]. Higher order n-grams are shown not to be more effective than uni-grams in sentiment detection tasks [32]. However, in some domains, product-review sentiment classification can benefit from bi-grams and tri-grams [36]. Part-of-speech (POS) information is commonly used in sentiment analysis and opinion mining. The reason for this is that POS tagging can be considered to be a simple form of word sense disambiguation [61]. Incorporating syntactic relations has also been investigated for sentiment classification. Such linguistic features seems particularly relevant with short pieces of text [62]. Parsing the text can also help in modelling negation, intensifiers, and diminishers [63].

Recently, deep learning models have been applied for identification of the sentiment. Such models do not depend on engineering domain or task-specific features. For instance, recursive neural networks have been used to hierarchically compose word embeddings based on syntactic parse trees. These vectors are then used to identify the sentiments of the phrases and sentences [64]. Bi-directional LSTMs have also been used for sentiment classification [65], outperforming recursive neural networks that are based on syntactic parse trees.

In the following sections, we look at approaches that are used for the tasks of target-dependent sentiment analysis and aspect-based sentiment analysis.

2.3.4.1 Target-Dependent Sentiment Analysis

Several approaches have been used to address the task of target-dependent sentiment analysis. Rule-based target-dependent features together with traditional target-independent features for sentiment analysis are used in [1]. Some approaches utilise the syntactic tree of a sentence. For instance, a recursive neural network is used in [41] which passes sentiment signals from sentiment related words to specific targets on a dependency tree. However, data generated on social media blogs and Twitter are not necessarily grammatically correct and can be challenging to parse [66, 67]. More recently, syntax independent features and models are used for solving this task. For instance, word embeddings are used to generate features using the left and the right-hand context of each entity [2]. Also, different neural network architectures such as Convolutional Neural Networks (CNN) and variations of Recurrent Neural Networks (RNN) [42] have been applied to this task.

2.3.4.2 Aspect-Based Sentiment Analysis

Aspect-based sentiment analysis is usually divided into two sub tasks [54]: aspect detection and sentiment identification. In the aspect detection task, the goal is to identify the presence of an aspect in a sentence. Aspects are concept names from a given domain ontology and do not necessarily occur as terms in a sentence. This task sometimes involves extraction of the aspect target expression. An aspect target expression is a span of text naming a particular aspect of the target entity. Sentiment polarity identification assigns a sentiment polarity label (e.g. positive, negative, neutral) to a given aspect of the entity. For example, in the sentence “The pizza is the best if you like thin-crust pizza”, extracted information should be: *food* (aspect), *pizza* (aspect term expression) and *positive* (sentiment).

Separate Tasks

To solve the task of aspect-based sentiment analysis, many of the existing work treat two tasks of aspect and sentiment detection as two separate tasks. Aspect category detection is often formulated as a classification task in a supervised set-

ting. In this framework, text features are defined over the unit of text and are fed into a classifier such as logistic regression or SVM [52, 68]. Convolutional neural networks have also been used in aspect category classification [69], achieving a great performance on SemEval shared task.

Sentiment polarity is then identified for each detected aspect category. Aspect sentiment identification has been addressed both with and without supervision. In a supervised setting, a classifier is often trained using a defined set of carefully designed features [68, 70]. Neural networks, especially variants of RNNs and LSTMs have also been used for the sentiment detection task [71, 72, 73]. These models achieve comparable results to feature-based models where a lot of effort is required for defining the features.

Conditional random fields (CRFs) [74] have been very successful for extracting aspect target expression [75].⁵ However, the success of CRFs depends heavily on the use of an appropriate feature set, which often requires a lot of engineering effort for each task at hand. Unsupervised methods are also popular for this sub-task [76, 58, 77, 76].

Joint Approaches

Joint models have been proposed for detecting aspects and their polarities [78, 79, 3, 80, 81]. In [78] a hierarchical sequential learning is applied using CRFs to jointly extract aspect terms boundaries, opinion polarity, and intensity. Multi-grained LDA has been used [82] to identify topics, sentiment and the evidence that support aspect ratings jointly. Hierarchical deep learning models have also been used by leveraging parse tree of a sentence [3] to extract aspects and their sentiment.

2.3.4.3 Our Work

The task of targeted aspect-based sentiment analysis that is proposed in this thesis, is very similar to the task of aspect-based sentiment analysis. However, in addition to identifying the relevant sentiment for each aspect, we also need to

⁵Aspect target expression is an intermediary task to help in identifying the accurate sentiment of an aspect

identify the target entity that the aspect and sentiment are expressed for. Therefore, the existing methods for the task of aspect-based sentiment analysis are not sufficient for this task. To solve this task, we propose a joint approach in which the target location, aspects, and the polarity towards each aspect are identified in a single step.

Traditionally, a classifier was trained using representations of sentences based on extensive feature engineering which resulted in great performances for different sub-tasks of aspect-based sentiment analysis [83, 84]. Recurrent neural networks (RNN) and specifically Long Short Memory Networks (LSTM) [85] have become increasingly popular, resulting in the state of the art performances in many NLP tasks [86, 87, 88, 71]. Variations of LSTMs and RNNs have also been used for sentiment classification in aspect-based sentiment analysis task [73, 72] which have resulted in comparable performances to the traditional bag of n-grams representations without the need for extensive feature engineering efforts.

Motivated by these successes, we propose discriminative models, based on representations that are obtained using sequential models such as LSTMs. We compare the results with the results obtained using discriminative models that are based on the traditional bag of n-grams representations. These representations can either be sparse and based on generic pre-defined syntactic and semantic features (e.g. uni-grams, bi-grams, POS) or dense and based on linear compositions of the embeddings of the words in the unit of text.

Neural models such as LSTMs often need a large number of training examples to learn good representations. Instead of relying on adding data through expensive human annotation, we investigate data augmentation. Using data augmentation, we can generate training samples with more lexical and syntactic variations compared to the samples in the training set. This can lead to models that can generalise better on unseen data. Data generation and augmentation have been used in the past in machine learning [89] to inject prior knowledge and to improve the performance of the prediction models. In NLP, data augmentation has been used in the past to generate positive [90] and negative examples [91].

Adding more sophisticated features to the traditional bag of n-grams representations or designing more sophisticated sequential neural networks can also be considered for improving the results. However, adding more features or employing a more sophisticated architecture are both orthogonal to data augmentation and can be incorporated further. Here, we look at data augmentation as we find it an intuitive way of incorporating domain knowledge into the representations and the models.

Part I

**Opinion Aggregation For
Neighbourhoods**

Chapter 3

Predicting Population Demographics

In this chapter, we investigate whether the discussions on QA platforms about neighbourhoods reflect the demographic attributes of their population. Examples of demographic attributes are deprivation levels, percent population of Muslim, and percent population of White ethnicity. The values of these attributes are reflected in census data statistics. The focus of this chapter is investigating the following hypothesis, specifically using the discussions from Yahoo! Answers QA platform:

***Hypothesis 1** The language used in QA discussions about neighbourhoods reflects the demographic attributes of their population taken from census records.*

To investigate the above hypothesis, in the next section, we raise appropriate research questions.

3.1 Research Questions

In this chapter, we investigate whether there are correlations between the language used in discussions on QA platform of Yahoo! Answers and the demographic attributes of neighbourhoods. We also investigate the extent in which Yahoo! Answers discussions can be used to predict such attributes. To provide baselines, we also apply our methods to the data from Twitter.

The work in this chapter is driven by the following questions:

Q1: Are there strong and significant correlations between the language used in Yahoo! Answers discussions and the demographic attributes of neighbourhoods?

Q2: How well can features based on text from Yahoo! Answers discussions predict demographic attributes of neighbourhoods?

Q3: What are the limitations of using Yahoo! Answers data in predicting demographic attributes of neighbourhoods?

In the following, we describe the technical background for the methods used in this chapter. The reader can skip Section 3.2 if already familiar with linear regression, non-linear regression using basis functions and Gaussian process regression. We define our approach in Section 3.3. This includes the scope of the problem, the entities of our models and the methods we use for correlation and prediction. This is followed by a description of our dataset, experimental setup and the results. At the end, we discuss our findings and answer the above questions.

3.2 Technical Background

In this section, we provide the technical background to the methods used in this chapter. To show that the discussions on the QA platform of Yahoo! Answers reflect the true demographic attributes of neighbourhoods, we investigate whether these attributes can be predicted using the text from such discussions. To do this, we use regression models. A regression model maps an input (a scalar or a vector) to a continuous-valued output (an attribute).

3.2.1 Regression

Let's assume we have data points $\{x^{(1)}, \dots, x^{(N)}\}$ and observations or output values $\{y^{(1)}, \dots, y^{(N)}\}$ where $x^{(i)} \in \mathcal{X}$ and $y^{(i)} \in \mathbb{R}$. The task of regression is to fit a function $f : \mathcal{X} \rightarrow \mathbb{R}$ to these points. In the simplest regression case, an input point is a scalar, i.e. $\mathcal{X} = \mathbb{R}$. In multidimensional regressions we have $\mathcal{X} = \mathbb{R}^D$. Multidimensional regression is used when an input point is represented using a

vector instead of a scalar. For instance, in text regression, a unit of text can be represented by all the words in the corpus where each dimension represents the value of a frequency function of a word in that text. Therefore, the dimension of the vector representing the text will equal to the number of words in the corpus. Figure 3.2 shows an example of input data points and their corresponding output values where input values are scalars, i.e. $\mathcal{X} = \mathbb{R}$. Regression consists of finding the best fitting line through the points.

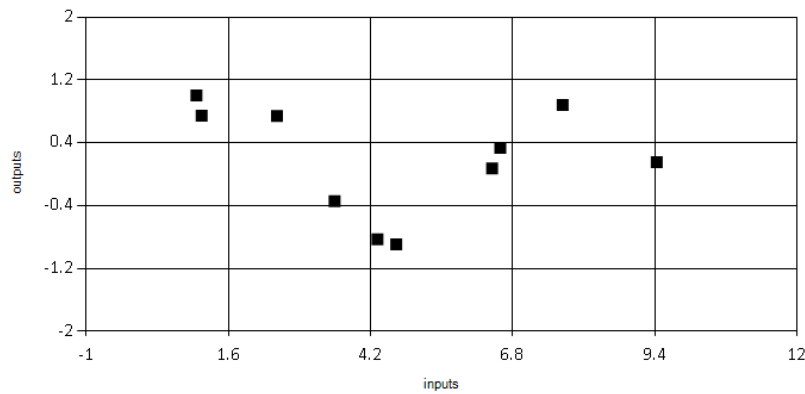


Figure 3.1: Input data and output values.

A straightforward approach in tackling a regression problem is to specify f as a linear combination of a finite set of functions spanned by a given basis as follows. Functions ϕ_0, \dots, ϕ_P map inputs in space \mathcal{X} to a value in \mathbb{R} .

$$f(x) = \sum_{i=1}^P \theta_i \phi_i(x) \quad (3.1)$$

We can then use the function $f(x)$ to perform regression and predict values for the unobserved points after finding the parameters $\theta_1, \dots, \theta_P$. Finding the parameters corresponds to fitting a line through the input points.

3.2.2 Linear Regression

The simplest regression model is linear regression. Linear regression attempts to model the relationship between the input and output values by fitting a linear equation to the observed data. In the one dimensional case, we would take

$\phi_0(x) = 1$ and $\phi_1(x) = x$ which will result in the following definition for $f(x)$:

$$f(x) = \sum_{i=0}^1 \theta_i \phi_i(x) = \theta_0 + \theta_1 x \quad (3.2)$$

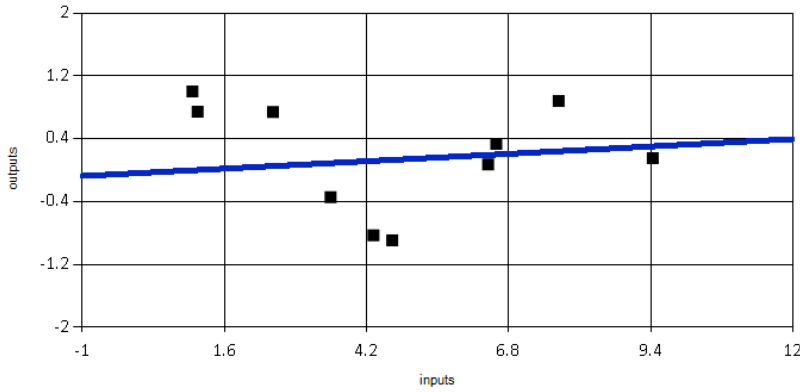


Figure 3.2: Linear regression fits a linear function (line) through the observed data.

In a multidimensional case, such as in text regression, we would define a basis function for each dimension of $\mathbf{x} \in \mathbb{R}^D$ where $\phi_d(\mathbf{x}) = \mathbf{x}[d]$. Then the linear function $f : \mathbb{R}^D \rightarrow \mathbb{R}$ can be defined as follows:

$$f(\mathbf{x}) = \sum_{d=0}^D \theta_d \phi_d(\mathbf{x}) = \theta_0 + \theta_1 \mathbf{x}[1] + \dots + \theta_D \mathbf{x}[D] \quad (3.3)$$

Note that we add a constant term, i.e. bias, by setting $\phi_0(\mathbf{x}) = 1$.

Learning The Parameters

To learn the parameters, $\theta_0, \dots, \theta_P$, we minimise a loss function. The squared loss is a common loss function for regression which is defined as below over all the observed points $\{x^{(1)}, \dots, x^{(N)}\}$. Note that \mathbf{y} is the vector of output values, θ is the vector of parameters and X is the matrix of input vectors where each element is identified with a superscript (i) , i.e. $\mathbf{x}^{(i)}, y^{(i)}$.

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N (y^{(i)} - \mathbf{x}^{(i)T} \theta)^2 = \frac{1}{N} \|\mathbf{y} - X\theta\|^2 \quad (3.4)$$

We can then find the optimum parameters θ analytically using the following formula:

$$\theta = (X^T X)^{-1} X^T \mathbf{y} \quad (3.5)$$

3.2.3 Non-linear Regression

The relationship between output values and input values is not always linear. Non-linear relationship between output and input values can be incorporated into the regression model through defining non-linear basis functions. Polynomial regression is an example of non-linear regression. In this case, for a *one-dimensional* input space, the function f can be defined as follows:

$$f(x) = \sum_{i=0}^P \theta_i \phi_i(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \dots + \theta_P x^P \quad (3.6)$$

This can be further extended into *multidimensional* case using the following formula when $D = 3$ and $P = 2$. The index for each parameter θ consists of three indices ($D = 3$), each indicating a dimension. Each index can take the value zero, one or two indicating the polynomial degree of $P = 2$. Note that here, \mathbf{x} is a vector.

$$\begin{aligned} f(x) = \sum_{d_1, d_2, d_3} \theta_i \phi_i(\mathbf{x}) = & \theta_0 + \theta_{(1,0,0)} \mathbf{x}[1] + \theta_{(0,1,0)} \mathbf{x}[2] + \theta_{(0,0,1)} \mathbf{x}[3] + \\ & \theta_{(1,1,0)} \mathbf{x}[1] \mathbf{x}[2] + \theta_{(1,0,1)} \mathbf{x}[1] \mathbf{x}[3] + \theta_{(0,1,1)} \mathbf{x}[2] \mathbf{x}[3] + \\ & \theta_{(2,0,0)} \mathbf{x}[1]^2 + \theta_{(0,2,0)} \mathbf{x}[2]^2 + \theta_{(0,0,2)} \mathbf{x}[3]^2 \end{aligned} \quad (3.7)$$

Infinite Basis Functions

We can also have an infinite number of basis functions. For instance, our basis functions can be based on the Gaussian Radial Basis Function (RBF) as below, where σ is a constant representing the variance and z is a point in \mathbb{R}^D .

$$\phi_z(x) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right) \quad (3.8)$$

Since z can be anywhere in the space of \mathbb{R}^D , the number of basis functions can be infinite. However, we usually tackle this situation by only considering the basis

functions at the available data points. This is a non-parametric model since the number of parameters grows with the number of training instances.

3.2.4 Gaussian Process Regression

Gaussian processes (GPs) [92] are powerful non-parametric tools that can be used in supervised learning. One of the main advantages of GPs is that they have the ability to provide uncertainty estimates and to learn the noise and smoothness parameters from training data. Figure 3.3 [93] shows how GPs can fit a non-linear function through data and estimate the uncertainty. The blue line indicates the predicted values. The grey region shows the 95% confidence interval (the distance of two +/- standard deviations from the mean) which indicates how uncertain the model is about the predicted value at each point.

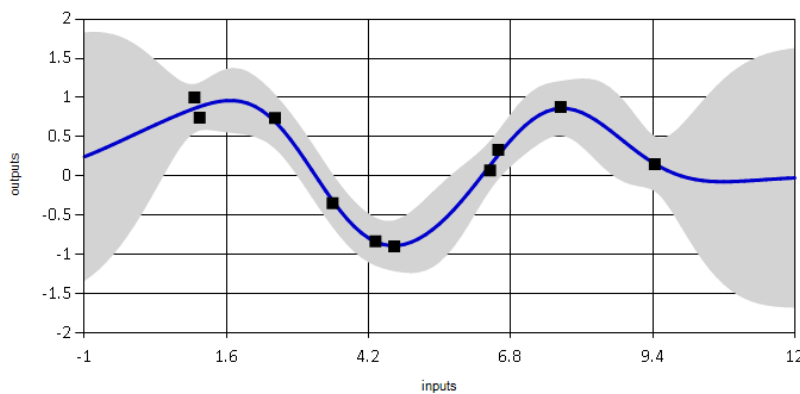


Figure 3.3: A GP can fit a non-linear function (line) through the observed data and estimate the uncertainty.

A GP is a collection of random variables in which the joint distribution of any of the subset of these variables is also a Gaussian distribution. A Gaussian process is different from a Gaussian distribution. A Gaussian distribution is a probability distribution that is defined by its mean μ and covariance σ : $x \sim \mathcal{N}(\mu, \sigma)$. A uni-variate Gaussian distribution can be defined by the function $f(x)$ as below:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.9)$$

Gaussian processes can be considered as a Gaussian distribution with in-

finitely many variables. A Gaussian process is a random function that is specified by a mean function $m(x)$ and a covariance function $k(x, x')$ as follows:

$$f(x) \sim GP(m(x), k(x, x')) \quad (3.10)$$

To define a GP, we need to choose a mean and a covariance function. In many applications, no prior knowledge is available about the mean function, $m(x)$ of a Gaussian process. This is usually assumed to be zero. The covariance function, $k(x, x')$ can be any function that takes two arguments. Covariance function should be able to generate a non-negative definite covariance matrix K . Choosing a covariance function is a way of incorporating prior knowledge about the process such as its smoothness, its periodicity, etc. Even though there are many possible covariance functions, the most frequently used is the RBF or squared exponential function. There are many other kernels such as linear,¹ periodic, noise, etc. The RBF kernel is defined as below:

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{\|x - x'\|^2}{2\ell^2}\right) \quad (3.11)$$

By looking at the Equation 3.11, we can see that if two points are very close to each other, the value of the kernel will be very close to σ_f^2 . The value of the kernel decreases exponentially as the distance between two inputs increases. Here, σ_f and ℓ are the hyperparameters of the model and can be tuned using cross-validation. Using the RBF as the covariance function is similar to regression using many Gaussian like basis functions on all the inputs and not only the training set [92].

The output of the Gaussian process regression is a normal distribution specified by a mean and a variance. The mean represents the predicted output value and the variance represents the confidence of the prediction. A useful feature of GPs is that different kernels can be defined over all the input dimensions or some of the input dimensions. These kernels can then be combined by summation or

¹GP regression with a linear kernel is equivalent to the bayesian linear regression.

multiplication to obtain new kernels as the following equations show. Here, K_1 and K_2 are kernels which are combined to obtain a new kernel K .

$$K = K_1 + K_2 \quad (3.12)$$

$$K = K_1 \times K_2 \quad (3.13)$$

3.3 Approach

In this section, we discuss the scope of the problem, entities that are used in our approach and the methods we use to answer the questions raised in this chapter.

3.3.1 Domain Entities and Concepts

Here, we formally define the entities: locations, attributes and the documents. These entities are used in Part I of this thesis.

Location: A location refers to a neighbourhood or an area in a city. Each location corresponds to an entity that is identified by its unique name. A location also has geolocation which consists of a latitude and a longitude. The set of all the locations under consideration are represented by the set:

$$L = \{\ell_1, \ell_2, \dots, \ell_M\}$$

Attribute: Demographics data reflected in the census is divided into several categories such as *ethnic origin*, *religion*, *employment status*, etc. Each category is further divided into subcategories which we refer to as attributes. For instance, the category *religion* includes attributes *Muslim*, *Jewish*, *Buddhist*, etc. An attribute a , therefore, refers to a subcategory of demographics data represented by a continuous value per each geographical unit. The continuous value often indicates the percentage of the population in the geographical unit. Examples are percent population of *Muslim* and percent population of *Asian*.

Document For each location ℓ_m , $m = 1, \dots, M$, we retrieve discussions from Yahoo! Answers or microblogs from Twitter. We combine all the discussions or

microblogs related to a location into a single document. Documents are represented by:

$$D = \{d_1, d_2, \dots, d_M\} .$$

where d_m is the document containing all QA discussions or Twitter microblogs for the location ℓ_m .

3.3.2 Unit of Analysis

In the experiments in this chapter, unit of analysis is a neighbourhood or a location entity.

3.3.3 Document Representation

To use a document in a regression model as input, we represent the document in vector space. To investigate which representation of a document is most suitable for the task of predicting demographic attributes, we perform predictions using several representations that are described below. Each representation converts a text document d (for location l) to a numeric vector \mathbf{x} . Let's assume that the document d^* obtained for the location Norbury is the following:²

*“I have heard that there is a big Jewish community living in **Stamford Hill** which are usually a peaceful bunch. But I have never been there. Have you checked it for yourself? I live in **Norbury**. There is a big population of Muslim here. Therefore, you can find many halal shops around.”*

In the following, we explain each representation and present an illustration of that representation using the above document.³

Normalised tf-idf A very popular method for representing a document using its words is the tf-idf approach [96]. Tf-idf is short for term frequency-inverse document frequency where tf indicates the frequency of a term in the document and idf is a function of the number of documents that a terms has appeared in. In

²In reality documents are much larger than this example.

³Recently, the use of character n-grams has become more popular [94, 95]. There are several advantages to this. For instance, character n-grams can capture the similarities between the different morphological forms of a word. They can also capture the similarity between words in documents when they are misspelt. In this thesis, the use of character n-grams was not explored due to the time limitations.

a tf-idf representation, we assume that the words that have appeared in a document d with a higher frequency while not present in many other documents are most indicative of the attributes of its location l .

To discount the bias for areas that have a high number of QAs or tweets, we normalise all tf-idf values by the length of the document. The length of each document is defined as the number of its nondistinctive words. In a tf-idf representation, the order of the words in the document is not preserved. Assume that d is a document representing a location and t is a term in the vocabulary where the number of terms in vocabulary is $|V|$. The normalised tf-idf for the term t in d can be calculated as below:

$$\text{Normalised tf}(d, t) = \frac{\text{Frequency of Term } t \text{ in Document } d}{\text{Number of Tokens in Document } d} \quad (3.14)$$

$$\text{Normalised tf-idf}(d, t) = \frac{\text{Normalised tf}(d, t)}{\log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing the term } t}\right)} \quad (3.15)$$

The normalised tf-idf representation of the document d^* is illustrated in Figure 3.4.

hindu	jewish	community	halal	muslim	alcohol				
0	0.7	...	0.1	...	0.29	...	0.62	...	0

Figure 3.4: The normalised tf-idf representation of word uni-grams of d^* .

Normalised tf-idf of n-grams This is similar to the normalised tf-idf but we use word n-grams instead of only word uni-grams. Here, we consider $n = 1, 2$. Bi-grams can capture linguistic phenomena such as negation (e.g. “not poor”) or intensification (e.g. “very poor”) in text. The normalised tf-idf of word uni-grams ($n = 1$) and word bi-grams ($n = 2$) of the document d^* is illustrated in Figure 3.4:

hindu	jewish	community	halal	muslim	halal_shop	alcohol									
0	0.7	...	0.1	...	0.29	...	0.62	...	0.5	...	0.2	...	0.1	...	0
jewish_community										big_population					

Figure 3.5: The normalised tf-idf of word uni-grams and word bi-grams of d^* .

Binary Here, a document is represented by a vector that contains only zeros or ones. With this representation, we assume that only the presence of a term in a document d is an indicator for attributes of its location l .

$$\text{Binary}(d, t) = \begin{cases} 1 & \text{if } t \in d \\ 0 & \text{otherwise} \end{cases} \quad (3.16)$$

The binary representation of d^* is illustrated in Figure 3.4:

hindu	jewish	community	halal	muslim	alcohol		
1	1	...	1	...	1	...	0

Figure 3.6: Binary representation of word uni-grams of the document d^* .

Context normalised tf-idf This representation is the same as the normalised tf-idf representation. However, a new document \tilde{d} for a location ℓ is now defined by combining the context window around each mention of the location name in the document d . The window includes the sentence with the location mention and two sentences on its either sides. This is mainly because in a QA more than one neighbourhood can be discussed. We want to observe whether the context around each neighbourhood’s name is more relevant to the neighbourhood than the entire QA. Normalised tf-idf is then calculated over the terms in the document \tilde{d} . Note that the contexts of two or more locations can overlap which leads to their corresponding documents having some text in common.

The \tilde{d} for the document d^* is then “*But I have never been there. Have you checked it for yourself? I live in **Norbury**. There is a big population of Muslim here. Therefore, you can find many halal shops around*”. The context normalised-tf-idf representation of d^* is illustrated in Figure 3.7. Note that the features cor-

responding to the term “jewish” and “community” is now zero in this representation.

hindu	jewish	community	halal	muslim	alcohol
0	0	...	0.29	0.62	0

Figure 3.7: The normalised tf-idf representation of the context around Norbury for the document d^* .

Context Binary This representation is the same as the binary representation but is defined over the document \tilde{d} for a location ℓ . The context binary representation of the document d^* is similar to its context normalised tf-idf representation which is illustrated in Figure 3.7. However, the context binary representation will contain only zeros and ones.

Context Point-Wise Mutual Information (PMI) Point-wise mutual information is a measure of association between two terms in a corpus. Here, we assume one of the terms to be a word in vocabulary and the second term to be a location name. Let’s assume that $\text{count}(\ell, t)$ is the number of times the term t and the name of the location ℓ have appeared in the same context window. $\text{count}(\ell)$ is the number of times the name of location ℓ has appeared in the document d (which is the same as the number of times that the location name has appeared in the corpus) and $\text{count}(t)$ is the number of times the term t has appeared in the corpus. PMI for each term t in vocabulary and the name of a location ℓ is calculated using the following formula:

$$\text{PMI}(\ell, t) = \log \frac{\text{count}(\ell, t)}{\text{count}(\ell) \times \text{count}(t)} \quad (3.17)$$

Note that here, document d for location l is not defined. A vector representation for location l consists of the PMI values for each word in the vocabulary with the location l . The context window around the name of the location name consists of the current sentence (with the location name) and two sentences on either sides of the current sentence. The representation for the location Norbury in

the above example is then similar to the context normalised tf-idf with non-zero values obtained through the PMI formula.

3.3.4 Correlation Analysis

Correlation analysis measures the strength of the association between two variables. To investigate whether discussions on QA platform of Yahoo! Answers about neighbourhoods reflect the demographic attributes of their population, we study the correlations between term frequencies in such discussions and different demographic attributes. Here, we mainly look at the normalised tf-idf frequency measure.⁴ To calculate the correlations, for each term, we define a vector with the dimension of the number of locations. The value of the m -th cell in this vector represents the normalised tf-idf value of the term for the location ℓ_m . For each demographic attribute, we also define a vector with the dimension of the number of locations. The value of the m -th cell in this vector represents the value of that attribute for the location ℓ_m . We then calculate the Pearson correlation coefficient (ρ) between these two vectors to measure the strength of the association between a term and an attribute. Pearson correlation coefficient evaluates the linear relationship between two continuous variables. We calculate the Pearson correlation coefficient between all the terms in the corpus and each demographic attribute.

Since we run many correlation tests, we need to correct the significance values (p-values) for multiple tests [97]. The Bonferroni correction [98] is a multiple-comparison correction to the p-value and is used when several dependent or independent statistical tests are being performed simultaneously. Bonferroni adjustment ensures an upper bound for the probability of having an incorrect significant result among all the tests. We adjust all the p-values in our experiments for multiple tests using the Bonferroni correction.

⁴Similar approach can be taken for other frequency measures introduced in the previous section.

3.3.5 Prediction

To further investigate the extent in which the language used in QA discussions reflects the demographic attributes of neighbourhoods, we study whether we can use the QA discussions about neighbourhoods to predict their demographic attributes. We define the task of predicting a continuous-valued demographic attribute for unseen locations as a regression task given the text feature representations of documents defined for those locations. A separate regression task is defined for each demographic attribute.

3.3.5.1 Linear Regression

We first assume a linear relation between the input variables, i.e. term frequency features, and the output variables, i.e. demographic attributes. Since the dimension of the input space (equal or greater than the size of the vocabulary⁵) is very high relevant to the number of training points (locations), we use elastic net regularisation to avoid over-fitting. Elastic net combines the $L1$ and $L2$ penalties of lasso and ridge methods linearly. Therefore the loss function from equation 3.4 can be further modified as follows, where the hyperparameters λ_1 and λ_2 can be tuned using cross-validation. Linear regression is described in details in Section 3.2.

$$\mathcal{L} = \frac{1}{N} \|\mathbf{y} - X\theta\|^2 + \lambda_1 \|\theta\| + \lambda_2 \|\theta\|^2 \quad (3.18)$$

3.3.5.2 Spatial Regression

A location is a spatial entity with a latitude and a longitude, as mentioned earlier. In spatial regression, not only we can use the term frequencies as features, we also take into account the geographical positions of location entities. For instance, assume that we are predicting the deprivation index of a neighbourhood. Our regression model may find that the terms “poor” and “deprived” are strong indicators for a neighbourhood being highly deprived. Moreover, we know that the neighbourhood we are making predictions for is very close to other neighbourhoods with high levels of deprivation. This information can be incorporated

⁵The number of frequent terms used for vector representation is 8k for Yahoo! Answers corpus and 17k for the Twitter corpus.

in our prediction model to improve its performance. For spatial regression, we propose the three following methods which are inspired by the models described in Section 3.2.

Using Coordinates Information

To incorporate the geographical information of locations, we simply add the coordinates information, i.e. latitude and longitude, of the locations to their feature representations. We then apply a linear regression model on the new combined feature vectors.

Gaussian Process Regression

Gaussian process regression is often used when predicting values for unobserved spatial entities. Gaussian process regression is related to Kriging [99]. Kriging was originally used in the field of mineral resource and reserve valuation where a relatively small set of samples were available. Kriging or GP regression is now used in many other fields.

GP regression is capable of modelling non-linear regression problems. This is well-suited to spatial prediction problems where non-linearities can be assumed between the points in space and their output values.⁶ To incorporate the spatial information of location entities in our model, we add the coordinates of each location to its representation, as above. To capture the non-linearity assumption, we propose defining an RBF kernel on the features defined by the coordinates information (latitude and longitude). We also define a linear kernel over the text features. This is because the non-linearity relation exists mainly in the geographical space. We then combine these two kernels using summation to obtain a single kernel. As we have seen in Section 3.2, in GP, kernels can be summed or multiplied to obtain new kernels.

Non-linear Basis Functions

The results of a GP model are less interpretable in comparison to a linear regression model where we can observe the coefficients or parameters of the model, i.e. θ . To use a linear regression model and to capture the non-linear property

⁶The relevancy of points in space rapidly decreases non-linearly as the distance between the points increases.

of the attributes of neighbourhoods at the same time, we use a linear regression and RBF basis functions as defined in Equation 3.8. Here, $(x - z)$ is defined as the physical distance of two locations on earth which is calculated using the latitude and the longitude of the two locations.⁷ We take $\sigma^2 = 2$ (kilometres) which is the optimal value obtained using cross-validation. Figure 3.8 shows the value of this RBF function as the distance between two locations in terms of kilometres increases. As we can see, the value of the function tends to zero when the distance grows larger than 4 kilometres.

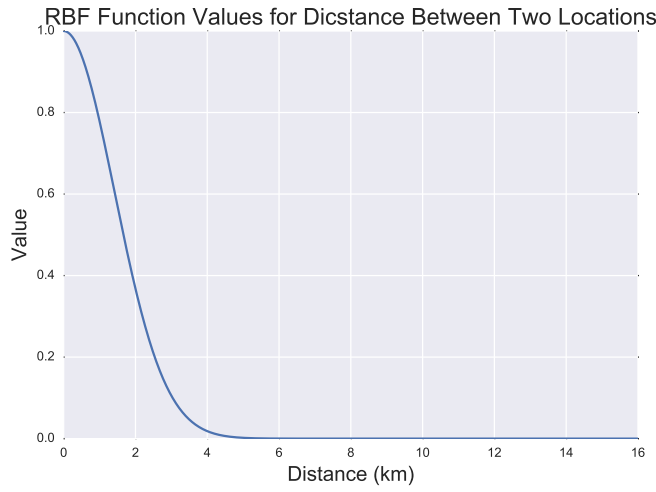


Figure 3.8: The value of the RBF function as the distance between two locations increases.

Therefore, the regression function can be defined as below:

$$f(x) = \sum_{i=0}^M \hat{\theta}_i \hat{\phi}_i(x) + \sum_{j=0}^D \tilde{\theta}_j \tilde{\phi}_j(x) \quad (3.19)$$

Here, $\hat{\phi}$ is the RBF basis function. Note that there are M RBF basis functions, one for each of the available locations (training plus test). This is different than N that is the number of the observed points, i.e. training set. Therefore, $\hat{\phi}_i = \phi_{\text{RBF}(x^{(i)})}$ measures the non-linear distance of a location to the location ℓ_j . The basis function $\tilde{\phi}$ is linear and defined for every D dimension of text features space, e.g. $\tilde{\phi}_j = x[j]$. Moreover, D is the size of the vocabulary if a word uni-gram represen-

⁷The distance is calculated using the Haversine method [100].

tation is used.

3.3.5.3 Evaluation Metric

To measure the performance of a regression model, residual-based methods such as mean squared error are commonly used. Ranking metrics such as Pearson correlation coefficient ρ have also been used in the past [101, 11, 19]. Using a ranking measure, in general, has few advantages compare to a residual-based measure. First, ranking evaluation is more robust against extreme outliers compared to an additive residual-based evaluation measure [102]. Second, it is suggested that in tasks where ranking is the main underlying goal in building a regression model, ranking performance is the correct evaluation metric [102]. Finally, ranking metrics are very interpretable [102].

In discovering attributes of neighbourhoods, we usually care about the relative value of an attribute rather than its absolute value. For instance, it might be sufficient to know whether an area has a lower rate of crime in comparison with other areas. In this case, knowing the exact number of reported crimes in the area is not necessary. Therefore, we measure the performance of our prediction models using Pearson correlation coefficient.

3.4 Dataset

In this section, we explain how we select neighbourhoods of interest and describe the procedure in which we obtain text from both QA platform of Yahoo! Answers and Twitter for each neighbourhood. This is followed by the description of demographic attributes taken from census records and methods for unifying the unit of analysis across all the datasets.

3.4.1 Neighbourhoods

The names of the neighbourhoods for a city are taken from the GeoNames gazetteer.⁸ Each location entity in the gazetteer is defined by its name, category and geolocation. Geolocation includes a latitude and a longitude. Therefore, a

⁸<http://www.geonames.org/>

neighbourhood is defined by a point on earth and not a geographical shape with boundaries.

We then take all the entities in the gazetteer for a city that have a category relevant to an area.⁹ This approach provides us with 589 location entity names for the Greater London metropolitan area.

3.4.2 Yahoo! Answers Data

We collect questions and answers (QAs) from Yahoo! Answers using its public API.¹⁰ For each neighbourhood, a query consists of the name of the neighbourhood together with keywords “London” and “area”. This is to prevent obtaining irrelevant QAs for ambiguous entity names such as Victoria. No time limit has been imposed on the period in which these QAs have been logged. Each QA consists of a title and a content which is an elaboration on the title. This is followed by a variant number of answers. In total, we collect 12,947 QAs across all London neighbourhoods. The obtained QAs span over a period of around five years.

In a QA thread, it is common for users to discuss characteristics of several neighbourhoods. This means that the same QA can be assigned to more than one neighbourhood. Some examples of such QA threads can be seen in Table 3.1.

Table 3.1: Examples of QA threads where more than one area is discussed.

<p>Q: What area of london should i live in? A: Cool areas to live in at the moment are: / <u>Clapham</u> / <u>Balham</u> / <u>Battersea</u> / <u>Hoxton</u> / <u>Camden</u></p>
<p>Q: Where can i find a jewish shop in london? A: The main Jewish Communities in London are <u>Stamford Hill</u> and <u>Golders Green</u>, plus <u>Hendon</u> and <u>Edgware</u>. All have many Kosher and Judaica stores on their high streets.</p>

Figure 3.9 shows the histogram of the number of QAs for each neighbourhood. As the figure shows, the majority of areas have less than 100 QAs and out of those areas, some have less than 10. There are a few number of areas with over 100 QAs. Well-known and popular London neighbourhoods such as Camden Town and Chelsea are amongst the areas with a high number of QAs.

⁹PPL (populated place), PPLX (section of a populated place) and AREA

¹⁰<https://developer.yahoo.com/answers/>

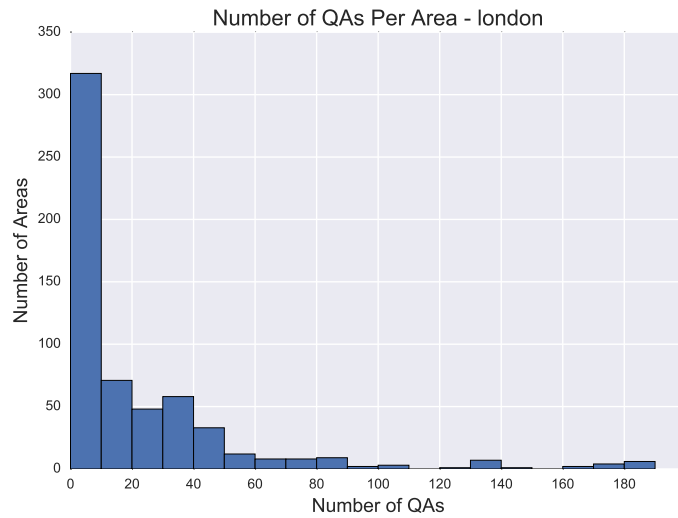


Figure 3.9: Histogram of the number of QAs per areas of London.

Pre-processing and Filtering We split each document of all the relevant QAs for a neighbourhood into sentences. We then remove neighbourhoods that contain less than 40¹¹ sentences as we consider them to be under-represented. At the end, we are left with 363 areas. Figure 3.10 shows the histogram of the number of sentences per each remaining neighbourhood. As the figure shows, most neighbourhoods have less than 1000 sentences. In extreme cases, there are neighbourhoods that have up to 4000 sentences.

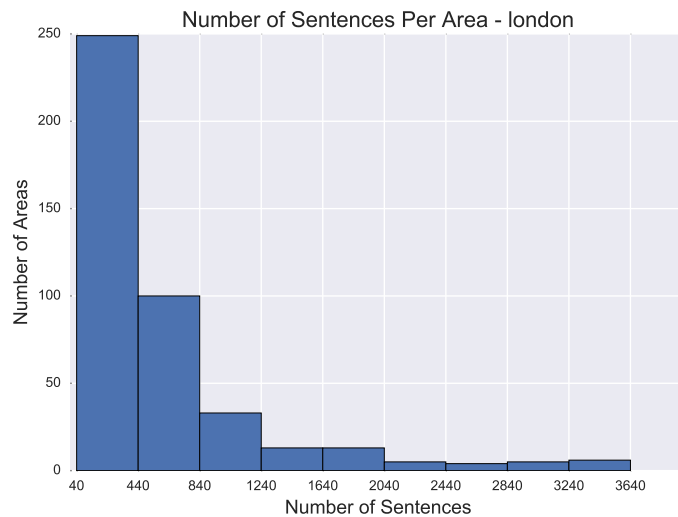


Figure 3.10: Histogram of the number of sentences per areas of London.

¹¹This number was selected heuristically by observing the data

We then remove URLs and stop words from all the documents. All the tokens in all documents are lowercased and then lemmatised. Lemmatisation is a special case of text normalisation. Lemmatisation removes inflectional endings of a word and return the base of a word. For example, a lemmatiser will transform the word “dogs” to “dog” and “children” to “child”. To keep the most frequent words, we remove any token that has appeared less than 5 times in total across the whole corpus and also in less than 5 unique QAs. This leaves us with 8k distinct tokens.

3.4.3 Twitter Data

To collect data from Twitter, we use the geographical bounding box of London, defined by the northwest and southeast points of the Greater London. We then use this bounding box and collect all the tweets that are geotagged and are created within this box. We do this using Twitter streaming API.¹² We collect Twitter data for 6 months between July 2015 and December 2015. At the end, we have around 2,000,000 tweets. In a heuristic approach, to filter out the tweets that are spam or advertisement, in each day, we remove the tweets of users that have tweeted more than once in the same hour. We assume people usually do not log more than one tweet per hour.¹³

To assign tweets to different neighbourhoods, for each tweet, we calculate the distance between the location that it was blogged from and the centre points of all the neighbourhoods in our dataset. Note that the centre point for each neighbourhood is provided in the gazetteer as discussed in Section 3.4.1. We then assign the tweet to the closest neighbourhood that is not further than 1 km from the tweet’s geolocation. At the end of this process, we have a collection of tweets per each neighbourhood. We combine all the tweets of a neighbourhood to create a single document. Figure 3.11 shows the number of tweets per each neighbourhood. As we can see, the majority of neighbourhoods have less than 1000 tweets. The West End and Oxford Circus are amongst areas with the highest number of tweets.

¹²<https://dev.twitter.com/streaming/overview>

¹³This assumption is made by observing our data.

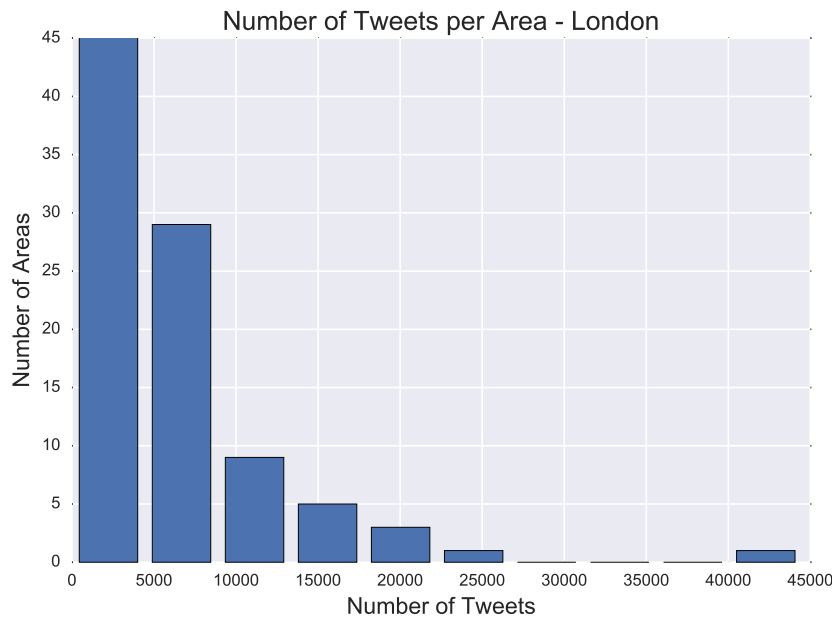


Figure 3.11: Histogram of the number of tweets per areas of London.

Pre-processing and Filtering We remove all the target words (words starting with the symbol @ are called target words) from all the documents. The pre-processing of each document is then similar to the QA documents. At the end, we have 17k distinct frequent tokens across all the Twitter corpus. As we can see, the number of distinct tokens in Twitter data is much higher than the number of distinct tokens in Yahoo! Answers data (8k). This can be because in Twitter, unlike in Yahoo! Answers, people create and use many compound words (e.g. sundayroast, poshwashlondon) or informal abbreviations (e.g. imo, tbh).

3.4.4 Population Demographics Data

Population demographics data is taken from the UK census provided by the Office for National Statistics.¹⁴ The last UK census was carried out in 2011. Census data collection is repeated every 10 years. In the UK, census data is provided for specific geographical units that are created solely for the purpose of census data collection. The smallest unit that census data is aggregated for, is called LSOA

¹⁴<http://www.ons.gov.uk/>

(Lower Super Output Layer).¹⁵ An LSOA is defined by its ID and a geographical shape which has a centroid point.¹⁶ Greater London is divided into 4,835 LSOAs. LSOAs are not necessarily equal in their geographical size, but they have been designed to have a population of around 1,500.

3.4.5 Unification of Geographical Units

Attributes in census data are collected for geographical shapes (with boundaries) called LSOAs as explained earlier. Our units of analysis for text are neighbourhoods which have coordinates representing their centre points and not boundaries. To unify our units of data across text and demographic attributes, we align the units by keeping the gazetteer units. For this, we map the values of attributes from LSOAs to neighbourhoods using the following heuristic approach.

By observing the QA data, we have noticed that often, when people talk about a neighbourhood, e.g. Camden Town, they refer to the area close to its centre point. In other words, the information provided for neighbourhoods in QA discussions are very local to this point. To keep this locality, for each attribute, we assign only the values of the nearby LSOAs to the respective neighbourhood. For this, we calculate the distance between each neighbourhood and all the LSOAs within London. The distance is calculated between the coordinates of a neighbourhood and the coordinates of the centroid point of each LSOA. For each neighbourhood, we select the m closest LSOAs that are not further than k kilometres away. The value of each attribute for a neighbourhood is then calculated by averaging the values of that attribute over the selected LSOAs. We apply this mapping to all the selected demographic attributes. The values for m and k are selected heuristically to be $m = 10$ and $k = 1$.¹⁷

To provide a view of the geographical mismatch between LSOAs and neighbourhoods, we show the map of London which is divided into LSOAs in Fig-

¹⁵<http://www.ons.gov.uk/ons/guide-method/geography/beginner-s-guide/census/super-output-areas--soas-/index.html>

¹⁶Centroid is the centre of mass of a geometric object of uniform density

¹⁷We also experimented with other heuristic methods. For instance, instead of mapping the demographic attributes into neighbourhoods, we mapped the text features into LSOAs and other geographical units used for aggregation of census data. However the results of these unifications when used in correlation analysis or predictions were poor.

ure 3.12. LSOAs are geographical shapes with centroid points. Centroid points of LSOAs are marked with dark dots and neighbourhoods are marked with green circles.

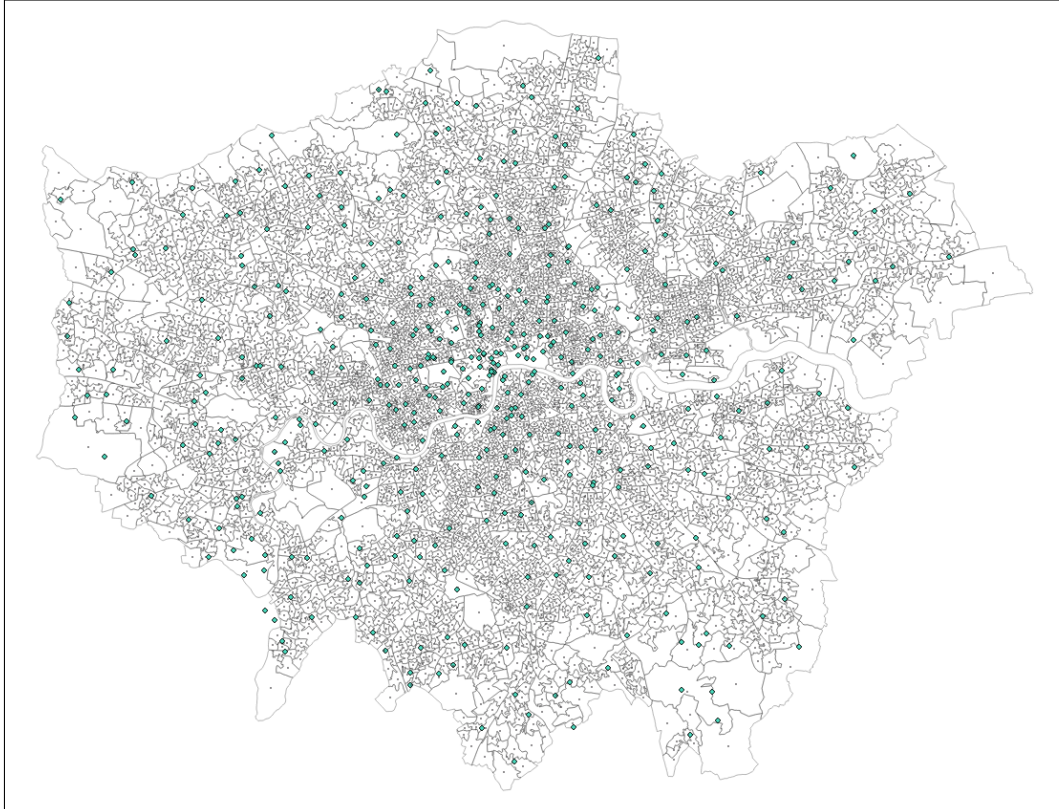


Figure 3.12: Figure shows the division of London by LSOAs. It also shows the neighbourhoods of London. Centroid points of LSOAs are marked with dark dots and neighbourhoods are marked with green circles.

3.5 Experiments

In this section, we describe the experimental set up.

3.5.1 Scope

In this chapter, we mainly focus on attributes of neighbourhoods of London. London is a big cosmopolitan city and a popular destination for people from other countries or cities, to visit or to immigrate to. Therefore there are many discussion threads on characteristics of its neighbourhoods on Yahoo! Answers. For comparison purposes and for studying the limitations of using Yahoo! Answers

data, we briefly discuss the availability of data for neighbourhoods of other cities in Section 3.7.

3.5.2 Demographic Attributes

There are many attributes across several categories in the census data. We will carry most of our experiments on a selected set of diverse attributes that are taken from *religion*, *ethnicity*, *price* and *deprivation* categories. These attributes are: percent population of *Jewish*, percent population of *Muslim*, percent population of *Hindu*, percent population of *Buddhist*, percent population of *Black* ethnicity, percent population of *White* ethnicity, percent population of *Asian* ethnicity, *Price* (average house prices) and IMD. IMD is the index of multiple deprivations created by the UK government to measure deprivation in local authorities. It covers multiple aspects of deprivation such as income, employment, education, crime, living environment and health deprivation. The higher the value of IMD, the more deprived an area is. Later, we report results on a wide range of 62 demographic attributes that are available in our dataset.

3.5.3 Evaluation Setup

We evaluate the performance of each regression task using 10 folds cross-validation. In each fold, we use 75% of the data for training and the remaining 25% for validation. At the end, we report the average performance over all the folds together with the standard deviation.

Training and validation sets are sampled using stratified sampling [103] for each attribute. Stratified sampling is used when sub-populations within an overall population vary. In these cases, it is advantageous to sample each subpopulation independently. Each subpopulation is called a stratum. A stratified sample is made up of different strata of the population, for example, samples from areas with a low or a high rate of crime. The sample size for each stratum is proportional to the size of the stratum.

3.5.4 Implementation

Linear regression models with elastic net is implemented using the scikit-learn¹⁸ library in Python. Gaussian process regression was implemented using the GPy library [104]. Moreover, to calculate the distance between two locations on earth using the Haversine method, we use the jcoord library in Java.

3.6 Results

In this section, we discuss our findings on correlation analysis and prediction results.

3.6.1 Correlation

To investigate the hypothesis that we raised in this chapter and to study whether discussions on the QA platform of Yahoo! Answers about neighbourhoods reflect the true demographic attributes of these neighbourhoods, in this section, we study whether meaningful correlations exists between the term frequency features of Yahoo! Answers discussions and the values of the demographic attributes for neighbourhoods. We compare these correlations with the correlated terms from Twitter.

Yahoo! Answers vs. Twitter We first quantitatively compare whether significant correlations exist between the demographic attributes and term features of Yahoo! Answers and Twitter. We then present examples of the correlated terms from both sources with a few demographic attributes for a qualitative observation.

NUMBER OF CORRELATED TERMS The number of significantly correlated terms from both Yahoo! Answers and the Twitter with the selected demographic attributes are shown in Table 3.2. Note that the number of unique frequent words in Twitter(17k) is almost twice as in Yahoo! Answers (8k). The column “#significant” shows the total number of terms with a significant correlation¹⁹ to the attribute presented in the first column. The next columns show the counts of terms that have significant correlations with Pearson correlation coefficients (ρ) in the

¹⁸<http://scikit-learn.org/>

¹⁹p-value < 0.001 and adjusted using Bonferroni correction

given ranges. The last column shows the number of terms that are negatively and significantly correlated with the attribute. The source that has the highest number of correlated terms with each attribute is highlighted in bold.

Table 3.2: The number of significantly correlated terms (p -value < 0.001 and adjusted using Bonferroni correction) from both Yahoo! Answers and Twitter. “Y! A” is used in place of Yahoo! Answers due to the space limit. To see examples of the correlated terms, refer to Table 3.3 and 3.4.

Attribute	Source	#significant	#> 0.4	#0.3 – 0.4	#0.2 – 0.3	# $\rho < 0$
IMD	Y! A	115	1	48	66	0
	Twitter	17	0	10	7	0
Price	Y! A	50	2	36	12	0
	Twitter	1120	312	533	275	0
Jewish%	Y! A	48	7	31	10	0
	Twitter	6	0	5	1	0
Muslim%	Y! A	87	0	59	28	0
	Twitter	13	1	8	4	0
Hindu%	Y! A	8	2	3	3	0
	Twitter	5	0	3	2	0
Buddhist%	Y! A	1	0	1	0	0
	Twitter	934	18	728	188	0
Black%	Y! A	114	4	59	51	0
	Twitter	2	0	2	0	0
White%	Y! A	8	0	0	0	8
	Twitter	0	0	0	0	0
Asian%	Y! A	6	0	3	3	0
	Twitter	1	0	1	0	0

As the table shows, the number of highly correlated terms in Yahoo! Answers are much higher than in Twitter for the majority of the attributes. This is especially the case for attributes *Jewish%*, *Muslim%* and *Black%*.

On one hand, Twitter has a high number of significantly correlated terms with attributes *Price* (1120) and *Buddhist%* (934). On the other hand, there is only 1 term from Twitter that is significantly correlated with the attribute *Asian%*. This number is 2 for *Black%* and 6 for *Jewish%*.

SEMANTIC RELATEDNESS To observe whether the correlated terms from Yahoo! Answers and Twitter are semantically related to the respective attributes, we present some of the top correlated terms with some of the attributes for both of these sources.

Yahoo! Answers Table 3.3 shows the Pearson correlation coefficients (ρ) of some of the top correlated terms with a few selected demographic attributes. The selected attributes are those that Yahoo! Answers has many correlated terms with. As we can see from the table, many of the top correlated terms are semantically related to the respective attribute with strikingly high correlations. For example, “poverty”, “notorious” and “rundown” are related to areas of higher deprivation. Similarly the terms “matzo” (Jewish bread), “jewish” and “jew” are relevant to the attribute *Jewish*.

Even though, finding ethical biases is not the objective of this thesis, this correlation analysis provides us with some insights into such biases in the society.²⁰ For example, the terms that are highly correlated with the percent population of *Black* ethnicity may reflect the stereotypes that are present in people’s perception. Examples of such stereotypes are the terms “gang”, “violent” and “drug”.

Table 3.3: Significantly correlated (p-value < 0.001) terms with the highest correlation coefficients for the selected demographic attributes using the normalised tf-idf features of **Yahoo! Answers** data.

Black%		Jewish%		(High) Price		IMD	
Term	ρ	Term	ρ	Term	ρ	Term	ρ
violent	0.44	matzo	0.45	townhouse	0.4	hurt	0.4
gang	0.43	harmony	0.45	fortune	0.39	poverty	0.36
drug	0.42	jewish	0.41	qatar	0.39	drug	0.36
rob	0.4	jew	0.41	diplomat	0.39	boy	0.36
danger	0.39	unfairly	0.42	exclusive	0.37	cockney	0.35
knife	0.39	flyover	0.41	hectic	0.36	victim	0.35
integration	0.38	ark	0.38	refine	0.36	mug	0.34
black	0.38	staw	0.38	desirable	0.35	trouble	0.34
boy	0.38	arab	0.39	celeb	0.34	notorious	0.34
evenly	0.38	freehold	0.37	cosmopolitan	0.33	rundown	0.33
dangerous	0.37	kosher	0.32	aristocratic	0.33	redevelop	0.33
stab	0.37	traditional	0.35	fashionable	0.32	slum	0.32

Twitter Table 3.4 shows the top correlated terms with the attributes for which Twitter has a high number of correlated terms. For many of the top correlated terms with the attribute *Price*, semantic relevance is apparent and the correlations are very high ($\rho > 0.5$). For example, the terms “luxury”, “classy”, and

²⁰Finding discriminative and hateful messages in users’s tweets has been the subject of studies in NLP and ethics [105].

“stylish” seem related to aspects of expensive areas. “Tea”, “teatime”, “delight” and “truffle” seem related to the social activities of the upper class. Many of the top correlated terms with *IMD* that are presented in the table are very specific to London. For example, areas of East London are known to be more deprived. The terms “east”, “eastend” and “eastlondon” refer to this fact. Also, “cockney” is a dialect traditionally spoken by the working class, and thus less advantaged Londoners. The relevance of terms seems less salient for the attribute *Buddhist%*, even though there is a high number of correlated terms from Twitter with this attribute. Some of the terms that can be considered related to aspects of the *Buddhist* religion are “think”, “learn” and “mind”.

Table 3.4: Significantly correlated (p -value < 0.001) terms with the highest correlation coefficients for selected demographic attributes using the normalised tf-idf features of **Twitter** data.

Buddhist%		(High) Price		IMD	
Term	ρ	Term	ρ	Term	ρ
think	0.44	luxury	0.66	east	0.39
en	0.42	tea	0.64	eastlondon	0.36
long	0.42	teatime	0.61	eastend	0.36
learn	0.40	delight	0.60	yeah	0.33
presentation	0.40	truffle	0.60	studio	0.33
mind	0.40	car	0.60	shit	0.32
para	0.40	classy	0.59	craftbeer	0.30
todo	0.40	stylish	0.59	ass	0.30
thing	0.40	gorgeous	0.59	music	0.30
heart	0.40	lamborghini	0.59	neighbour	0.29
remember	0.39	interiordesign	0.58	tune	0.28
beautiful	0.39	couture	0.56	progress	0.28

Interestingly, terms extracted from Yahoo! Answers and Twitter seem to offer two different kinds of insights. On one hand, terms extracted from Yahoo! Answers are more encyclopedic as they tend to offer definitions or aspects related to each attribute. For example, terms “matzo”, “harmony”, and “kosher” are related to the cultural aspects of the *Jewish* religion and the terms “jew” or “jewish” are linguistically associated with its name. On the other hand, Twitter terms can offer geographically related information (e.g. “east”, “eastend” for *IMD*) or knowledge about related socio-cultural aspects (e.g., “tea”, “truffle” for *Price*).

Overall, correlation results suggest that there is a wealth of terms, both in Yahoo! Answers and Twitter, which can be used to predict the attributes from the population demographics.

3.6.2 Prediction

To investigate the hypothesis that we raised in this chapter and to show that the discussions on QA platform of Yahoo! Answers about neighbourhoods reflect their attributes, in this section, we look at how well their attributes can be predicted using the terms used in these discussions. If demographic attributes can be predicted with high correlation coefficients using Yahoo! Answers data, we conclude that the discussions on this platform are reflective of the attributes of the neighbourhoods. To do this, we first find the best way to represent the text from Yahoo! Answers. To find the most suitable representation, we look at the prediction results of different representations proposed earlier in Section 3.3.3. We then observe whether we can achieve better prediction results using spatial regression methods. Finally, we compare the prediction performances using Yahoo! Answers and Twitter data for a wide range of attributes.

3.6.2.1 Representations

In this section, we look at the performances of different representations of Yahoo! Answers text. Table 3.5 shows the prediction results of the set of selected attributes using different representations of Yahoo! Answers data. The results using normalised tf-idf of word n-grams (uni-grams and bi-grams) and the normalised tf-idf of context window are lower compared to other representations and therefore are omitted from the table due to the space limit. Results are averaged over 10 folds and standard deviations are shown in parenthesis. All correlations are statistically significant with a p-value < 0.01 . Results having a * superscript have at least 2 folds with a p-value > 0.01 .

As we can see from the table, the performances of representations that are based on the text obtained from the context around the location names are in general higher than the representations that are based on the text from the en-

Table 3.5: Prediction results in terms of ρ using different feature representations of **Yahoo! Answers** data. Results are averaged over 10 folds and standard deviations are shown in parenthesis. All correlations are statistically significant with a p-value < 0.01 . Results having a * superscript have at least 2 folds with a p-value > 0.01 .

	Normalised tf-idf	Binary	Context PMI	Context Binary
Muslim %	0.51(0.07)	0.54 (0.15)	0.59 (0.11)	0.58 (0.08)
Jewish %	0.42(0.08)	0.46 (0.08)	0.52 (0.09)	0.56 (0.07)
Hindu %	0.32(0.10)*	0.36 (0.14)*	0.39 (0.17)*	0.45 (0.11)*
Buddhist %	0.24(0.10)*	0.32 (0.12)*	0.31 (0.16)*	0.35 (0.08)*
Black %	0.60(0.07)	0.61 (0.08)	0.59 (0.10)	0.72 (0.08)
Asian %	0.40(0.07)	0.44 (0.14)	0.49 (0.12)	0.46 (0.11)
White %	0.58(0.06)	0.63 (0.06)	0.59 (0.19)	0.61 (0.11)
IMD	0.69(0.03)	0.65 (0.05)	0.74 (0.03)	0.75 (0.06)
Price	0.69(0.05)	0.63 (0.07)	0.55 (0.13)	0.73 (0.07)
Average	0.48(0.07)	0.50 (0.09)	0.52 (0.12)	0.57 (0.09)

tire QAs. This can be because QA discussions can often contain noise or they can include information about other neighbourhoods. The context around the name of a neighbourhood seems to be more representative of the attributes of the neighbourhood.

Further, we can see that the binary representations of both the entire QAs and the context achieve higher performances compared to the normalised tf-idf of the entire QAs and the context PMI representations, respectively. This means that only the presence (and not the frequency) of specific terms in the discussions for a neighbourhood or in the context of a neighbourhood is sufficient for predicting the attributes of the neighbourhood. The best performance is achieved using the context binary representation with an average Pearson correlation coefficient of 0.57.

3.6.2.2 Spatial Prediction

Attributes of neighbourhoods present spatial properties. This means that neighbourhoods that are geographically very close to each other share similar characteristics. This similarity decreases rapidly (often non-linearly) as the distance between neighbourhoods increases. This results in neighbourhoods forming clusters of high and low values with regards to different characteristics. This is illus-

trated in Figure 3.13 where maps show the distribution of different attributes over neighbourhoods of London. Darker colours indicate higher values for each attribute. The clustering effect is especially evident for the attribute *Jewish%* where there is mainly one region with areas of high concentration of people with the Jewish religion. Distinct clusters are also present for percent population of *Asian* origins. *IMD* and average house *Price*, however, have many clusters of high and low values that are spread across London.

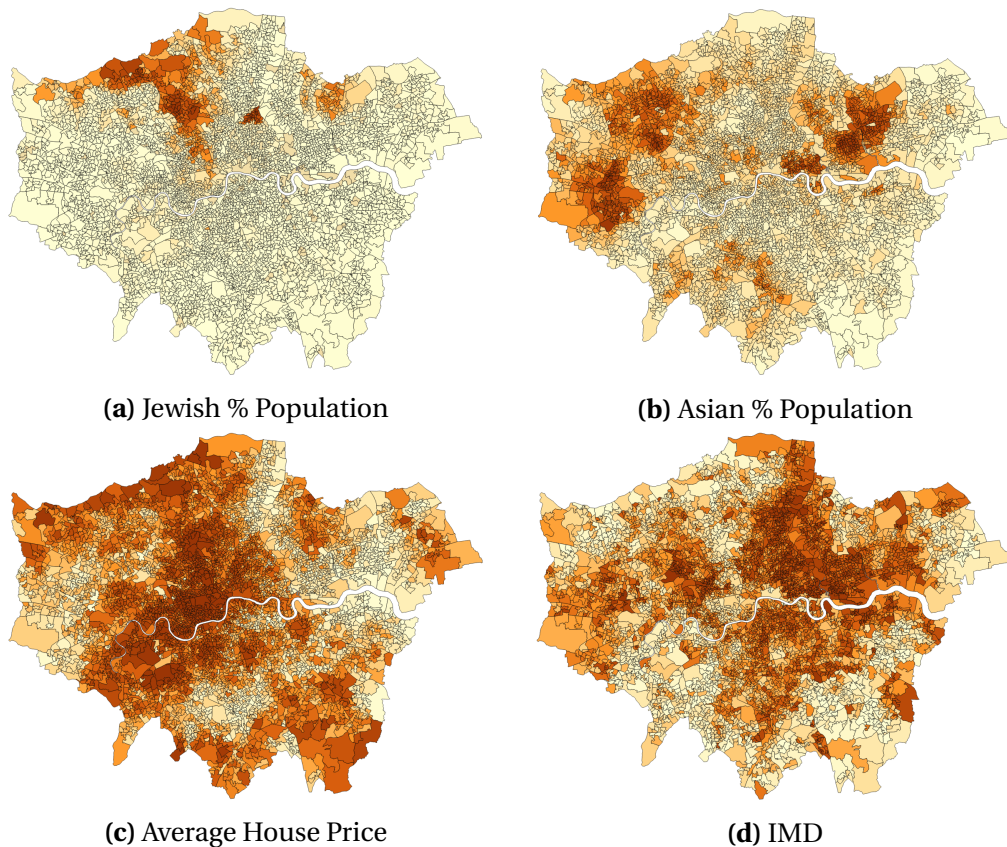


Figure 3.13: Maps of selected demographic attributes over LSOAs of London. Darker regions indicate higher values for an attribute.

In this section, we investigate whether we can predict attributes of neighbourhoods using text from Yahoo! Answers with a higher performance using spatial prediction methods. Here, we show the results of spatial predictions for the selected attributes using different spatial regression models introduced in Section 3.3.5.2. Results are presented in Table 3.6. In this table, *Lin + Lin* is a linear regression model applied on a combination of text features and coordinates

information. *Lin+RBF* is the regression model that uses non-linear radial basis functions over the coordinates information and linear basis functions over the text features. *GP Lin+RBF* is a GP regression model applied on a combination of coordinates information (RBF) and the text features (Lin). *GP RBF* applies a GP regression model with an RBF kernel on the coordinates information only without using any text features. *W* and *C* indicate whether text (words) or coordinates information (latitude/longitude) are used as features for each model. Results are averaged over 10 folds and standard deviations are shown in parenthesis. All correlations are statistically significant with a p-value < 0.01. Results having a * superscript have at least 2 folds with a p-value > 0.01. An upward arrow indicates an increase of performance in comparison with the results obtained using a linear regression model and text features presented in Table 3.5.

Table 3.6: Prediction results on a selected set of attributes using spatial regression and **Yahoo! Answers** data. *W* and *C* indicate whether text features (context-binary representation) or coordinates are used as features. All correlations are statistically significant with a p-value < 0.01. Results having a * superscript have at least 2 folds with a p-value > 0.01. An upward arrow indicates an increase of performance in comparison with the results obtained using a linear regression model and text features presented in Table 3.5.

	Lin + Lin (W+C)	Lin+RBF (W+C)	GP Lin+RBF (W+C)	GP RBF (C)
Muslim %	0.58 (0.08)	0.80 (0.06) ↑	0.82 (0.06) ↑	0.71 (0.24) ↑
Jewish %	0.56 (0.07)	0.67 (0.07) ↑	0.67 (0.06) ↑	0.70 (0.12) ↑
Hindu %	0.45 (0.11)*	0.67 (0.08) ↑*	0.82 (0.04) ↑*	0.54 (0.24) ↑*
Buddhist %	0.35 (0.08)*	0.57 (0.05) ↑*	0.69 (0.07) ↑*	0.60 (0.12) ↑*
Black %	0.72 (0.08)	0.82 (0.03) ↑	0.85 (0.05) ↑	0.82 (0.05) ↑
Asian %	0.46 (0.11)	0.73 (0.09) ↑	0.73 (0.15) ↑	0.67 (0.29) ↑
White %	0.61 (0.11)	0.80 (0.06) ↑	0.79 (0.11) ↑	0.23 (0.21) ↑
Price	0.73 (0.07)	0.84 (0.02) ↑	0.70 (0.08)	0.49 (0.34)
IMD	0.75 (0.06)	0.85 (0.03) ↑	0.86 (0.04) ↑	0.81 (0.08) ↑
Average	0.57 (0.09)	0.76 (0.05) ↑	0.78 (0.07) ↑	0.62 (0.18) ↑

Results in Table 3.6 show that adding coordinates information to a linear model (Lin + Lin) does not improve the prediction performance of any of the selected attributes. Using a GP regression model and coordinates only (GP RBF) results in a higher performance by 5% than using text features only in a linear regression model. This is particularly evident for the attribute percent popula-

tion of *Jewish%*. This is because there is only one cluster of neighbourhoods with high values for this attribute. This means that by knowing only whether a neighbourhood is geographically very close to this cluster, we can infer if it has a high population of *Jewish%*. Predictions using coordinates information only and a GP regression model (GP RBF), however, result in very high standard deviations. This means that the geographical locations of the neighbourhoods selected for the training can have a high effect on the performance of this model.²¹

The highest spatial prediction results are achieved when both text and coordinates information are used with a non-linear model. Using RBF basis functions on coordinates information and linear basis functions on text features (Lin+RBF) results in 19% increase in performance compared to a linear non-spatial model. The best results are obtained when a GP regression model (GP Lin+RBF) is used over a combination of text and the coordinates information with 21% increase on average. It is interesting to note that the performance of the Lin+RBF model is comparable to the performance of the GP Lin+RBF model (only 2% lower) while being highly interpretable. For instance, we observe that for the percent population of *Jewish%*, the distance to the areas of Mill Hill, Hendon and some of the other well-known Jewish areas of London are amongst the features with the highest coefficients.

Incorporating the non-linear spatial information is particularly helpful in predicting attributes such as *Hindu%*, *Asian%* and *Jewish%*. Improvements are less pronounced when predicting the attribute *Price*. Spatial prediction models improve the prediction results for attributes that present higher clustering effects. Clustering effect for attributes *Asian%* and *Jewish%* is evident in Figure 3.13.

3.6.2.3 Yahoo! Answers vs. Twitter

To provide comparisons, in this section, we present the results of predictions using Twitter data and Yahoo! Answers over a wider range of attributes. To com-

²¹Note that we choose the training set using stratified sampling with respect to an attribute, disregarding the geographical locations of the areas.

pare the strength of *text* features of Yahoo! Answers discussions with the Twitter microblogs, we compare their performances using the linear regression models that utilise text features only. We use the best representation of Yahoo! Answers text for this task²² which is the context-binary representation. To provide a fair comparison, we also examined different representations of Twitter data for the prediction tasks. Binary representation of Twitter data results in the highest prediction performance and therefore will be used in this section.

Tables 3.7 and 3.8 show prediction results in terms of Pearson correlation coefficients over a wide range of 62 attributes. Results having a * superscript have at least 2 folds with a p-value > 0.01. Results are averaged over the entire 10 folds. Attributes are divided into categories such as *Religion*, *Ethnicity*, *Employment*, *Education*, etc.

Overall, the results show that Yahoo! Answers performs slightly better than Twitter with an average 4% increase over all the attributes. The Wilcoxon signed rank test shows that their results are significantly different from each other (p-value < 0.01).

Results further indicate that text features based on Yahoo! Answers perform strongly in predicting some of the attributes that are related to religion (*Jewish%*, “*Muslim%*”) and ethnicity (*Black%*). Yahoo! Answers can predict the attributes *IMD* and *Price* with strikingly high correlation coefficients ($\rho > 0.7$).

Twitter can predict most of the religion related attributes well with the exception of the *Jewish%*. Twitter is poor in predicting ethnicity related attributes such as *Black%* and *White%*. This is consistent with the correlation results that we have seen in Table 3.2 where Twitter, unlike Yahoo! Answers, does not have many significantly correlated terms with attributes *White%* (0), *Black%* (2) and *Jewish%* (6). Twitter, however, performs stronger than Yahoo! Answers when predicting the attributes in the categories of *Age Group*, *Residential Status*, and *Car Ownership*.

²²Over the selected attributes

Table 3.7: Prediction results on a wide range of attributes in terms of ρ using *context-binary* representation for **Yahoo! Answers** and binary representation for **Twitter**. Results are averaged over 10 folds and standard deviations are shown in parenthesis. All correlations are statistically significant (p-value < 0.01). Results having a * superscript have at least 2 folds with a p-value > 0.01.

Attribute	Yahoo! Answers	Twitter
Price & Deprivation	0.74 (0.07)	0.66 (0.09)
Mean Price	0.73 (0.07)	0.68 (0.09)
IMD	0.75 (0.06)	0.63 (0.09)
Religion	0.49 (0.08)	0.41 (0.1)
Jewish %	0.56 (0.07)	0.15 (0.16) *
Muslim %	0.58 (0.08)	0.52 (0.09)
Hindu %	0.45 (0.11) *	0.50 (0.09)
Buddhist %	0.35 (0.08) *	0.49 (0.08) *
Ethnicity	0.59 (0.1)	0.43 (0.09)
White %	0.61 (0.11)	0.40 (0.08)
Asian %	0.46 (0.11)	0.27 (0.10)
Black %	0.72 (0.08)	0.52 (0.11)
Mixed %	0.55 (0.13)	0.54 (0.06)
Residential Status	0.58 (0.1)	0.63 (0.07)
Owned Outright %	0.72 (0.09)	0.60 (0.09)
Owned With A Mortgage Or Loan %	0.69 (0.10)	0.75 (0.09)
Social Rented %	0.61 (0.10)	0.53 (0.07)
Private Rented %	0.63 (0.07)	0.59 (0.04)
At Least One Usual Resident %	0.35 (0.16)	0.58 (0.08)
No Usual Residents %	0.38 (0.12)	0.54 (0.06)
Whole House Or Bungalow Detached %	0.54 (0.11)	0.71 (0.07)
Whole House Or Bungalow Semi Detached %	0.68 (0.11)	0.71 (0.05)
Flat Maisonette Or Apartment Percent Sale	0.72 (0.05)	0.74 (0.08)
	0.51 (0.13)	0.52 (0.09)
Employment	0.62 (0.08)	0.50 (0.08)
No Adults In Employment-Dependent Children	0.66 (0.08)	0.53 (0.05)
All Lone Parent With Dependent Children	0.62 (0.08)	0.45 (0.10)
Lone Parents Not In Employment	0.60 (0.08)	0.48 (0.07)
Lone Parent Not In Employment %	0.62 (0.05)	0.53 (0.08)
Economically Active Total	0.53 (0.08)	0.55 (0.11)
Economically Inactive Total	0.59 (0.11)	0.55 (0.07)
Economically Active Employee	0.41 (0.09)	0.48 (0.08)
Economically Active Self Employed	0.72 (0.04)	0.49 (0.05)
Economically Active Unemployed	0.76 (0.04)	0.56 (0.07)
Economically Active Full Time Student	0.53 (0.15)	0.44 (0.11)
Employment Rate	0.66 (0.08)	0.48 (0.07)
Unemployment Rate	0.71 (0.06)	0.46 (0.07)

Table 3.8: cont.

Attribute	Yahoo! Answers	Twitter
Education	0.60 (0.08)	0.62 (0.07)
No Qualifications %	0.72 (0.03)	0.61 (0.06)
Highest Level Qualification 1 %	0.74 (0.06)	0.76 (0.05)
Highest Level Qualification 2 %	0.69 (0.08)	0.78 (0.04)
Highest Level Qualification Apprenticeship %	0.57 (0.08)	0.73 (0.05)
Highest Level Qualification-3 %	0.21 (0.13)*	0.26 (0.13)*
Highest Level Qualification Level 4+ %	0.72 (0.06)	0.74 (0.04)
Highest Level Of Qualification Other %	0.59 (0.12)	0.45 (0.13)
Schoolchildren/Full Time Students 18+ %	0.59 (0.09)	0.63 (0.06)
Age Group	0.61(0.09)	0.63 (0.05)
0-15 %	0.56 (0.10)	0.59 (0.05)
16-29 %	0.69 (0.07)	0.70 (0.07)
30-44 %	0.60 (0.07)	0.62 (0.04)
45-64 %	0.47 (0.08)	0.64 (0.05)
65+ %	0.69 (0.08)	0.64 (0.06)
Working Age %	0.66 (0.10)	0.60 (0.05)
Health	0.55 (0.08)	0.42 (0.09)
Day To Day Activities Limited A Lot %	0.54 (0.10)	0.29 (0.14)
Day To Day Activities Limited A Little %	0.47 (0.08)	0.52 (0.09)
Day To Day Activities Not Limited %	0.50 (0.11)	0.41 (0.05)
Very Good Or Good Health %	0.64 (0.07)	0.43 (0.10)
Fair Health %	0.63 (0.04)	0.60 (0.09)
Bad Or Very Bad Health %	0.53 (0.09)	0.31 (0.09)
Car Ownership	0.65 (0.05)	0.77 (0.05)
No Cars Or Vans In Household %	0.77 (0.04)	0.83 (0.07)
1 Car Or Van In Household %	0.69 (0.03)	0.68 (0.05)
2 Cars Or Vans In Household %	0.77 (0.03)	0.80 (0.02)
3 Cars Or Vans In Household %	0.67 (0.06)	0.81 (0.04)
4 Or More Cars Or Vans In Household %	0.54 (0.09)	0.73 (0.07)
Cars Per Household	0.50 (0.09)	0.80 (0.04)
Household Composition	0.62 (0.09)	0.57 (0.08)
Couple With Dependent Children %	0.60 (0.07)	0.73 (0.06)
Couple Without Dependent Children %	0.66 (0.09)	0.58 (0.08)
Lone Parent Household %	0.56 (0.09)	0.38 (0.10)
One Person Household %	0.63 (0.10)	0.70 (0.06)
At Least One Aged 16 + English Main Language %	0.64 (0.08)	0.55 (0.08)
No Aged 16 + Have English Main Language %	0.63 (0.12)	0.52 (0.11)
Average	0.62 (0.08)	0.58 (0.07)

We also looked at the terms with the highest coefficients in the regression models for each attribute and source. We observe that similar to the correlated terms, the terms with the highest coefficients in Yahoo! Answers tend to be re-

lated to the definition or the concept of the respective attribute. Examples are the term “caribbean” for the attribute *Black%* and “asian” for the attribute *Muslim%*.

Moreover, in Yahoo! Answers, sometimes the name of the attribute is amongst the terms with the highest coefficients, e.g. the term “asian” for the attribute *Asian%*, and the term “jewish” for the attribute *Jewish%*. This is something that is not usually observed in Twitter. Terms from Twitter that have the highest regression coefficients can be related to the areas or geographical regions of London (e.g. “mileend” for the attribute *Muslim%*,²³ “southlondon” for the attribute *Black%*²⁴ and “eastlondon” for *IMD*). Also, many terms related to the activities of people are amongst the terms with high coefficients in Twitter. Examples are: “golf” for the attribute *Age Group 45-64* and “personaltrainer” for the attribute (high) *Price*.

It is interesting to observe the coefficients of the regression models when predicting the attributes that are related to car ownership using Twitter data. Note that these attributes can be predicted with strikingly high correlation coefficients of up to 0.83. Terms “cocktail”, “gig”, “pub”, “cinema”, “beer” and “wine” are amongst the terms with the highest negative coefficients for attributes related to car ownership. This can indicate that it is less likely for people to own cars in areas that are good for going out which often tend to be central.

3.7 Limitations

Here, we look at some of the limitations of using Yahoo! Answers and Twitter data for predicting attributes of neighbourhoods.

3.7.1 Yahoo! Answers

The success of a prediction model depends heavily on the availability of data for training. London is a big cosmopolitan city and many discussions can be found on Yahoo! Answers regarding its neighbourhoods. In this section, we look

²³Mile End is an area of London with a high population of Muslims.

²⁴Neighbourhoods with a black majority tend to be located in the southern part of London.

at the coverage of Yahoo! Answers QAs for neighbourhoods of two other cities and compare it with the coverage for neighbourhoods of London.

Availability of Data

So far, we have seen that there are many discussions on Yahoo! Answers platform for many neighbourhoods of London. The availability of data makes it possible to discover patterns in text and to predict demographic attributes using text features based on these discussions.

Here, we look at the availability of discussions on Yahoo! Answers platform for neighbourhoods of the two selected cities of Manchester and Birmingham. We provide comparisons with the data from Twitter. Yahoo! Answers and Twitter data for neighbourhoods of both cities are collected using the same methods that we employed for the neighbourhoods of London (discussed in Section 3.4).

Table 3.9 shows the number of areas with at least one QA discussion on Yahoo! Answers for cities of Manchester, Birmingham and London. The number in parenthesis next to the name of the city indicates the number of neighbourhoods taken from the gazetteer for each city. The table also shows the maximum, the minimum and the median number of QAs for each neighbourhood in different cities. We can see that while 89% of London neighbourhoods are discussed on Yahoo! Answers, these percentages are 25% and 27% for Birmingham and Manchester, respectively.

Table 3.9: Number of QAs discussing neighbourhoods of Birmingham, Manchester and London.

	#Areas with QA (% #Area)	Max #QA	Min #QA	Median #QA
Birmingham (321)	83 (25%)	51	0	0
Manchester (302)	82 (27%)	134	0	0
London (589)	527 (89%)	186	0	3

Table 3.10 shows the number of areas that have at least one tweet associated with them in our dataset (collected over 6 months), the minimum, maximum and the median number of tweets per areas of each city.

As tables 3.9 and 3.10 show, a larger number of areas are covered in Twitter

data²⁵ compared with the data from Yahoo! Answers. Attributes of over 70% of areas in Manchester and Birmingham cannot be predicted using Yahoo! Answers data due to the lack of coverage. Moreover, the number of areas that have associated QAs may not be enough to develop a regression model that can generalise well to the unseen neighbourhoods that have been discussed on Yahoo! Answers. Manchester and Birmingham are amongst the most populated and cosmopolitan cities of the UK. The coverage of Yahoo! Answers discussions can be lower for smaller and less known cities.

Table 3.10: Number of tweets collected for the areas of Birmingham, Manchester and London.

	#Areas with Tweets (% #Area)	Max #Tweet	Min #Tweet	Median #Tweet
Birmingham (321)	259 (80%)	4341	1	21
Manchester (302)	296 (98%)	6506	2	51.5
London (589)	587 (99%)	44510	1	456

3.7.2 Twitter

People use Twitter to express their spontaneous feelings and opinions about their lives and the events that are happening in the world. Therefore, depending on the time that Twitter data is obtained, different topics and trends can be dominant which subsequently can affect the correlation and regression results. For instance, when analysing the number of correlated terms from Twitter and demographic attributes, we observe a high number of correlated terms with the attribute *Price* and the percent population of *Buddhist*.

The high number of significantly correlated terms from twitter with the attribute *Price* is due to the fact that there are many terms that are related to how expensive an area is. Some of these terms can be found in Table 3.4. However, the high number of correlated terms with the attribute *Buddhist* can not be quite justified. We found many French stop words such as “en” and “le” to be amongst the top correlated terms with the attribute *Buddhist*. It is possible that a French

²⁵Note that we have only collected Twitter data for a period of 6 months. The coverage from Twitter may improve further by collecting tweets for a longer period of time.

event was organised in an area with a high population of Buddhists and the related tweets have affected our correlation results. The same issue is observed when studying the correlated terms from Twitter with the attribute *IMD*. The term “londontattoconvention” appears to be strongly correlated with high deprivation. The reason behind this high correlation is that a tattoo convention is held annually around September in London area of Hackney which is known to be a deprived area. These examples show that when Twitter data is obtained using its streaming API, the data can get influenced by such temporal events, some of which can be one-off cases. Community question answering platforms such as Yahoo! Answers are less prone to such issues and biases.

3.8 Discussion

In this chapter, we investigated the hypothesis that the discussions on QA platforms about neighbourhoods reflect their demographic attributes. For this, we studied the relation between the text taken from the discussions on QA platform of Yahoo! Answers about the neighbourhoods of London and the demographic attributes reflected in the UK census data. This included studying the correlations between the text features from Yahoo! Answers discussions and a diverse set of demographic attributes. Moreover, we studied how well these attributes can be predicted using such text features. We compared our results to the performances achieved using the text features of the microblogs of Twitter; a platform that has been used in studying deprivations of neighbourhoods in the past.

Here, we can answer the questions that we raised at the beginning of this chapter.

Q1: *Are there strong and significant correlations between the language used in Yahoo! Answers discussions and the demographic attributes of neighbourhoods?*

A1: Our correlation analysis results indicate that for a diverse set of selected attributes, there is a high number of text features from QA discussions that have strong and significant correlations with each attribute. This is specifically

evident for the deprivation score, *IMD* and many of the ethnicity related attributes. For the majority of the selected attributes, a higher number of significantly correlated terms exists in Yahoo! Answers discussions in comparison with Twitter. Even though, for some attributes especially *Price*, there are higher numbers of correlated terms from Twitter.

Q2: *How well can features based on text from Yahoo! Answers discussions predict demographic attributes of neighbourhoods?*

A2: In this chapter, we have shown that predictions using Yahoo! Answers data can achieve on average a correlation coefficient of 0.62 over a wide range of attributes taken from census data. This is 4% higher than what can be achieved using Twitter data. While Yahoo! Answers text features on average can achieve higher performances on categories of *Price*, *IMD*, *Health*, *Religion* and *Ethnicity*, Twitter can perform better on categories of *Car Ownership*, *Education* and *Age Group*. We further show that we can improve the prediction performances of many demographic attributes by using spatial prediction models.

Q3: *What are the limitations of using Yahoo! Answers data in predicting demographic attributes of neighbourhoods?*

A3: One of the main limitations of using the discussions of a QA platform such as Yahoo! Answers for predicting attributes of neighbourhoods is the coverage of these discussions for different neighbourhoods. For example, less known or less central areas are not discussed as much as central and popular areas within London. In our experiments, we only use 363 out of 589 London areas because the remaining 226 areas are under-represented (i.e. have less than 40 sentences). The same problem exists for the neighbourhoods of cities that are less cosmopolitan. In this chapter, we looked at the availability of data for neighbourhoods of two other major cities in the UK. As we have seen, the amount of data that is available for neighbourhoods of the cities of Manchester and Birmingham is very limited compared to the amount of the data that is

available for the neighbourhoods of London. While the coverage of Twitter data is also lower for neighbourhoods of Manchester and Birmingham, the limitation is less pronounced.

In summary, by observing the results of our experiments in this chapter, we conclude that the discussions on QA platform of Yahoo! Answers about neighbourhoods are reflective of the attributes of those neighbourhoods. Not only we can find many correlated terms from these discussions with many attributes of neighbourhoods, these attributes can be predicted on average with a Pearson correlation coefficient of 0.62. The hypothesis raised in this chapter holds specifically for the city of London which is a big cosmopolitan city. This can be true for other major cities around the world, but the investigation is out of the scope of this thesis.

Chapter 4

Predicting Perceived Characteristics of Neighbourhoods

In the previous chapter, we showed that there are strong and meaningful correlations between the language used in Yahoo! Answers discussions and many of the demographic attributes of the population of neighbourhoods. Moreover, features based on these discussions can be used to predict a wide range of attributes with a high accuracy.

In this chapter, we investigate whether we can also predict the perceived characteristics of neighbourhoods using the text from Yahoo! Answers discussions. We refer to these characteristics as *aspects*. The values for the perceived characteristics, unlike the demographic attributes, are not available in census records. Moreover, the values for these aspects cannot be obtained by a population count or measured through objective statistics. Finally, these aspects are subject to personal opinions. Take the aspect *Trendy* as an example. Different people can describe trendiness of an area in different ways. The value of this aspect for a neighbourhood cannot be measured with a number. However, often a consensus can be found on whether an area is perceived to be trendy or not.

Predicting perceived characteristics of neighbourhoods is important since these characteristics are not available in census records or other statistics. The importance of identifying aspects of areas for settlers and travellers is recognised

by big travel and neighbourhood expert sites such as AirBnB¹ and Spareroom,² where their experts provide information on a set of aspects for a limited number of areas in some cities including London. This information is in the form of aspect labels for areas. For instance, on Spareroom, Camden Town is labeled with *Nightlife*, *Multicultural* and *Well-connected* but not with *Quiet*.

The coverage of the provided aspects and areas through sites such as AirBnB and Spareroom is limited. It is expensive to rely on experts to provide information on aspects of new areas and new cities or information on new aspects for areas. This is because one needs to have a good understanding of a city and its neighbourhoods to provide such information. Alternatively, the values of these aspects for different neighbourhoods can be inferred from people's discussions about neighbourhoods on QA platforms such as Yahoo! Answers. In this chapter, therefore, we investigate whether discussions on QA platforms about neighbourhoods reflect the perceived characteristics of neighbourhoods, similar to the demographic attributes. We explore the following hypothesis that we introduced in Chapter 1:

Hypothesis 2 *The language used in QA discussions about neighbourhoods reflects their perceived characteristics.*

To investigate whether this hypothesis holds, in the next section, we raise appropriate research questions.

4.1 Research Questions

To study whether QA discussions reflect the perceived characteristics of neighbourhoods, we focus on the two following perspectives. First, we aim to study whether there are meaningful and strong correlations between the terms used in QA discussions and the presence of aspects in neighbourhoods. Second, we

¹<https://www.airbnb.co.uk/>

²<https://www.spareroom.co.uk>

investigate whether aspects of interest can be predicted using the frequency features of the terms used in such discussions with a high accuracy.

In the absence of official statistics for the values of the aspects of neighbourhoods, we use the labels provided by experts on Spareroom. The values provided for these aspects are not numeric. Each aspect is treated as a label. This means that if an area is labeled with an aspect, the area is known for having that aspect. The area lacks an aspect if it is not labeled with that aspect. Therefore, the values for these aspects are binary.

We expect the task of predicting aspects of neighbourhoods to be more challenging than predicting the demographic attributes. This is because both the values for these aspects and the opinions expressed about these aspects are subjective. Moreover, the labels collected from expert sites are partial. In other words, aspect labels are not necessarily provided for all areas or all aspects of interest. Obtaining aspect labels to train a prediction model is a costly task which requires expert knowledge. Therefore, it is important to make predictions for new aspects in a cost-effective and yet an accurate manner. This chapter is driven by the following questions.

Q1: Are there significant correlations between text features from Yahoo! Answers discussions and the perceived characteristics of neighbourhoods?

Q2: How well can the perceived characteristics of neighbourhoods be predicted using text features from Yahoo! Answers discussions?

Q3: Can we predict aspects of neighbourhoods in a cost-effective way using the discussions on Yahoo! Answers?

Q4: What are the limitations of using the discussions from Yahoo! Answers in predicting perceived characteristics of neighbourhoods?

To provide baselines for correlations and predictions, we also apply our methods to the data from Twitter.

In the following, we define our approach. This includes the scope of the problem, the entities of the system and the methods we use for correlation and

prediction. We then describe the technical background to the models used in this chapter. The reader can skip this section if familiar with document classification models such as MaxEnt or Generalised Expectation for feature labeling. We then provide a description of our dataset and the experimental setup. Finally, the results of our experiments are presented, after which we discuss our findings and answer the above questions.

4.2 Technical Background

Predicting perceived characteristics of neighbourhoods using text from Yahoo! Answers discussions can be framed as a document classification task. In this setting, each neighbourhood is an instance which is presented as a document (i.e. the collection of discussions on Yahoo! Answers) and each aspect is a binary label.

For a given aspect, labeled instances (i.e. neighbourhoods) can be used as supervisions to train a classifier. The classifier can then make predictions for the aspects of unlabeled neighbourhoods. To train a classifier without any labeled instances, we rely on methods that can make use of alternative cost-effective sources of supervision. In this section, we look at models for document classification in cases where labeled instances for an aspect are available and in cases where these labels are missing.

4.2.1 Classification

Let's assume we have a few data points $\{x^{(1)}, \dots, x^{(N)}\}$ and a set of corresponding output values $\{y^{(1)}, \dots, y^{(N)}\}$. Here, output values are categorical variables. In a binary classification task, there are two categories: one and zero, i.e. $y^{(i)} \in \{0, 1\}$. The task of binary classification for a new input point is to determine whether the value of its output value is zero or one. Hence, binary classification can be seen as a function $f : \mathcal{X} \rightarrow \{0, 1\}$. In the simplest case of classification, $x^{(i)}$ is a continuous value, i.e. $\mathcal{X} \in \mathbb{R}$.

Document Classification

Document classification is the task of assigning a document to one or more

classes or categories. In this case, \mathbf{x} is the numerical representation of a document and therefore $\mathcal{X} \in \mathbb{R}^D$. There are two general categories of classifiers for a document classification task: generative and discriminative. A generative model, such as Naive Bayes, defines the prior on the probability of classes $p(y)$ and the likelihood of data $p(\mathbf{x}|y)$. Bayes rule is then used to calculate the probability $p(y|\mathbf{x})$ as follows:

$$p(y|\mathbf{x}) = p(\mathbf{x}|y)p(y) \quad (4.1)$$

A discriminative classifier, on the other hand, directly determines the conditional probability $p(y|\mathbf{x})$ by discriminating amongst the different possible values of the class y , instead of computing the likelihood. Logistic regression is a discriminative classifier and is described in the following section.

4.2.2 Logistic Regression

A logistic regression classifier, which is also referred to as maximum entropy (MaxEnt) within the language processing community, specifies the probability of a binary output $y \in \{0, 1\}$ to be 1 given the vector representation \mathbf{x} of a document of dimension D as follows:

$$p(y = 1|\mathbf{x}) = \frac{1}{1 + \exp^{-(\theta_0 + \theta_1 \mathbf{x}[1] + \dots + \theta_D \mathbf{x}[D])}} \quad (4.2)$$

which can be written in matrix form as follows, where $\theta = \{\theta_0, \theta_1, \dots, \theta_D\}$ is the set of all the model parameters:

$$p(y = 1|\mathbf{x}) = \frac{1}{1 + \exp^{-\mathbf{x}^T \theta}} \quad (4.3)$$

Parameters of the model, θ , are calculated by minimising the negative log likelihood of the data. Therefore to estimate the parameters θ , the following objective function, i.e. the negative log likelihood, is minimised. N is the number of data points in the training set. The input vector for a data point is identified by

$\mathbf{x}^{(i)}$ and its output by $y^{(i)}$.

$$\mathcal{L} = - \sum_{i=1}^N [y^{(i)} \log(p(y^{(i)} = 1|\mathbf{x}^{(i)})) + (1 - y^{(i)}) \log(1 - p(y^{(i)} = 1|\mathbf{x}^{(i)}))] \quad (4.4)$$

4.2.3 Classification without Labeled Instances

Models that we have described so far are applicable when a few number of labeled instances are available for supervision. In predicting perceived characteristics of neighbourhoods, it is important to do so even if no labeled instances are available for an aspect of interest. In this section, we describe a model which is based on Generalised Expectation Criterion which requires no labeled instances for training.

Generalised Expectation Criterion for Feature Labeling

A generalised expectation (GE) criterion [106] is a term in a parameter estimation objective function that expresses some preferences about the values of a model expectations.

In machine learning, models need sufficient amount of data (labeled instances) for training; when not available, we can resort to human knowledge. This knowledge can be captured through labelling instances by experts which can be costly for some tasks. Ideally, we can inject cost-effective domain knowledge into a prediction model.

GE makes it easy for a human to directly express domain knowledge that can be used as cost-effective supervision. Therefore, training does not need to depend on the labeled instances only. For example, a human can make a statement such as the following: “*For an area to be good for **nightlife**, I expect words such as nightclub, dance, bar, music, and gig to be mentioned when people discuss the characteristics of the area*”.

The common approach for incorporating domain knowledge, without GE, is through selecting the structure of the model and choosing the features. Although model selection and feature selection are very important, they are very technical concepts which do not provide the most suitable way for a human with domain

knowledge and a machine learning expert to communicate.

Supervision for GE can come from different sources. As the example above shows, for a document classification task, the domain knowledge can be expressed in terms of a list of words that are in affinity with a label. For instance, terms “nightclub”, “dance”, “bar”, and “music” are in affinity with the label *Nightlife*. These terms are referred to as *labeled features* [106] and can be provided by annotators without the need for specific domain expertise.³ The domain knowledge can be incorporated into the model by specifying the probability that a document can be labeled with *Nightlife* if one of these terms (i.e. labeled features) appear in the document.

Below, we explain the GE criteria and how it can be used within a discriminative model such as MaxEnt. In this setting, we consider that no labeled instances are available and therefore U refers to a set of available unlabeled instances (i.e. a set of unlabeled documents representing neighbourhoods). Here, the only source of supervision is the labeled features. Let us assume that \mathbf{x} is a vector of input term features and y is a binary class label. The probability of the label y to be one is calculated by the probability $p_\theta(y = 1|\mathbf{x})$ using a MaxEnt model provided in Equation 4.3.

To estimate the parameters of the MaxEnt model, θ , we take the following approach. Assume that $f_w(\mathbf{x})$ is a function of an input vector \mathbf{x} that indicates whether the word w is present in the document represented by \mathbf{x} . Moreover, \tilde{p} is the empirical distribution of the unlabeled data U and \hat{p} is the reference distribution defined by human experts. The probability $\tilde{p}(y|f_w(\mathbf{x}))$ is then computed as below:

$$\tilde{p}(y = 1|f_w(\mathbf{x})) = E_U[E[p_\theta(y|\mathbf{x})]] \quad (4.5)$$

The parameters θ of the model p_θ are estimated such that the empirical distribution $\tilde{p}(y|f_w(\mathbf{x}))$ is close to the reference distribution $\hat{p}(y|f_w(\mathbf{x}))$ for a labeled feature w when the term w is present in \mathbf{x} . Here, KL divergence is used for the

³In the domain of neighbourhoods, labeling areas with aspect labels need knowledge of the city and its areas. However, providing a list of terms that are related to an aspect, i.e. labeled features, only need familiarity with the language.

distance measure between the two distributions. A GE term for one labeled feature can then be defined as follows. Note that one GE term is defined for each labeled feature in the objective function.

$$\sum_{y \in \{0,1\}} \hat{p}(y|f_w(\mathbf{x})) \log \frac{\hat{p}(y|f_w(\mathbf{x})=1)}{\tilde{p}(y|f_w(\mathbf{x})=1)} \quad (4.6)$$

In the following, we give an example of how the above model works. Assume that the classification task is to label each document with 0 or 1 with respect to the aspect *Nightlife*, given a set of labeled features including the term “dance”. Table 4.1 shows five documents which are represented by input vectors $\mathbf{x}_1 \dots \mathbf{x}_5$. The values in the column $f_{\text{dance}}(\mathbf{x})$ indicate whether each input vector contains the term “dance” or not. The column $p_\theta(y=1|\mathbf{x})$ shows the predicted probability of the label *Nightlife* to be one given the input vector and using the model presented in Equation 4.3.

Table 4.1: Values of the labeled feature “dance” for the label *Nightlife* and the probabilities of the predicted class under the model.

Input Vector	“dance”	
	$f_{\text{dance}}(\mathbf{x})$	$p_\theta(y=1 \mathbf{x})$
\mathbf{x}_1	1	0.3
\mathbf{x}_2	0	0.2
\mathbf{x}_3	1	0.9
\mathbf{x}_4	1	0.1
\mathbf{x}_5	0	0.7

We can now calculate the empirical label distribution on the set of unlabeled instances that contain the labeled feature “dance” as below:

$$\tilde{p}(y=1|f_{\text{dance}}(\mathbf{x})=1) = \frac{1}{3}(0.3 + 0.9 + 0.1) = 0.43$$

Let’s assume that the reference distribution $\hat{p}(y|f_{\text{dance}}(\mathbf{x})=1) = 0.70$. The parameters θ of the model should now be updated such that these two distributions are closer to each other. This model will assign larger weights to the labeled features and also to the terms that often co-occur with these terms.

4.3 Approach

In this section, we explain the scope and the domain entities of our models. We further describe the methods that we implement to answer the questions raised in this chapter. These methods are based on the models explained in Section 4.2.

4.3.1 Domain Entities and Concepts

Some of the entities that are used in this chapter are already introduced in Chapter 3. These entities are locations and documents. Here, we introduce the following entities and concepts:

Aspect: An aspect refers to a perceived characteristic of a neighbourhood. The value of an aspect for a neighbourhood is binary which indicates the presence or the lack of the aspect for that neighbourhood. Examples of such aspects are *Nightlife*, *Quiet*, *Posh* and *Multicultural*.

Labeled Feature: A labeled feature is a term that is in affinity with an aspect. For every aspect, we collect a list of labeled features from a group of annotators (explained further in section 4.4.2). We represent the list of labeled features for an aspect as:

$$F = \{w_1, w_2, \dots, w_K\}$$

where w_k is a term in the vocabulary and K is the number of labeled features for a given aspect.

4.3.2 Unit of Analysis

The unit of analysis in a prediction task is a neighbourhood. This is the same as in the previous chapter. A neighbourhood, as we have defined, is a location entity which is known by its name and its coordinates. Aspect labels are also provided for neighbourhoods through their names (e.g. Camden Town: *Nightlife*, Finchley: *Quiet*). Since the unit of analysis is the same for the aspect labels and the text documents, the need for unifying the units is eliminated.

4.3.3 Correlation Analysis

To investigate whether discussions on QA platform of Yahoo! Answers reflect the perceived characteristics of neighbourhoods, we first study whether significant and meaningful correlations exist between the language used in these discussions and a set of selected perceived characteristics. Correlations are calculated using the point-biserial correlation coefficient, r [107].⁴ The point-biserial correlation is used to measure the relationship between a binary variable and a continuous variable. Like other correlation coefficients, the point-biserial correlation coefficient varies between -1 and $+1$ with 0 implying no correlation. Each correlation coefficient has also an associated p-value. Similar to Chapter 3, we use the correlation coefficient between each term in the vocabulary and each of the selected aspects. We then correct the p-values for multiple tests using Bonferroni correction [98].

4.3.4 Prediction

As mentioned in the previous sections, we formulate the task of predicting an aspect of a neighbourhoods as a classification task. We use the text features of Yahoo! Answers discussions or the Twitter data as input vectors. This can be viewed as a document classification task. In a supervised learning setting, we need a reasonable amount of supervision to train a classifier. The supervision usually comes in the form of labels for entities. Here, the labels are aspects for the neighbourhoods.

To predict labels for the aspects where no labeled instances are available, we employ methods that incorporate cost-effective domain knowledge. The domain knowledge about an aspect is defined using a list of terms that are in affinity with the aspect. The terms are referred to as labeled features.

Therefore, to predict the aspects for neighbourhoods, we look at the following two general approaches: learning from labeled instances and learning from labeled features. Learning from both labeled instances and features is briefly dis-

⁴We use r to represent the correlation coefficients obtained using point-biserial correlation, as opposed to ρ which we have used in the previous chapter for Pearson correlation.

cussed in the Appendix A.3.

4.3.4.1 Learning from Labeled Instances

Learning a prediction model for aspects of locations using few labeled instances can be framed as a document classification task, where we learn a classifier that maps documents (locations) to classes (zero and one) for each given aspect. We choose this classifier to be a logistic regression. To void over-fitting due to the large number of parameters ($|\theta| = |V|$), we use L_2 regularisation. Therefore, the objective function in Equation 4.4 is updated using the following equation. The hyperparameter λ can be tuned using cross-validation:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N [y^{(i)} \log(p(y^{(i)} = 1|\mathbf{x}^{(i)})) + (1 - y^{(i)}) \log(1 - p(y^{(i)} = 1|\mathbf{x}^{(i)}))] + \lambda \|\theta\|^2 \quad (4.7)$$

Spatial Prediction

Aspects of neighbourhoods show spatial properties: neighbourhoods that are closer to each other are more likely to be similar in the aspects they have. We assume this relation to be non-linear. This means that the similarity diminishes rapidly as the distance between neighbourhoods grow. We address this property using the non-linear RBF basis functions described in the previous chapter (Equation 3.8). Unlike in a GP prediction model, the parameters of a MaxEnt model when using the values of RBF basis functions as features are highly interpretable. We have seen in the previous chapter that by using RBF basis functions, we can achieve comparable performances to a GP prediction model. Therefore, in this chapter, we use RBF basis functions for spatial prediction. To incorporate RBF basis functions into the logistic regression, we calculate the value of RBF function between every two neighbourhoods in our dataset. The features representing a neighbourhood will then include the text features and the RBF values between the neighbourhood and all the M neighbourhoods in our dataset, i.e. $\mathbf{x}' \in \mathbb{R}^{|V|+|M|}$. We then use this representation in the logistic regression model. The parameters of this model are estimated by minimising the loss function in Equation 4.7 where for each instance, we replace the representation \mathbf{x} with \mathbf{x}' .

4.3.4.2 Learning from Labeled Features

We leverage the domain knowledge into our prediction models by using labeled features instead of labeled instances. To make predictions using labeled features only, we use the following two methods: the frequency score method and the GE model for feature labeling.

Frequency Score

To provide a baseline for the GE model, we propose a frequency-based approach for the classification task. For a given aspect, the frequency score of a document is the sum of the number of times each of the labeled features have appeared in the document. Since the lengths of the documents for different neighbourhoods vary largely, we normalise the frequency scores by the lengths of the documents.

The frequency score of a document d with respect to an aspect is then calculated as below where F is the set of all the labeled features for the aspect:

$$\text{frequency score}(d) = \frac{\sum_{w \in F} \text{count}(w, d)}{|d|} \quad (4.8)$$

Finally, $p(y = 1|\mathbf{x})$ for a document is calculated by normalising its frequency score (across the documents). This is so that the value is between 0 and 1 and represents probability.

Generalised Expectation (GE) for Feature Labeling

GE for feature labeling has achieved great results in document classification tasks [106]. The GE model is described in detail in Section 4.2.3. Unlike the frequency score method which calculates the probability using only the provided labeled features, a discriminative model with GE criteria can recognise other terms in the vocabulary that are related to the aspect. These terms are those that often co-occur with the labeled features. The parameters of the discriminative model in Equation 4.4 is learned by minimising the following objective function. The objective function is composed of a GE term (Equation 4.6) for each labeled fea-

ture $w \in F$ (F is the set of all labeled features for an aspect) and the L_2 regularisation term as stated in the below equation. The regularisation term is added because the model is under-parameterized otherwise. This is because the number of labeled features is usually much smaller than the number of parameters in θ which is the size of the vocabulary.

$$\mathcal{L} = \sum_{w \in F} \text{KL}(\hat{p}(y|f_w(\mathbf{x}) = 1), \tilde{p}(y|f_w(\mathbf{x}) = 1)) + \lambda \|\theta\|^2 \quad (4.9)$$

4.3.4.3 Evaluation Metric

Similar to the previous chapter, we use a ranking metric for the evaluation of the classification tasks. However, since we are evaluating the classification results (as opposed to regression), we use AUC (Area Under the ROC Curve) and not correlation. This is a very natural decision as AUC is mostly used as the evaluation metric for binary classification tasks. The interpretability of AUC makes it appealing as a classification evaluation metric for ranking: given a random positive observation and a negative observation, the AUC calculates the proportion of the times that the model guesses their value (true/false) correctly. This means that a random system performs with an AUC equal to 0.5 and a perfect system will have an AUC equal to 1. Moreover, similar to predicting demographic attributes of neighbourhoods, in discovering aspects of neighbourhoods, we care about the relative value of an aspect rather than its absolute value. For instance, instead of predicting whether an area is safe or not, it might be sufficient to know whether an area is safer than other areas. Therefore, we report the performance of our prediction models using AUC as opposed to F1 or Accuracy.

4.4 Dataset

We use the documents from Yahoo! Answers and Twitter as explained in the previous chapter. In this section, we describe the procedure of collecting aspect labels and labeled features.

4.4.1 Aspects and Labeled Instances

Spareroom is a platform for searching and advertising room vacancies in the city of London. Apart from its room renting services, it provides aspects for some of the neighbourhoods of London through its location wizard.⁵ Aspects were extracted manually from the website. Altogether, there are 42 aspects and 130 neighbourhoods. Each neighbourhood can have one or more aspects and each aspect can belong to one or more neighbourhoods. Table 4.2 shows these aspects and the number of areas that are labeled for every aspect. As we can see, the number of labeled areas for each aspect varies from 1 to 45.

Table 4.2: Aspects provided by Spareroom and the number of areas that are labeled for each aspect.

Aspect	#Area	Aspect	#Area
Quiet	45	Underrated	11
Well-connected	43	Posh	10
Multicultural	39	Bars	8
Central	35	Cosy	8
Chilled	34	Modern	8
Safe	33	Picturesque	7
Eating out	29	Bohemian	6
Up and coming	27	Pubs	4
Cosmopolitan	23	Alternative	4
Shopping	23	Fun	4
Friendly	21	Commuterbelt	3
Lively	20	Young	3
Villagey	20	Urban	3
Fashionable	20	Historic	3
Leafy	19	Happening	3
Family friendly	18	Undiscovered	3
Market(s)	17	Suburban	3
Waterside	14	Gay	1
Cultured	14	Hilly	1
Nightlife	13	Quirky	1
Unpretentious	13	Studenty	1

As we can see from the table, some aspects cover very few areas. This can be because there are not many areas that have a specific characteristic (e.g. *Posh*). It can also be the case that some aspects are partially labeled. For instance, London probably has more than 4 areas that are known to be good for having *Pubs* or

⁵http://www.spareroom.co.uk/flatshare/where_to_live_wizard.pl

more than 1 *Hilly* area. The labeling of areas are therefore not perfect and we anticipate noise in the labels. This is unlike the demographics data that is the *precise* estimation of the population demographics for *all* the neighbourhoods of interest.

4.4.2 Labeled Features

To collect labeled features, we ask 10 participants to describe each given aspect using a list of words. Participants are graduate and undergraduate students that volunteered for the task. There are no limits on the number of labeled features that a participant can provide for an aspect. We then take the union of all the provided terms by all the participants for each aspect as the set of its labeled features.

4.5 Experiments

In this section, we describe the experimental set up.

4.5.1 Scope

Similar to the previous chapter, we focus on neighbourhoods of London. To investigate whether our methods can generalise beyond London, we also explore predicting aspects of areas of other cities across the world in Appendix A.4.

4.5.2 Aspects

We choose 8 aspects from Table 4.2 to carry our analysis on. We choose aspects that either have a high number of areas (*Quiet, Well Connected, Multicultural*) or a medium number of areas (*Eating Out, Shopping*) or very few areas (*Cultured, Nightlife, and Posh*). For each aspect, we take the provided areas as positive instances and all the other areas as negative instances. Note that in our experiments, we only consider areas that are covered in Spareroom dataset and not all the areas of London.

4.5.3 Labeled Features

We ask the participants in the feature labeling task to provide labeled features for the set of the 8 selected aspects. Table 4.3 shows the number of unique labeled

features for each aspect.

Table 4.3: The selected aspects and the number of unique labeled features provided by annotators collectively for each aspect.

Aspect	#Features
Quiet	32
Well Connected	42
Multicultural	53
Eating Out	67
Shopping	43
Cultured	43
Nightlife	47
Posh	21

4.5.4 Reference Distribution

When applying the GE model for predicting aspects using labeled features, we need to define a reference distribution of label values for each labeled feature. We choose the best distribution for all the labeled features using cross validation. This distribution assigns probabilities 0.7 and 0.3 for labels to be 1 and 0, respectively (e.g. $\hat{p}(y = 1 | f_{\text{dance}}(\mathbf{x})) = 0.7$ and $\hat{p}(y = 0 | f_{\text{dance}}(\mathbf{x})) = 0.3$ where y refers to the label for aspect *Nightlife*).

4.5.5 Evaluation Setup

In this section, we explain the evaluation set up of our experiments.

Labeled Instances When training a model using labeled instances, we take 80% of the labeled instances for training and take the remaining 20% for validation.⁶ This is done by randomly selecting the positive and the negative instances. We create up to 10 random folds. For each aspect, we report the mean and the standard deviation over these folds.

Labeled Features Since in learning from labeled features we do not need any labeled instances, we can report the prediction performances on the entire dataset.

⁶Note that unlike the previous chapter, we take 80% of the data for training, as opposed to 75%. This is because the number of available data samples is very small. To give our classification models a higher chance of learning patterns from the data, we increase the size of the training set to 80%.

However, for comparison purposes, we calculate the evaluation metric, AUC, on the test sets that are defined in the previous paragraph.

4.5.6 Implementation

To make predictions using the GE model, we use Mallet.⁷ The numerical method L-BFGS is used in this implementation to estimate the parameters of the model. For logistic regression, we use the Python library of scikit-learn.⁸

4.6 Results

In this section, we look at the results of our experiments. These experiments are designed to answer the questions that have been raised at the beginning of this chapter.

4.6.1 Correlation

In this section, we look at the correlations between the aspect labels and the term frequency features of Yahoo! Answers and Twitter data using their normalised tf-idf representations.

Number of Correlated Terms

Table 4.4 shows the number of significantly (p -value < 0.01) correlated terms from Yahoo! Answers and Twitter with each selected aspect. The number of significantly correlated terms from Yahoo! Answers is not very high for many aspects. There is only one correlated term from Yahoo! Answers with each *Quiet* and *Shopping* aspects. Yahoo! Answers has the highest number of correlated terms with the aspect *Posh* (i.e. 47). Twitter has no significantly correlated terms with aspects *Quiet*, *Eating Out* and *Multicultural*. But Twitter has many correlated terms with aspects *Cultured* (2143), *Posh* (121) and *Nightlife* (99). It is interesting to note that the correlation coefficients for all the correlated terms are very high ($r > 0.4$).

Semantic Relatedness

To qualitatively assess the correlations, we present some of the top correlated

⁷<http://mallet.cs.umass.edu/>

⁸<http://scikit-learn.org/>

Table 4.4: The number of significantly correlated terms (p-value < 0.01) from both **Yahoo! Answers** and **Twitter** with the selected aspects. “Y!A” is used in place of Yahoo! Answers due to the space limit. Examples of top correlated terms from both Yahoo! Answers and Twitter are provided in Table 4.5 and 4.6.

Attribute	Source	#significant	#> 0.4	#0 – 0.4	#r < 0
Quiet	Y! A	1	1	0	0
	Twitter	0	0	0	0
Well Connected	Y! A	8	8	0	0
	Twitter	10	10	0	0
Multicultural	Y! A	2	2	0	0
	Twitter	0	0	0	0
Eating Out	Y! A	4	4	0	0
	Twitter	0	0	0	0
Shopping	Y! A	1	1	0	0
	Twitter	3	3	0	0
Cultured	Y! A	3	3	0	0
	Twitter	2143	2143	0	0
Nightlife	Y! A	5	5	0	0
	Twitter	99	99	0	0
Posh	Y! A	47	47	0	0
	Twitter	121	121	0	0

terms with two aspects that have a high number of correlated terms with both Yahoo! Answers and Twitter. These aspects are *Well Connected* and *Posh*. For Twitter, we also show the correlated terms with the aspect *Cultured* since the number of correlated of terms is very high.

Table 4.5 shows some of the highest correlated terms from Yahoo! Answers. Note that there are only 8 correlated terms from Yahoo! Answers with the aspect *Well Connected*, many of which seem to be semantically related to this aspect. Examples are the terms “taxi”, “tube” and “line”. Terms that are correlated with the aspect *Posh* also seem very relevant to this aspect, e.g. terms “wealthiest”, “wealthy” and “exclusive”. The names of some of the posh areas of London⁹ (e.g. “kingsbridge” and “chelsea”) are also amongst the top correlated terms with the aspect *Posh*.

Table 4.6 shows some of the top correlated terms from Twitter with aspects *Posh*, *Well Connected*, and *Cultured*. Some of the top correlated terms that seem semantically relevant to the aspect *Well Connected* in this table are “rail”

⁹Note that names of areas are also included in the vocabulary.

Table 4.5: Top correlated terms from **Yahoo! Answers** with the selected aspects.

Posh		Well Connected	
Term	r	Term	r
conscious	0.70	door	0.46
patisserie	0.53	miss	0.45
knightsbridge	0.53	taxi	0.45
chelsea	0.53	bar	0.44
wealthiest	0.51	tube	0.43
merit	0.50	walk	0.42
exclusive	0.49	easily	0.42
eloquent	0.49	line	0.41
wealthy	0.48	–	0.40
congestion	0.48	–	0.39

and “transport”. For the aspect *Posh*, some of the related terms are “christiandior”, “chanelbag”, “poshwashlondon” and “elegance”. Twitter has the highest number of correlated terms with the aspect *Cultured*. Looking at the areas that are labeled with this aspect, we observe that *Cultured* refers to areas that are known for having theatres, art galleries and exhibitions. This is reflected in the correlated terms such as “arty”, “breathtaking”, “mindblowing” and perhaps “onceinlondon”.¹⁰

Table 4.6: Top correlated terms from **Twitter** with the selected aspects.

Posh		Well Connected		Cultured	
Term	r	Term	r	Term	r
elegance	0.56	medical	0.47	mindblowing	0.69
theroyalpark	0.55	upper	0.46	onceinlondon	0.68
accidentselfie	0.53	rail	0.45	arty	0.66
spotoftea	0.53	candidate	0.45	breathtaking	0.65
stylebymagazine	0.53	heathrowairport	0.45	michelle	0.65
christiandior	0.53	escalator	0.44	instagrame	0.63
saintlaurent	0.53	overcome	0.44	coffeesnob	0.63
chanelbag	0.51	rebuild	0.44	viewfromthetop	0.63
poshwashlondon	0.49	asset	0.43	viewpoint	0.62
irreplacable	0.48	transport	0.48	visitlondonofficial	0.62

As we have seen in the previous chapter, terms from Yahoo! Answers and Twitter offer different types of information about each aspect. Terms from Yahoo! Answers sometimes offer definitions for aspects. This is the case for the

¹⁰People may use this term (onceinlondon) when visiting London and what it culturally offers.

aspect *Posh* and the terms “wealthy” and “exclusive”. The names of areas in Yahoo! Answers data also seem to be strongly related to some aspects. For instance, the aspect *Posh* comes to mind when one thinks of the area “Chelsea”. Twitter terms are often related to activities or lifestyle of people. For example, the terms “chanelbag” and “christiandior” (related to aspect *Posh*) are the names of expensive clothing and accessorise brands which are associated with shopping. Moreover, Twitter terms are often compound words, something that cannot be seen in Yahoo! Answers. Examples of such terms are “poshwashlondon” for the aspect *Posh* and “mindblowing” and “viewfromthetop” for the aspect *Cultured*.

4.6.2 Prediction

In this section, we investigate how well we can predict aspects of neighbourhoods using Yahoo! Answers and Twitter data.

4.6.2.1 Learning from Labeled Instances

Here, we look at the performances of the classification tasks for the selected aspects when a number of labeled instances are available for training.

Representation

To obtain the best classification results, we experimented with different representations of text for both Yahoo! Answers and Twitter¹¹. The best performing representation for Yahoo! Answers data is the PMI over the context window. This is similar to what we observed in the previous chapter where the representations over the context window of the location performed better in prediction tasks compared to the representations defined over the entire QA document for the location. However, here, the number of times that a term co-occurs with a location name matters in knowing whether a location has an aspect. The best performing representation for the Twitter data is the binary representation.

Yahoo! Answers vs. Twitter

Prediction results for the selected aspects are shown in Table 4.7. These results are based on the best performing representations for Yahoo! Answers and Twitter

¹¹These representations are explained in Chapter 3.

data. The results of the best performing source have been highlighted in bold. For each aspect, two of the words with the highest model coefficients that are common amongst all the folds are displayed. As we can see, predictions using text features of Yahoo! Answers can reach an average AUC of 74%, an increase of 4% over the performance of text features of Twitter. Yahoo! Answers can outperform Twitter on most of the selected aspects. The only aspects in which we can reach higher performances using Twitter data are *Posh* and *Shopping*. Yahoo! Answers and Twitter perform the same when predicting the aspect *Well Connected*. Similar to the correlated terms, the terms with the highest coefficients provide some insight into the type of data that is available on two platforms of Yahoo! Answers and Twitter.

Table 4.7: Aspect prediction results in terms of AUC using **Twitter** and **Yahoo! Answers** data. Classifiers for aspects are trained using labeled instances. These results are based on the best performing representations of Yahoo! Answers (context PMI) and Twitter (binary) for these tasks. Two of the words with the highest coefficients that are common amongst most of the folds are displayed for each aspect.

Aspect	Yahoo! Answers		Twitter	
	AUC	Terms	AUC	Terms
Quiet	0.65 (0.11)	<i>south, suburb</i>	0.63 (0.12)	<i>dear, school</i>
Well Connected	0.80 (0.12)	<i>line, central</i>	0.80 (0.17)	<i>commence, latergram</i>
Multicultural	0.81 (0.08)	<i>music, market</i>	0.65 (0.12)	<i>lol, vibes</i>
Eating Out	0.65 (0.15)	<i>bite, place</i>	0.62 (0.15)	<i>sundayroast, tooth</i>
Shopping	0.70 (0.15)	<i>people, pay</i>	0.71 (0.22)	<i>airport, productive</i>
Cultured	0.80 (0.11)	<i>restaurant, bite</i>	0.65 (0.31)	<i>drinkwater, arty</i>
Nightlife	0.86 (0.10)	<i>tourist, expensive</i>	0.81 (0.10)	<i>margarita, timetoeat</i>
Posh	0.69 (0.27)	<i>bite, money</i>	0.84 (0.09)	<i>hairdresser, elegant</i>
Average	0.74 (0.16)		0.70 (0.19)	

Spatial Prediction

We assume that similar to the demographic attributes, aspects of neighbourhoods entail spatial smoothness. This means that areas that are closer to each other are more likely to have similar aspects. Table 4.8 shows the results of the spatial predictions. For comparison, we also perform the classification tasks using RBF values on coordinates information without text features (last column).

Table 4.8: Results of *spatial* prediction of aspects in terms of AUC using **Twitter** and **Yahoo! Answers** data when classifiers are trained using labeled instances. Upward arrows indicate an increase over the performance of each Yahoo! Answers and Twitter when spatial information is not incorporated into the features (Table 4.7).

Aspect	Yahoo! Answers	Twitter	Coordinates
Quiet	0.68 (0.10)	0.59 (0.10)	0.48 (0.11)
Well Connected	0.85 (0.09) ↑	0.88 (0.12) ↑	0.83 (0.14)
Multicultural	0.79 (0.09)	0.65 (0.14)	0.69 (0.16)
Eating Out	0.66 (0.14)	0.65 (0.14)	0.59 (0.17)
Shopping	0.69 (0.15)	0.75 (0.15)	0.66 (0.11)
Cultured	0.86 (0.13) ↑	0.67 (0.34)	0.78 (0.29)
Nightlife	0.93 (0.03) ↑	0.90 (0.05) ↑	0.87 (0.04)
Posh	0.86 (0.23) ↑	0.98 (0.04) ↑	0.98 (0.03)
Average	0.77 (0.15) ↑	0.74 (0.20) ↑	0.71 (0.21)

As the results show, using coordinates information on its own can result in an overall AUC of 71%. Adding distance information to text features improves the average performances of Yahoo! Answers and Twitter by 3% each.

The main improvements using the coordinates information and Yahoo! Answers text features are for the aspects *Posh* (17%), *Nightlife* (9%), *Cultured* (6%) and *Well Connected* (6%). Similar improvements can also be observed for Twitter. The main reason for this is that these aspects manifest spatial clustering as shown in Figure 4.1. The points on the map indicate neighbourhoods (or more precisely their centre points) that have been annotated positively for the given spect. For aspects *Posh*, *Nightlife*, *Well Connected* and *Cultured*, most of the positively annotated neighbourhoods are close to each other, forming one or more clusters. On the other hand, Figure 4.2 shows those aspects that do not present spatial clustering. As expected, spatial prediction does not improve the performance of the predictions on these aspects (e.g. *Eating Out* and *Shopping*).

Interestingly, the aspect *Posh* can be predicted using the coordinates information only with an AUC of 98%. This is because there is only one cluster of posh neighbourhoods as we can see from Figure 4.1. According to the coefficients of the prediction model, distances to Baker Street and Portobello Road are amongst the highest determinants of whether a neighbourhood is posh.

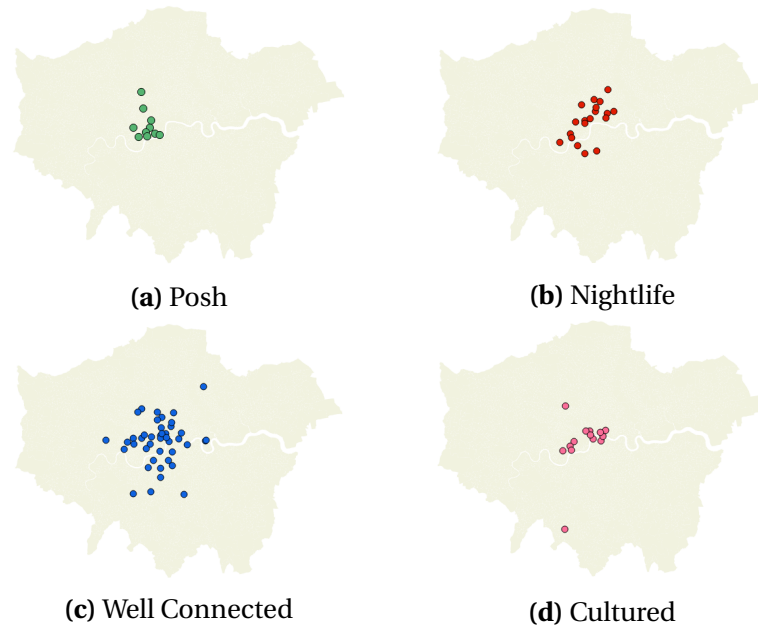


Figure 4.1: Aspects that present geographical clustering. The points on map indicate the centre points of neighbourhoods that have been annotated positively for each given aspect.

Cost Analysis

Labeling areas with their corresponding aspects is an expensive task. Ideally, we would like to make predictions for aspects of neighbourhoods in a cost-effective manner using very few labeled instances. Figure 4.3 shows the learning curves of the prediction performances in terms of AUC as the number of labeled instances increases using Yahoo! Answers data (these figures for Twitter data can be found in Appendix A, Figure 7.1). Red lines represent the performance plus and minus the standard deviation. Note that the scale of the horizontal axis represents the total number of available (positive) instances for the aspect. One negative instance is sampled per each positive instance.¹²

As we can see, the aspect *Well Connected* can reach an AUC of over 70% using very few labeled instances. However, aspects such as *Multicultural* and *Shopping* can reach an AUC of 60% (higher than the AUC of a random system, i.e. 50%) when we have around 10 positive and 10 negative instances. This means that

¹²This is because supposedly we do not know in advance the total number of positive and negative instances for an aspect in order to sample proportionally to the size of the positive and negative instances.

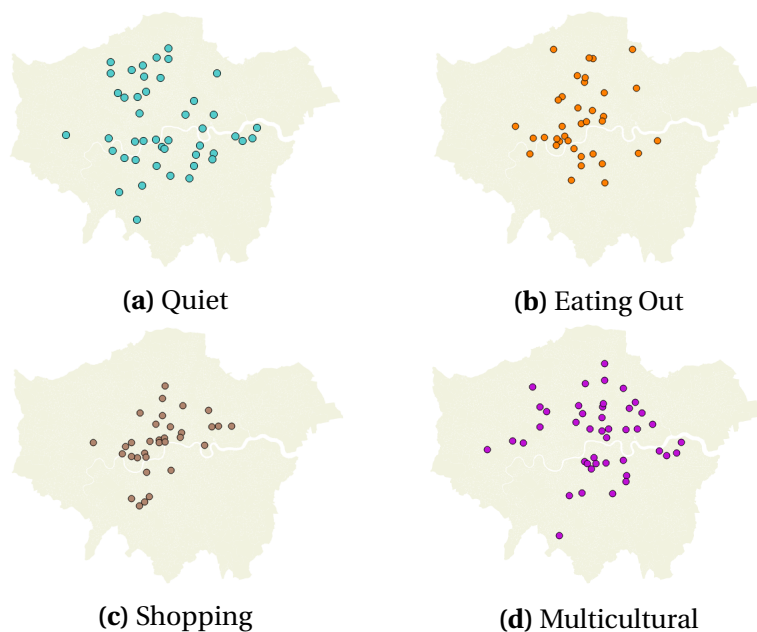


Figure 4.2: Aspects that do not present clustering in geographical space. The points on map indicate the centre points of neighbourhoods that have been annotated positively for each given aspect.

for these aspects, we need to label around 20 neighbourhoods to achieve a reasonable accuracy. Obtaining labels for around 20 areas can still be a costly task, if predictions are made for areas of a city where no knowledge about any of its areas is available. Therefore, in the next section, we look at the results of the prediction models that do not depend on labeled instances for training.

4.6.2.2 Learning From Labeled Features

In this section, we investigate predicting aspects using labeled features as supervision.¹³ Table 4.9 shows the performances of frequency score method and the GE model using Yahoo! Answers and Twitter data.

Frequency Score vs. GE

As we can see from Table 4.9, the GE model improves the overall prediction performance of Twitter data by 3%. A GE model assigns high weights to labeled features as well as the terms that co-occur often with the labeled features. The aspect *Posh* gains the highest performance boost (19%) when using the GE model

¹³Note that in this section, we report the performance on the test set only. We further present the performances of our experiments over the entire dataset in Appendix A (Section A.2.1).

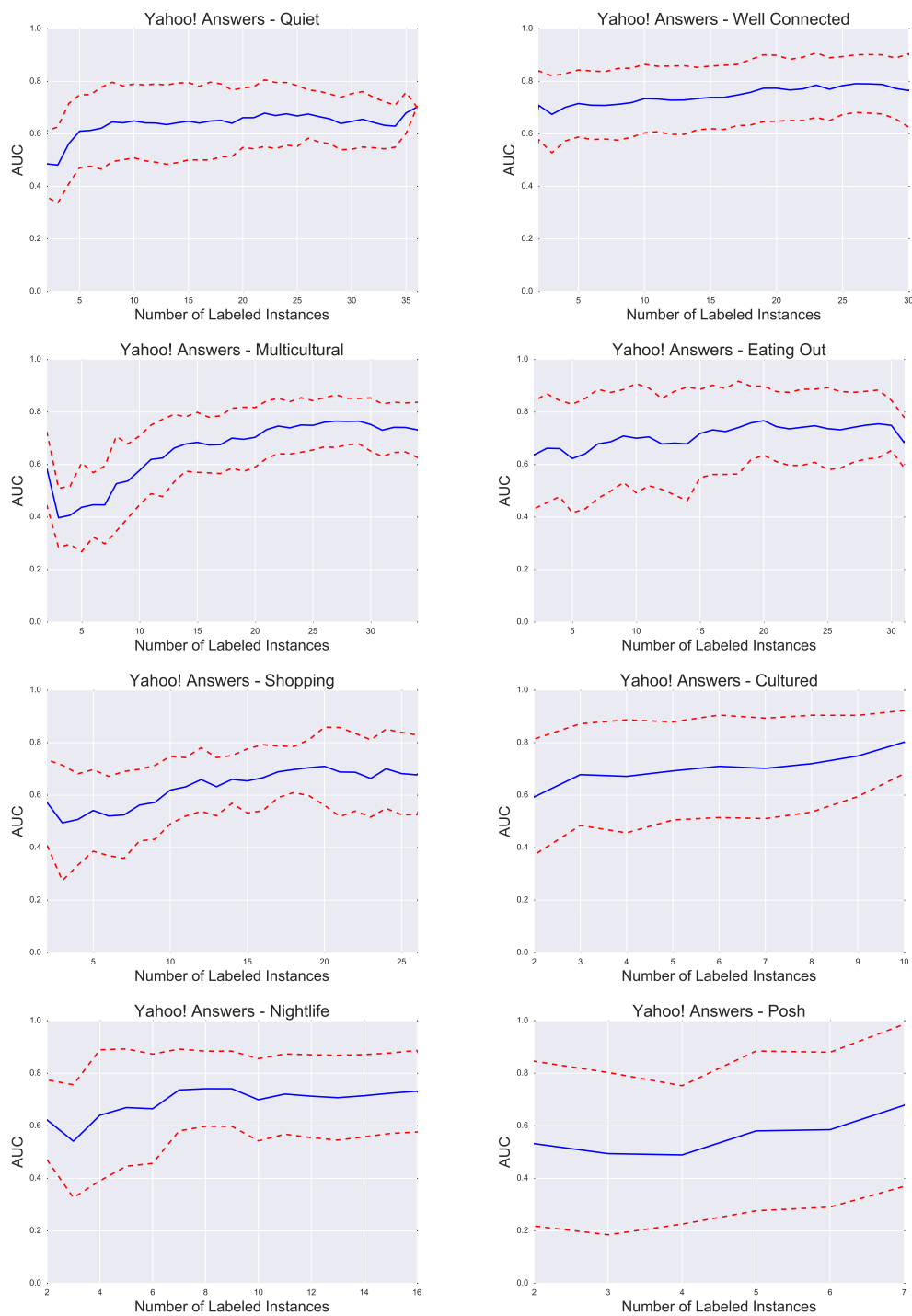


Figure 4.3: Learning curves of performances in terms of AUC for selected aspects using **Yahoo! Answers** text features when we train a model using labeled instances. Red lines represents the standard deviation of the performance (\pm std)

on Twitter data compared to the frequency score. Terms “*art*”, “*fashion*” and “*theatre*” are amongst the terms that are assigned high weights in the MaxEnt

Table 4.9: Prediction performances using labeled features applied on data from **Yahoo! Answers** and **Twitter** using the frequency score method and the GE model. The AUC is reported on the test set. Upward arrows indicate whether the GE model improves upon the performance of the frequency score method for the given aspect and the source.

Aspect	Yahoo! Answers Frequency	Twitter Frequency	Yahoo! Answers GE	Twitter GE
Quiet	0.39 (0.12)	0.31 (0.08)	0.43 (0.12) ↑	0.28 (0.07)
Well Connected	0.63 (0.14)	0.89 (0.09)	0.71 (0.08) ↑	0.89 (0.08)
Multicultural	0.57 (0.10)	0.42 (0.10)	0.53 (0.10)	0.41 (0.11)
Eating Out	0.64 (0.17)	0.68 (0.12)	0.60 (0.08)	0.71 (0.13) ↑
Shopping	0.71 (0.10)	0.74 (0.14)	0.51 (0.14)	0.73 (0.13)
Cultured	0.41 (0.34)	0.87 (0.14)	0.45 (0.33) ↑	0.89 (0.10) ↑
Nightlife	0.67 (0.15)	0.79 (0.09)	0.65 (0.21)	0.82 (0.07) ↑
Posh	0.60 (0.09)	0.66 (0.14)	0.67 (0.15) ↑	0.85 (0.12) ↑
Average	0.58 (0.15)	0.67 (0.11)	0.57 (0.15)	0.70 (0.10) ↑

model optimised through the GE objective function.

The GE model outperforms the frequency method on aspects such as *Well Connected* and *Posh* using Yahoo! Answers data. However, the performance of the GE model on Yahoo! Answers data drops for the aspect *Shopping*, compared to the frequency score method. This is not as we expected. To explain the drop in the performance, we look at the coefficients of the MaxEnt model. We observe that the terms “area”, “live”, “place”, and “london” are amongst the terms that are assigned high coefficients. These terms are very common and have perhaps appeared in the contexts of many neighbourhoods. Relying on these terms for identifying the areas that are good for *Shopping* can result in a poor performance as we have observed. Note that many of these terms are not amongst the labeled features for the aspect *Shopping*. However, a GE model assigns high weights to the terms that have co-occurred often with the labeled features.

Yahoo! Answers vs. Twitter

As we can see from Table 4.9, Twitter on average reaches an AUC higher than Yahoo! Answers by 9% when predicting aspects using labeled features. We hypothesise that the reason that GE does not perform well on Yahoo! Answers data is that often labeled features co-occur with common words (words such as “people” and “area”), as we have seen for the aspect *Shopping*. Giving high weights

to these common words through a GE objective function can result in assigning many neighbourhoods a high probability for the aspect which can lead to poor results. In Twitter, on the other hand, people often eliminate the common words and mainly mention the keywords.¹⁴ Hash tagging is a common example of a scenario where people use only keywords without connecting the words. An example of such a tweet is “*A night out in soho #dance #cocktail #music #fun*”. Another reason that GE can perform well on Twitter data is that in Twitter, people use many compound terms such as “sundayroast” or “poshwashlondon” that can be highly indicative of specific aspects. Such terms are hard to guess for annotators when they are providing a list of labeled features. However, these compound terms can co-occur with the labeled features. GE is therefore a suitable method for applying on Twitter data to identify the importance of these terms and consequently improving the classification results.

An interesting point to note is that the aspect *Quiet* is not predicted well using Yahoo! Answers or Twitter data. This aspect, however, can be predicted with an AUC higher than 70% using metadata such as the number of tweets or the number of users in a location (Table 7.3 in Appendix A.1). This is even higher than the performance achieved when this aspect is predicted using labeled instances and Yahoo! Answers text features. The reason for this can be because quiet areas are amongst the areas that are not known by many people to be discussed extensively on Yahoo! Answers. As for Twitter, it can be the case that users do not tweet their activities or observations when in quiet areas as much as when they are in busy areas. This can suggest that the language alone cannot be used to predict all the aspects of neighbourhoods. We show examples of two areas in Table 4.10 which are labeled negative (red) and positive (blue) for the aspect *Quiet*, respectively. For each of these two areas, we show examples of tweets that contain the term “quiet” which is a labeled feature for the aspect *Quiet*.

¹⁴The reason for this can be the strict length limit imposed on each tweet.

Table 4.10: Examples of neighbourhoods that contain the term “quiet” in their tweets and that are labeled negative and positive for the aspect *Quiet* respectively. The word “quiet” is a labeled feature for the aspect *Quiet*.

The West End, Quiet <i>quiet</i> #110
The quiet before the party! Loooonndon baby Perfect #snack at the beautifully quiet fernandezwells today. The streets are so quiet and the autumn chill is kept sketchlondon Gallery restaurant & bar Definitely love this place Quiet Sunday Went to a quiet little ramen place for lunch!
Blackheath, Quiet, <i>quiet</i> #0

As we can see from the table, there are no tweets in our dataset for the area of Blackheath that contain the term “quiet”. However, Blackheath is labeled as a quiet area in our dataset. On the other hand, there are 110 tweets containing the term “quiet” for The West End which is not considered to be a quiet area.

Cost Analysis

To further highlight the differences between the frequency score method and the GE model, in this section, we look at their performances as the number of labeled features increases. Figures for Yahoo! Answers results are shown in Appendix A (Section A.1). Figure 4.4 shows the learning curves obtained by applying the frequency method on Twitter data while Figure 4.5 shows the results obtained using GE. The figures show some of the selected aspects, especially those that are predicted well using the Twitter data.

From the figures, we see that unlike the frequency score, GE reaches the optimum performance right away. This is because while the frequency score method only considers the occurrences of the labeled features in a document, a GE model assigns high weights to labeled features as well as the terms that co-occur with the labeled features in many documents.

4.7 Discussion

In this chapter, we first created a dataset for some of the perceived characteristics of neighbourhoods of London. For a selected set of aspects, we investigated whether there are significant and meaningful correlations with the terms

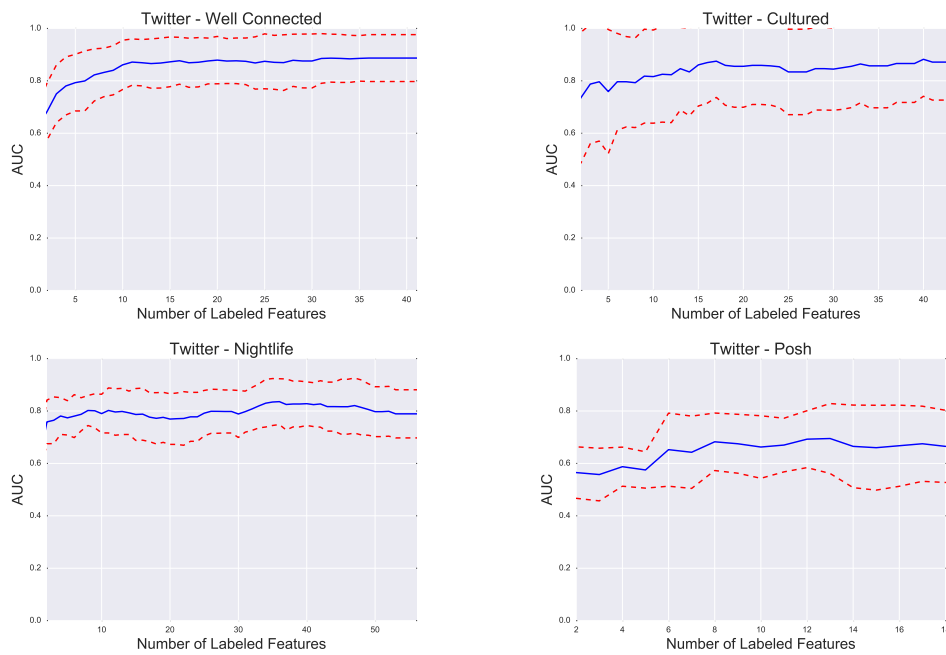


Figure 4.4: Learning curves of the classification performance in terms of AUC for the selected aspects using **Twitter** data and the frequency score method.

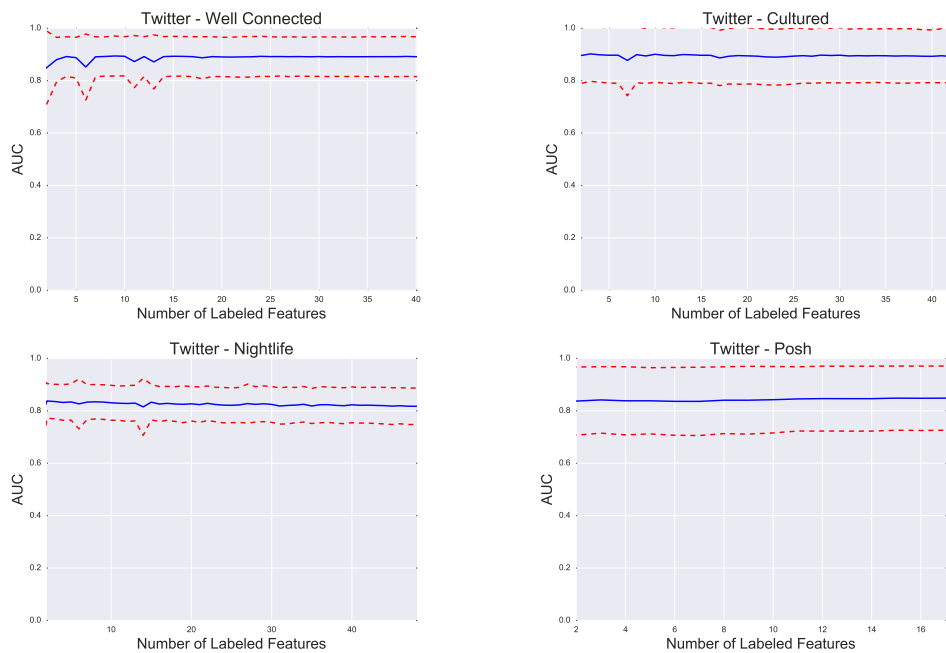


Figure 4.5: Learning curves of the classification performance in terms of AUC for the selected aspects using **Twitter** data and the GE model.

from Yahoo! Answers. We also investigated how well these aspects can be predicted using text features of Yahoo! Answers discussions. Since obtaining labels for aspects of neighbourhoods is an expensive task, we investigated whether we

can make accurate predictions using cost-effective sources of supervision. For this, we used labeled features which can be obtained through less costly domain knowledge. We can now answer the questions that we raised at the beginning of this chapter.

Q1: *Are there significant correlations between text features from Yahoo! Answers discussions and the perceived characteristics of neighbourhoods?*

A1: For many of the selected aspects, there are few terms in Yahoo! Answers discussions that have significant and strong correlations. The number of correlated terms can vary from one or two for aspects *Quiet* and *Multicultural* to 8 for the aspect *Well Connected* and 47 for the aspect *Posh*. These numbers are generally much lower in comparison with the number of correlated terms with the demographic attributes studied in the previous chapter. While Twitter has no significantly correlated terms with aspects *Quiet*, *Multicultural* and *Eating Out*, it has a high number of significantly correlated terms with aspects *Cultured* (2143), *Posh* (121) and *Nightlife* (99).

Q2: *How well can the perceived characteristics of neighbourhoods be predicted using text features from Yahoo! Answers discussions?*

A2: In the presence of a few labeled instances for aspects of interest, text features from Yahoo! Answers discussions about neighbourhoods can predict the aspects of neighbourhoods with an average AUC of 74%, a 4% increase over the text features from Twitter data. We showed that the prediction performances can be further improved by incorporating the non-linear spatial information of neighbourhoods. This is especially apparent for aspects such as *Posh* and *Nightlife* which present spatial clustering effects.

Q3: *Can we predict aspects of neighbourhoods in a cost-effective way using the discussions on Yahoo! Answers?*

A3: In this chapter, we experimented with appropriate methods for predicting aspects of neighbourhoods using labeled features instead of labeled instances.

Labeling features, unlike labeling instances, does not require costly expert knowledge. Results show that while the average prediction accuracy using the Yahoo! Answers data can only exceed a random system, the average accuracy can reach an AUC of 70% using the Twitter data.

Q4: *What are the limitations of using the discussions from Yahoo! Answers in predicting perceived characteristics of neighbourhoods?*

A4: In the previous chapter, we discussed the limitations of Yahoo! Answers data in providing coverage for all the neighbourhoods of all the cities. The same limitations exist when predicting *aspects* of neighbourhoods. Additionally, obtaining the supervision signal (labeling neighbourhoods with aspects) is costly. Even though the prediction performance using Yahoo! Answers data and costly labeled instances can reach an average AUC of 74%, the prediction performance can only reach an average AUC of 58% using the cost-effective labeled features.

In summary, the results of our experiments show that the discussions on Yahoo! Answers QA platform about neighbourhoods reflect many characteristics of neighbourhoods. Yahoo! Answers data can outperform Twitter in predicting many of these aspects using costly supervision. However, Twitter is superior to Yahoo! Answers in predicting such aspects using the cost-effective supervision. This suggests that Yahoo! Answers and Twitter can be complementary to each other in predicting aspects of neighbourhoods in cases where partially labeled instances are available and also in cases where no labeled instances are available. Overall, we conclude that the hypothesis that was raised at the beginning of this chapter holds for QA platform of Yahoo! Answers and the neighbourhoods of London.

Part II

Opinion Mining For Neighbourhoods

Chapter 5

Fine-grained Opinion Mining from Social Media Data

In the previous chapters, we showed that many characteristics of neighbourhoods can be predicted using the text features based on QA discussions. However, predicting an overall value for an attribute (objective characteristic) or an aspect (perceived characteristic) is not sufficient for providing users with a full picture of people's opinions about a neighbourhood. Therefore, in this chapter and the next chapter, we investigate extracting *fine-grained* opinion information for each neighbourhood. To provide a fine-grained summary, a very popular approach in opinion mining is to extract sentiments expressed towards different aspects of a given entity in a *small* unit of text such as a sentence. This is also referred to as fine-grained opinion mining or *aspect-based* sentiment analysis in literature [80, 54].

Currently, no datasets exist for fine-grained opinion mining in the domain of neighbourhoods. Existing work on fine-grained opinion mining has so far only utilised text from review data. Fine-grained opinion mining from generic social media text such as QA discussions has not been investigated. Discussions on QA platforms are not written with writing reviews in mind. Such text is noisier and less constrained in comparison with review-specific text. Therefore, extracting fine-grained opinion information from such text can present further challenges. In this chapter, we investigate challenges involved in using text from a QA plat-

form such as Yahoo! Answers for the task of aspect-based sentiment analysis for the domain of neighbourhoods. For this, we propose the following hypothesis:

***Hypothesis 3** Discussions on QA platforms about neighbourhoods can be used for extracting fine-grained opinion information for neighbourhoods.*

To investigate the above hypothesis, we raise related research questions in the following section.

5.1 Research Questions

In this part of thesis, we aim to use the text from QA discussions to extract fine-grained opinion information about neighbourhoods. For this, we look at the existing tasks in this field; the type of data they can process and their expected outputs. Existing tasks are based on datasets that utilise text from review-specific platforms. Hence, they may not be suitable for the less constrained text from QA discussions. Therefore, we investigate whether the above hypothesis holds by aiming to answer the following questions:

Q1: What are the shortcomings of the existing tasks for fine-grained opinion mining from QA discussions and how these shortcomings can be addressed?

Q2: What are the challenges in creating a dataset for fine-grained opinion mining from QA discussions?

In the following, we look at the existing approaches in the field of opinion mining for extracting the fine-grained information from opinionated text. We discuss their shortcomings for processing the text from QA discussions. To address these shortcomings, we propose a new task by combining the two existing tasks in this field. We also provide a formal definition of the task and propose suitable evaluation metrics. We then describe the steps taken to create a human-annotated dataset from QA discussions for the proposed task. Finally, we provide a description of the annotated dataset and compare it to an existing benchmark dataset for the task of fine-grained opinion mining.

5.2 Task

In this section, we examine the suitability of the existing tasks in the field of sentiment analysis for fine-grained opinion mining from QA data. We mainly look at two tasks that are the most relevant in terms of the granularity level for extracting opinion information.

5.2.1 Existing Tasks

Aspect-based sentiment analysis (ABSA) relates to the task of extracting fine-grained opinion information by identifying the polarity towards different aspects of an entity in the same unit of text, often a sentence. Take the below example (name of the area is underlined, aspect related terms are in bold):

*“St Johns Wood is a very **nice** area, it’s conveniently **located** but house **prices** can be ridiculously high”*

In ABSA task, the above sentence can be processed to extract the following information: a Positive sentiment in general (“a very nice area”), a Positive sentiment for the aspect *location* (“it’s conveniently located”) and a Negative sentiment for the aspect *price* (“house prices can be ridiculously high”). This task is the closest task in the field of sentiment analysis in terms of the level of the granularity of information we desire to extract from QA discussions for neighbourhoods. However, the existing datasets for this task are mostly based on the text from dedicated review platforms such as Yelp where it is assumed that only one entity is discussed in one review snippet and therefore in each sentence. In QA discussions, on the other hand, often more than one neighbourhood is discussed in the same sentence. Handling more than one entity is not considered in the ABSA task.

In *targeted* (a.k.a. target-dependent) sentiment analysis task [1, 2], we classify opinion sentiments towards a specific target entity, instead of the entire sentence. In this task, we extract only the *overall* sentiment for the entity. Current datasets for this task are based on Twitter data. Even though the task definition

does not put a limit on the number of entities that can be present in a tweet, existing corpora only contain annotations for a *single* target entity per each tweet. For example, in the sentence below, targeted sentiment analysis can identify a Negative sentiment towards Stockwell, despite other positive emotions being expressed in the sentence.

*“I love visiting my friends, even though they live in **gloomy** Stockwell”*

5.2.2 Targeted Aspect-Based Sentiment Analysis

The settings of both tasks of aspect-based and targeted sentiment analysis are limiting. There exist many scenarios in QA discussions in which sentiments towards different aspects of several neighbourhoods are expressed in the same unit of text. In such cases, identifying only the overall sentiment for an entity or identifying aspects and their related sentiment, irrelevant of the target entity is not sufficient. In particular, it is necessary to identify aspects that are discussed *for each neighbourhood* (target entity) together with their relevant sentiments. The following, is an example where sentiments towards different aspects of several neighbourhoods are expressed in one sentence (names of the areas is underlined, aspect related terms are in bold).

*“Other places to look at in South London are Streatham which has a good range of **shops and restaurants**, maybe a bit **far out** of central London but you get more for your **money**) and Brixton which has good **transport links** is **trendy** but can be a bit **edgy**.”*

The example above does not perfectly fit into the existing tasks in the field of sentiment analysis but resembles a big proportion of the sentences present in QA discussions. To make comparison to the existing aspect-based sentiment analysis task, take the following example from the restaurant dataset used by SemEval shared ABSA [54] task. *“The design of the **space** is good but the **service***

is horrid!". ABSA task can process the sentence and identify that a Positive sentiment towards the *ambiance* aspect is expressed. Moreover, a Negative sentiment is expressed towards the *service* aspect. In this example, it is assumed that both of these opinions are expressed about a single restaurant which is not mentioned explicitly. However, take the following *synthetic* example that ABSA is *not* addressing:

*"The design of the **space** is good in Boqueria but the **service** is horrid, on the other hand, the **staff** in Gremio are very friendly and the **food** is always delicious."*

In the above example, more than one restaurant is discussed and restaurants for which opinions are expressed, are explicitly mentioned. We call these "target entities". In the current ABSA task, we can only recognise that positive and negative opinions towards the aspect *service* are expressed. But we do not identify the target entity for each of these opinions (i.e. Gremio and Boqueria respectively).

To cater for such scenarios, we introduce a new task that subsumes the existing sub-fields of *targeted* and *aspect-based* sentiment analysis and makes fewer assumptions about the number of entities and aspects discussed in a sentence. We call this task *targeted aspect-based* sentiment analysis. Targeted aspect-based sentiment analysis allows extracting the target entities in an opinion as well as the aspects it expresses and the relevant sentiments.

5.2.3 Formal Definition

We formally define the task of targeted aspect-based sentiment analysis as follows: given a sentence, we provide a list of labels as tuples $\{(\ell_t, a_t, p_t)\}_{t=0}^T$, where p_t is the polarity expressed for the aspect a_t of location entity ℓ_t . Each sentence can have zero to T number of labels associated with it.

Within the existing aspect-based sentiment analysis task, to identify the correct sentiment for a given aspect, three sub-tasks are defined [68]: classifying the aspect, detecting the opinion target expression and classifying the sentiment. Detecting the opinion target expression is an optional *intermediary* task for iden-

tifying the relevant sentiment expressed for an aspect.

In this thesis, we by-pass this intermediary sub-task of identifying the aspect target expression and we focus on identifying the sentiment of an aspect with regard to a location.

To jointly identify an aspect and its related sentiment for a location, we introduce a new polarity class called “None”. None indicates that a sentence does not contain an opinion for the aspect a_t of location ℓ_t . Note that this is different than the sentiment class Neutral.¹ Therefore the overall task can be defined as a multi-class classification task for each (ℓ_t, a_t) pair. Multi-class refers to the sentiment classes such as Positive and Negative, plus None. To simplify the task, we make the assumption that no sentence has conflicting information about the same location and aspect pair.

Table 5.1 shows an example of an input sentence and its output labels. Note

Table 5.1: Example of an input sentence and the output labels.

Sentence	Labels
<u>Camden Town</u> is good for going out but I recommend <u>Kentish Town</u> for living as it's very quiet	(Camden Town,nightlife,Positive) (Camden Town,live,None) (Camden Town,quiet,None) ... (Kentish Town,live,Positive) (Kentish Town,quiet,Positive) (Kentish Town,nightlife,None) ...

that we create a tuple for each aspect that exists in the dataset. For those aspects where no opinion is expressed in the sentence, we add a tuple using the sentiment class None.

¹A neutral opinion discusses an aspect with a neutral polarity. For instance, in the sentence “*I went to a restaurant in Camden Town*”, a Neutral sentiment is expressed for the aspect *dining* of Camden Town. However, the sentence “*I went to Camden Town last night*” will have a sentiment label None with respect to the aspect *dining*.

5.2.4 Evaluation Metric

Similar to the aspect-based sentiment analysis task, we propose to evaluate the output of the targeted aspect-based sentiment analysis task using F_1 measure for aspect identification, and accuracy for sentiment detection [54]. F_1 score is often calculated with a threshold that is optimised on the validation set.

We also propose AUC (area under the ROC curve) metric for both aspect and sentiment classification tasks. This is because AUC does not rely on a threshold and captures the quality of the ranking of output scores.

5.3 Dataset

For the task of targeted aspect-based sentiment analysis for the domain of neighbourhoods, we use the existing data that we collected for neighbourhoods of London from the QA platform of Yahoo! Answers. In this section, we describe the properties of this data, annotation guidelines and procedure, and the statistics of the annotated dataset.

5.3.1 Preprocessing

A unit of analysis for the task of targeted aspect-based sentiment analysis is a sentence. This is also the case in many of the existing fine-grained opinion mining tasks and datasets. Sentences are separated using the sentence tokeniser of the NLTK library² in python. Due to the fact that we are dealing with social media data where not all the sentences are delimited by punctuation, it is possible to end up with many long sentences.

Table 5.2 shows the statistics of the number of unique QAs, the total number of sentences and sentences that contain the names of one or more locations.

Table 5.2: Statistics of the QA dataset

Number of unique QAs	4146
Number of sentences	93330
Number of sentences with location mentions	19975

²<http://www.nltk.org/>

Location Mentions

As we can see in Table 5.2, approximately 80% of the sentences do not contain a location name. Unlike the review data in which one can perhaps assume that all the sentences in a review express opinions for the same selected entity, in QA discussions, such an assumption cannot be made. Take as an example the below text snippet that contains a sentence that does not have a location mention (highlighted in bold):

*“... Balham has become a bit more up-market in recent years, and it's not too bad as a place. I'd certainly prefer it to Streatham and Brixton. **I would be a bit wary of wandering around there late at night, but it's no worse than many areas ...**”*

In the above example, the first sentence expresses opinions for the area of Balham. The second sentence talks about Streatham and Brixton with a reference to Balham. The last sentence discusses Balham but does not have an explicit mention of its name. Coreference resolution is needed to determine which location “there” refers to. Current coreference (co-ref) resolution systems are far from perfect even for high-quality newswire data. These systems are even less successful for the noisy and often ungrammatical social media data. We applied the Stanford co-ref system³ to our dataset but results were not satisfactory. Therefore in our dataset, we only keep sentences that have at least one location mention.

Number of Location Mentions

In QA discussions, often more than one location is discussed in the same sentence. Apart from neighbourhoods that we obtained from the GeoNames gazetteer,⁴ we also consider geographical regions such as North London, South London, South West London, etc. as location mentions since there are many sentences that discuss these geographical regions. Figure 5.1 shows the histogram of the number of location mentions per sentence. The top figure shows the his-

³<http://nlp.stanford.edu/software/dcoref.shtml>

⁴This is explained in Chapter 3

togram of the number of sentences that contain 10 or fewer location mentions. Note that sentences that have no location mentions are not included in this histogram. The figure at the bottom shows the number of sentences that mention over 10 location names. There are almost 12,000 sentences that contain the name of only one location and there are over 4,000 sentences that contain the names of two locations. Moreover, there are many sentences that contain more than 10 location names. The maximum number of location names mentioned in a sentence is 28. To simplify the problem, we only consider sentences that contain the names of *one* or *two* locations for annotations.

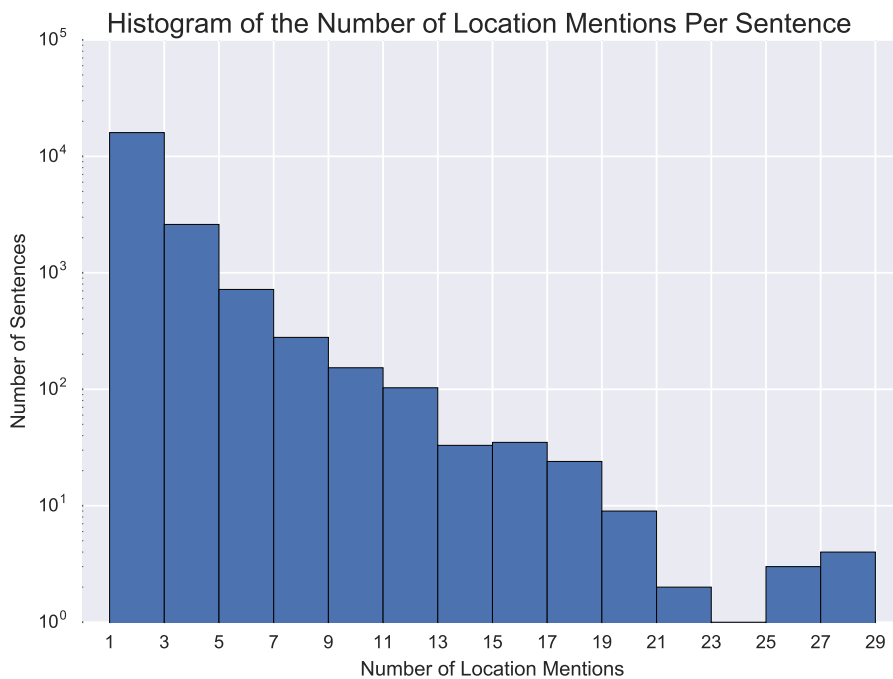


Figure 5.1: Histogram of the number of location mentions per sentence.

Sentence Length

Figure 5.2 shows the histograms of the length of sentences in our dataset in terms of the number of tokens. Separate histograms for short sentences (number of tokens less than or equal to 100) and long sentences (number of tokens more than 100) are provided. As the figure shows, even though the majority of sentences have a length of 100 or fewer tokens, there are many sentences that contain more than 100 tokens. We expect longer sentences to be harder to process when ex-

tracking opinion information.

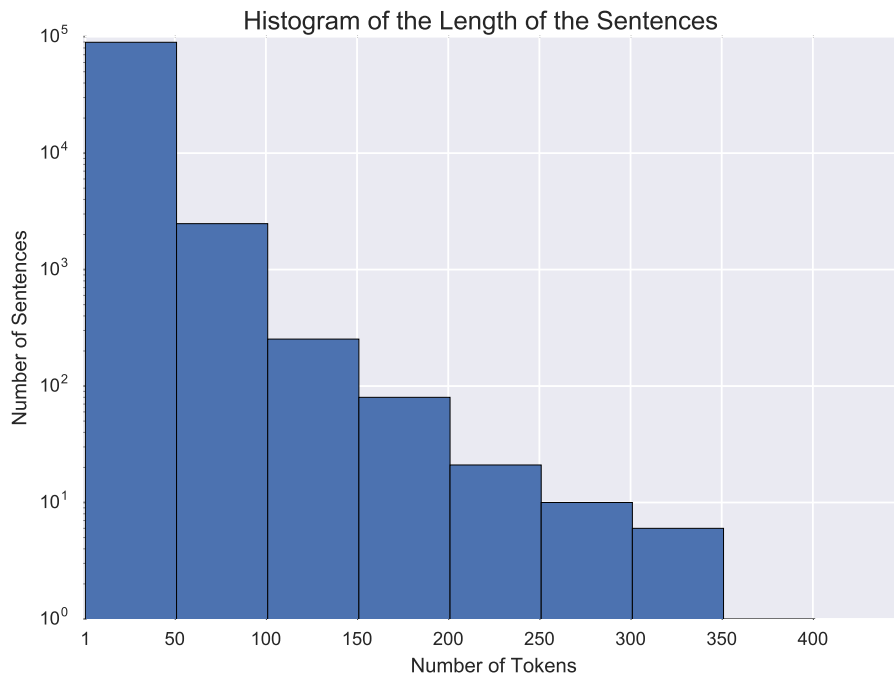


Figure 5.2: Histogram of the length of sentences in our dataset in terms of the number of tokens..

5.3.2 Annotation

In this section, we explain the elements of annotation that should be identified by annotators. Annotation guidelines are provided in Appendix B.⁵

Aspect Category

Similar to the existing annotation guidelines for the task of aspect-based sentiment analysis [68], a pre-defined list of aspects is provided for annotators to choose from. These aspects are *general*, *live*, *safety*, *price*, *quiet*, *dining*, *nightlife*, *transit-location*, *touristy*, *shopping*, *green-nature* and *multicultural*. The aspect *general* refers to an overall positive or negative opinion about a location. It is mainly used when a sentiment is expressed for a location but it cannot be related to a specific aspect. Examples of sentences expressing a *general* aspect are: “*I love Camden Town*”, “*Avoid Camden Town*”, “*I don’t recommend Camden Town*”

⁵More explanations and examples can be found here: <http://annotate-neighborhood.com/guidelines/start.html>

or “*Camden Town is such an amazing area*”. Adding an additional aspect of *misc* was considered. However, in the initial round of annotations, we realised that it had a negative effect on the decisiveness of annotators which could lead to a lower overall agreement.

Annotators were required to highlight a term in the sentence that is indicative of the aspect. We call this term *aspect term expression*. In the sentence “*House **prices** are very high in Hampstead but you can find **cheaper** options in Kentish Town*”, aspect term expressions are “prices” and “cheaper” that are related to the aspect *price*. The aspect term expression for the aspect *general* is always “null”.⁶

Aspect term expressions do not have to be predicted as part of the task output. This information, however, can be used for aspect or sentiment detection using lexicon-based methods. Aspect term expressions can also be used to infer statistics about the variations of the lexicon for each aspect.

Sentiment

For each selected aspect in a sentence, annotators were required to select a sentiment. Most work in this area considers three sentiment categories of Positive, Negative, and Neutral. In the initial round of annotations, we considered all the three sentiment categories. However, the sentiment Neutral was never chosen by any of the annotators. For the remaining of the annotation process we only provided the two sentiment categories of Positive and Negative. In the sentence “*House prices are very high in Hampstead but you can find cheaper options in Kentish Town*”, there is a Negative sentiment expressed for the aspect *price* of Hampstead but a Positive one for the aspect *price* of Kentish Town.

Target Entity

A target entity is a location name. Location names are by default highlighted in a sentence to be identified easily by annotators (we append stars to the beginning and the end of each location name e.g. ***camden town***). This is because some of the annotators are not familiar with the names of London neighbour-

⁶This is similar to the annotation guidelines for ABSA datasets.

hoods. Up to two target entities can be present in a sentence. Opinions can be expressed towards none, one, or both of the target entities present in the sentence. In the sentence “*House prices are very high in Hampstead but you can find cheaper options in Kentish Town*”, opinions are expressed towards two target location entities: Hampstead and Kentish Town.

Out of scope

We define the two following special labels for filtering out sentences that do not comply with our schema or sentences that annotators find difficult to annotate. Sentences marked with any of these labels are removed from the dataset.⁷

1. **Irrelevant:** When the identified name does not refer to a location entity. For example, in the sentence “*Notting Hill (1999) stars Julia Roberts and Hugh Grant use the characteristic features of the area as a backdrop to the action*”, Notting Hill refers to a movie and not the London area.
2. **Uncertain:** Annotators can label a sentence as “uncertain” when they find it hard to make a decision with regards to choosing a label for an annotation element. This happened mainly in the following cases: either when two contradicting sentiments are expressed for the same location and aspect (e.g. “*Like any other area, Camden Town has good and bad parts*”) or when the opinion is expressed for an area without an explicit mention of the name of the area in the sentence (e.g. “*It’s a very trendy area and not too far from King’s Cross*”).

5.3.3 Procedure

We use BRAT [108], a popular annotation tool for fine-grained opinion mining datasets [54, 109] and datasets for many other NLP tasks [110, 111]. Figure 5.3 shows examples of two annotated sentences in BRAT. Target locations have been highlighted for each opinion. Aspects and their relevant sentiments are connected to the target location by an edge. An aspect is identified using an aspect

⁷In an application, these sentences together with the sentences included in the dataset can be used to train a classifier to identify out of the scope sentences automatically.

term expression. For instance, in the top example, the term “restaurants” is the term indicating the aspect *dining*.

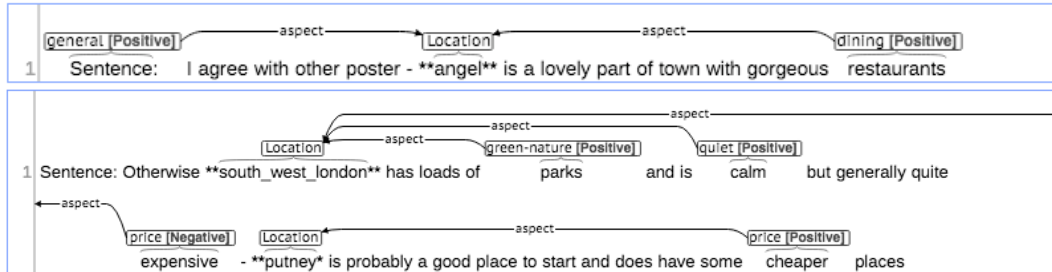


Figure 5.3: Examples of annotated sentences in BRAT.

Annotators

Three annotators were initially selected from a group of undergraduate volunteers. for the task. None of the annotators are experts in linguistics. Annotators started by reading the guidelines⁸ and the examples.⁹ Each annotator was required to annotate a small subset of the data. After each round of annotation, agreements between annotators were calculated and disagreements were discussed. This procedure continued until they reached a reasonable agreement over a small subset of randomly selected sentences. Afterward, all annotators annotated a randomly selected subset of the sentences which comprised 10% of the whole dataset. The pairwise agreement was calculated between each pair of annotators on that subset. The annotator with the highest inter-annotator agreement was selected to annotate all the remaining sentences in the dataset. This approach to annotation has been taken in the past [112, 113], especially when the annotation task is expensive. Note that even though using the best annotator for carrying the annotation of the entire dataset is cost-effective, the quality of the annotated dataset can suffer as a result of relying on a single annotator. We advise the readers to use three annotators for their entire set if their resources allow.

⁸Guidelines are provided in Appendix B

⁹Examples are provided at <http://annotate-neighborhood.com/guidelines/start.html>

Agreement Measure

Cohen's Kappa coefficient (κ) [114] is often used for measuring the pairwise agreement between each two annotators for the task of aspect-based sentiment analysis [115, 5] and other tasks [113].

5.3.4 SentiHood

In this section, we describe SentiHood,¹⁰ the dataset that was annotated for the task of fine-grained opinion mining for neighbourhoods based on the text from QA discussions. SentiHood currently consists of annotated sentences that contain one or two location entity mentions. SentiHood contains 5,215 sentences with 3,862 of sentences containing a single location and 1,353 sentences containing multiple (two) locations. Skipping the sentences that have more than two location mentions, annotators labeled approximately 10,000 sentences as *uncertain* or *irrelevant*, with the majority of them being labelled *uncertain*. Location entity names are masked by *location1* and *location2* in the dataset, so the task does not involve recognising and segmenting the location entity names.

Agreements

The Kappa coefficient is calculated over aspect-sentiment pairs per each location. The pairwise inter-annotator agreements in terms of κ are 0.73, 0.78 and 0.70, which is deemed of sufficient quality [76] for this task. It is worth mentioning that the agreements on different aspect categories varied, with some aspects having a higher agreement rate. Aspect *general* was amongst the hardest aspects to annotate. This was because annotators found the highest variations in how people express a *general* opinion for a neighbourhood.

Disagreements

The main disagreements between annotators occurred in detecting the aspects rather than detecting the sentiments or the target locations. For instance, some annotators associated the expression “residential area” with a Positive sentiment for the aspect *quiet* or *live*. Others, however, did not agree that “residential” im-

¹⁰SentiHood data can be obtained at <http://annotate-neighborhood.com/download/download.html>

plies quietness or desirable for living. In the case of a disagreement, the vote of the majority was considered as the correct annotation which was consequently adopted for the rest of the annotations.

Some ambiguity was also observed with respect to detecting whether an opinion exists for a particular target location in the sentence. This occurred mainly when a location is mentioned to describe a geographical relation with another location. For instance, the sentence “*Angel, in Islington, has many great restaurants for eating out*” expresses a Positive sentiment for the aspect *dining* of Angel which is within the borough of Islington. Some annotators suggested that the sentence also implies the same opinion for Islington. Another example is the sentence “*Stockwell, which is very close to Oval, is not a safe area*”. In this sentence, a Negative sentiment is expressed explicitly for the *safety* aspect of Stockwell. Some annotators argued that the fact that Oval is close to Stockwell might imply that it is also unsafe. However, at the end, all annotators agreed that in such cases no implicit assumptions should be made. Therefore, only explicit opinions were annotated.

5.3.4.1 Dataset Properties

In this section, we describe the properties of SentiHood and compare it with a well-known benchmark dataset for the task of fine-grained opinion mining. We refer to this dataset as SemEval dataset because it is created for the SemEval shared task of aspect-based sentiment analysis [38, 54] for the domain of restaurants. We make the comparisons to show that SentiHood is comparable to a benchmark dataset in terms of the size of the dataset, the number of annotated sentences per aspect and the distribution of sentiments per aspect. This makes SentiHood a plausible dataset to be used by the researchers in the opinion mining community.

In SentiHood, an opinion is a tuple consisting of a location, an aspect, and a sentiment. In SemEval, an opinion is a pair consisting of an aspect and a sentiment. Unlike in SentiHood, in the SemEval dataset, target entities are not annotated for an opinion. This is because all opinions in a review snippet are ex-

pressed for a single entity (i.e. a restaurant) and therefore the target entity is implicit. Moreover, aspect categories are hierarchical in the SemEval dataset. The combination of high-level aspects and low-level ones can be treated as one level aspects. High-level aspects are *restaurant, service, food, drinks, ambience* and *location*. Example of subcategories are: *general, quality, prices, style_options* and *miscellaneous*. Therefore the categories to be predicted can be considered as *restaurant#general, food#price, drinks#quality*, etc. Finally, there are three sentiment classes in the SemEval dataset: Positive, Negative, and Neutral. In SentiHood, we only consider Positive and Negative sentiments.

Statistics

SENTIHOOD: Table 5.3 shows some statistics for the sentences in the SentiHood dataset. This includes the total number of sentences, single location sentences (Single) and multi-location sentences (Multi). Moreover, for multi-location sentences, it shows the number of sentences that contain opinions about two locations which are in agreement in terms of the aspect and the sentiment. We call this category of sentences “Multi-Agree”. For example in the sentence “*House prices are very high in location1 and location2*”, a Negative sentiment is expressed for the aspect *price* of both *location1* and *location2*. Table 5.3 also shows the number of sentences that contain opinions on both locations with disagreeing sentiments which we call “Multi-Disagree”. An example of such a sentence is “*location1 is a very expensive area, for more affordable prices try location2*” where Negative and Positive sentiments are expressed for the aspect *price* of *location1* and *location2*, respectively. We expect classifying aspects with their correct sentiments in a sentence with disagreeing opinions to be harder in comparison with other categories of sentences.

Notice that since each sentence can contain one or more opinions, the total number of opinions in the each dataset is higher than the number of sentences.

#Sentences	#Aspects	#Single	#Multi	#Multi Agree	# Multi Disagree	#Opinions
5,215	12	3,862	1,353	1,762	508	5,920

Table 5.3: SentiHood dataset statistics.

#Sentences	#Aspects	#Single	#Agree Sentiments	#Disagree Sentiments	# Multi	#Opinions
2,000	13	2,000	365	32	-	2,797

Table 5.4: SemEval dataset statistics.

SEMEVAL: Table 5.4 shows the number of sentences and aspect categories in the SemEval dataset. It further shows the number of times more than one opinion exists in the same sentence for the same aspect. The sentiment of both opinions can be the same (Agreeing) or different (Disagreeing). An example of a sentence with the same sentiment expressed for the same aspect more than once is “*The anti-pasta was excellent, especially the calamari, as were the filling pasta mains”.* In this sentence, two positive opinions are expressed for the aspect *food#quality*.

SENTIHOOD VS. SEMEVAL: As the tables 5.3 and 5.4 show, SentiHood dataset contains a higher number of sentences than the SemEval dataset. Moreover, the number of opinions in SentiHood is 5,920 which is also higher than the number of opinions annotated in the SemEval dataset. Additionally, the number of agreeing and disagreeing opinions per sentence is much higher in SentiHood in comparison with the SemEval dataset. Overall, SentiHood provides a higher number of examples for a model to be trained on, on its different aspects and category of sentences.

Distribution of Opinions

In this section, we look at the distribution of the number of opinions in a sentence which is illustrated in Figure 5.4. The major difference between these two datasets is that unlike SemEval, SentiHood includes sentences that do not contain any opinions. We have included these sentences in SentiHood to provide

more examples of sentences that do not contain an opinion with respect to each of the aspects. The majority of the sentences in SentiHood contain one or two opinions. In SemEval, the majority of the sentences include one opinion. Moreover, SentiHood contains sentences that have up to 10 opinions.¹¹ The maximum number of opinions per sentence in SemEval is 8.

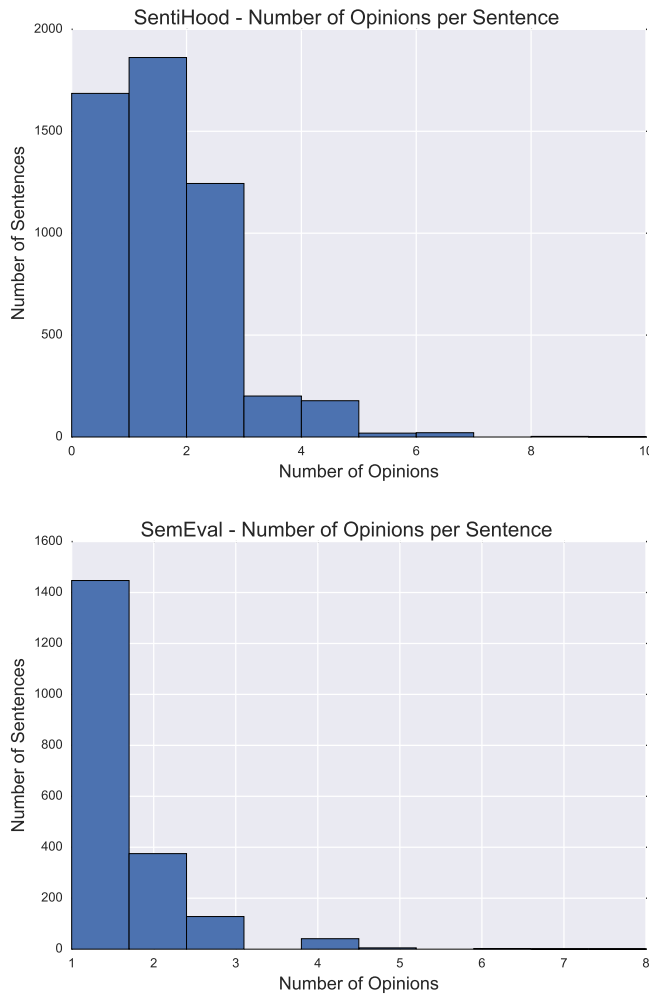


Figure 5.4: The number of opinions per sentence for SentiHood and SemEval datasets.

Number of Aspects and Distribution of Sentiments

Figure 5.5 shows the number of opinions that are labeled with each aspect for both datasets of SentiHood and SemEval. Moreover, the figure shows the dis-

¹¹An example of a sentence with a high number of opinions in SentiHood is “*location1 and location2 are both very safe, leafy London boroughs with a wide range of good restaurants, bars and clubs as well as excellent transport links to the centre*”

tribution of the sentiment classes for each aspect in both datasets. As the figure shows, the Positive sentiment is dominant for most of the aspects in both datasets. There are very few opinions that are labeled with the sentiment Neutral in the SemEval dataset, with many aspects not having any opinions with a Neutral sentiment (e.g. *service#general*, *restaurant#prices*, and *restaurant#general*). Overall, the number of labeled opinions for each aspect and the distribution of the sentiment classes per aspect in SentiHood dataset is comparable (even higher in number) to the benchmark dataset of SemEval.

Aspect Expressions

An aspect expression is a term or a span of text that is an indicator of the presence of an aspect. For instance, in the sentence “*location1 has great transport links to the centre of London*”, “transport” is the term indicating the aspect *transit-location*. Figure 5.6 shows the number of unique aspect expressions per each aspect in the SentiHood dataset. As we can see, the aspect *transit-location* has the highest number of aspect expressions. Aspect *general* has zero number of expressions because the aspect expression is always “null” for this aspect.¹² We expect that an aspect with a higher number of expressions (more lexical variation) to be more difficult to classify as a model needs to learn all the variations.

To provide examples of aspect expressions in SentiHood and their variations, we illustrate the word clouds of aspect expressions for the aspects *transit-location* and *dining* in Figure 5.7. These two aspects represent aspects with a high and a low number of distinct expressions. The larger words in the word clouds indicate the words that appear more often as aspect expressions in the corpus for each corresponding aspect.

5.4 Discussion

In this chapter, we first proposed a new task in the field of sentiment analysis which we call *targeted aspect-based* sentiment analysis. This task subsumes the existing sub-fields of *targeted* and *aspect-based* sentiment analysis and makes

¹²This is the same as the annotation procedure for the *general* aspect in the SemEval dataset.

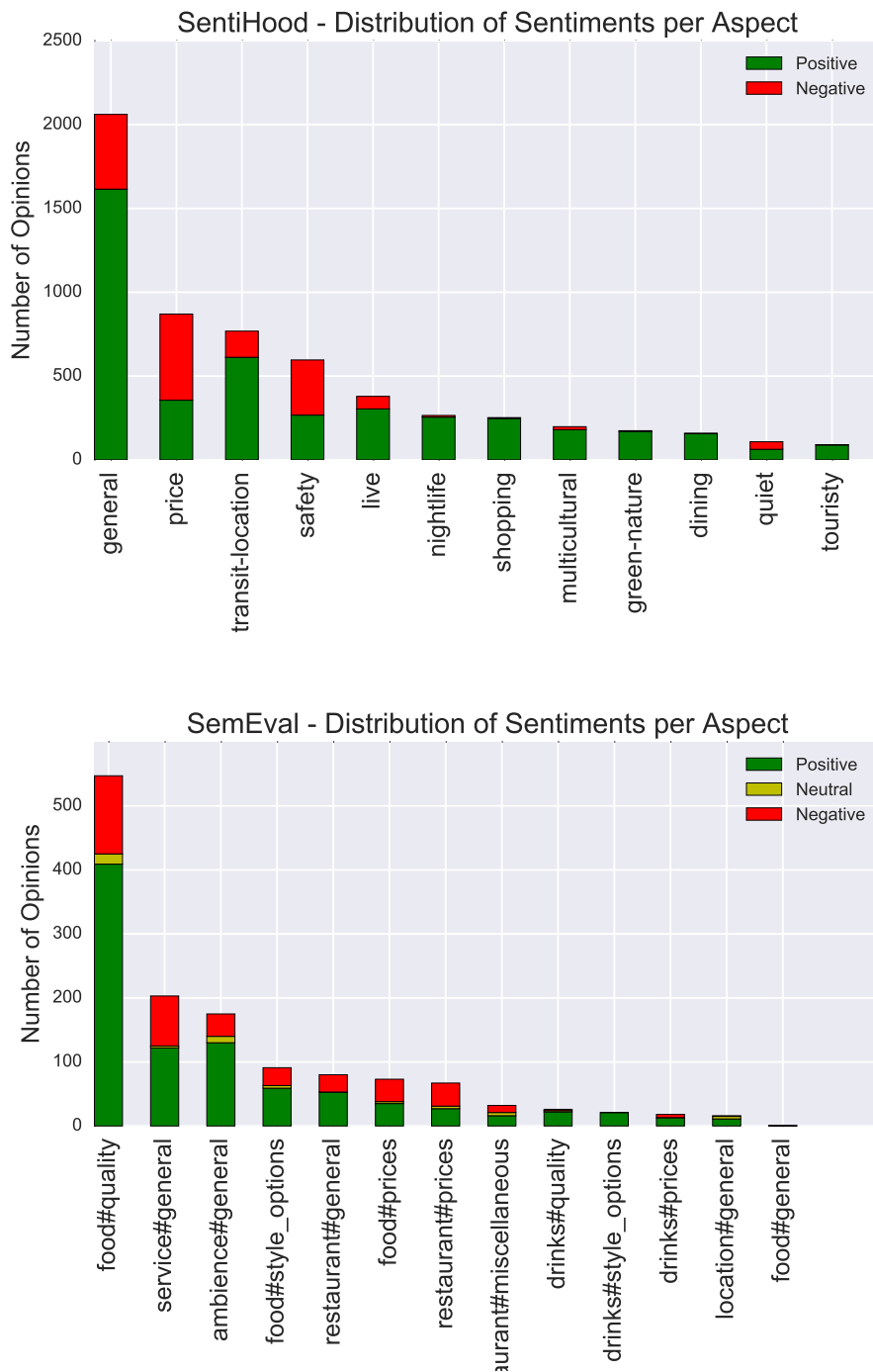


Figure 5.5: The number of labeled aspect categories and their sentiment distributions for SentiHood and SemEval datasets.

fewer assumptions about the number of entities and aspects discussed in a sentence. Not only this task is very relevant in practice in handling text from more generic sources, it also raises interesting modelling questions.

Q1: *What are the shortcomings of the existing tasks for fine-grained opinion mining from QA discussions and how these shortcomings can be addressed?*

A1: The current task of aspect-based sentiment analysis aims to extract fine-grained opinion information from opinionated text, often generated on review platforms. In aspect-based sentiment analysis task, we extract sentiments towards different aspects of a single entity in a unit of text which is often a sentence. Out of 19,975 sentences in QA discussions that contain location names, around 8,000 sentences discuss more than one neighbourhoods. To process this type of data, we introduce a new task that subsumes the two existing tasks in the field of sentiment analysis. *Targeted aspect-based* sentiment analysis allows for extracting sentiments towards aspects of more than one entity.

Q2: *What are the challenges in creating a dataset for fine-grained opinion mining from QA discussions?*

A2: The text from QA discussions is noisy and is not written with writing a review in mind. There are many sentences that describe side stories which do not contain opinions about neighbourhoods. Moreover, unlike review data, there are no implicit or explicit constraint on a text snippet to be about a single entity. Therefore, it is common for people to discuss multiple entities in the same unit of text. This means that the sentences that do not contain an entity name cannot be associated with a location entity by default. Co-reference resolution is needed to identify the reference entities in such cases. But the performance of the current coreference resolution systems are not optimal.

Therefore, we only annotate sentences that contain the name of at least one neighbourhood. From 93,330 sentences in our QA discussions, only 19,975 contain names of neighbourhoods. Moreover, due to the complexity and noise in sentences from QA discussions, 7,000 sentences were marked difficult to annotate. This shows that even though a dataset can be created with a reasonable inter-annotator agreement rate, there are many sentences that are not used due to the complexity and ambiguity of the text.

In summary, we show that by extending the existing tasks in the field of sentiment analysis, we can process the text from QA discussions to extract fine-grained opinion information. We further show that QA data can be used for creating a dataset for this task. Therefore, we can conclude that the hypothesis that we raised in the beginning of this chapter holds. To further demonstrate that we can extract fine-grained opinion information for neighbourhoods from QA data with a good accuracy, in the next chapter, we will raise and investigate a related hypothesis.

Chapter 6

Targeted Aspect-Based Sentiment

Analysis

In the previous chapter, we introduced the task of targeted aspect-based sentiment analysis. We also created SentiHood, an annotated dataset for this task based on the text from QA discussions for neighbourhoods. In this chapter, we aim to address this task based on SentiHood. Targeted aspect-based sentiment analysis task can be formulated as a classification task where the objective is to identify the correct sentiment class that is expressed for each aspect of each location entity in a sentence. What defines the success of a classification model is the suitability of the representations of sentences that are fed into the classifier. Here, we mainly focus on two general classes of representations which we refer to as the bag of n-grams and the sequential representations.

A bag of n-grams representation of a sentence is based on the isolated parts of the sentence. These representations are not usually capable of embedding the entire sequence of words in a sentence. A bag of n-grams representation is usually *defined* using a set of different semantic (e.g. n-grams, membership of words in the lexicon classes, etc.) and syntactic features (i.e. part-of-speech tags and syntactic relations between words, etc.). These representations, so far, have resulted in great performances [70, 116] in fine-grained opinion mining tasks. Sequential representations, on the other hand, are *learned* using sequential models such as recurrent neural networks. These representations have been successful

in many NLP tasks [87, 88] in the past. Sequential representations, in theory, can capture the full sequential nature of a sentence. This is important because the meaning of a sentence often can only be fully captured by considering the order of all its words. However, models such as neural networks often need a large number of training samples to learn good representations. In this chapter, we examine the suitability of these two classes of representations for addressing the task of targeted aspect-based sentiment analysis applied on the SentiHood dataset. We raise the following hypothesis.

Hypothesis 4 *Sequential representations can perform better than bag of n-grams in extracting opinion information in a targeted aspect-based sentiment analysis task.*

To investigate this hypothesis, we raise related research questions in the next section.

6.1 Research Questions

We expect a suitable representation to embed all the necessary information needed to classify the correct sentiments expressed towards different aspects of location entities in a sentence. As we have seen in the previous chapter, SentiHood contains different categories of sentences. These categories include sentences with a single location entity and categories with multiple location entities. Sentences with multiple location entities can be further divided into sentences where for a given aspect, the sentiments expressed towards different location entities are the same or different. A good representation should be able to address all categories of sentences.

Sequential representations obtained using models such as recurrent neural networks often need a large number of training samples in order to achieve a good performance. To fully investigate the capabilities of different classes of representations, we explore generating more samples. Instead of relying on expensive human annotation, we look at data augmentation. Using data augmen-

tation, we can generate training samples with more lexical and syntactic variations. This can lead to models that can generalise better on unseen data.

Therefore, this chapter is driven by the following questions:

Q1: Are sequential representations superior to the traditional bag of n-grams representations for addressing the task of targeted aspect-based sentiment analysis on SentiHood dataset?

Q2: Which type of sentences are more suitable to be addressed using sequential representations compared to the bag of n-grams representations?

Q3: Can generating more training examples through data augmentation improve the performances of representations, especially the sequential representations?

The rest of this chapter is structured as follows. We first give a brief description of the technical background. The reader can skip this section if already familiar with part-of-speech tagging, word embeddings, neural networks, recurrent neural networks and long short term memory networks (LSTMs). We then define the model for the prediction task. Afterwards, we propose an extensive list of representations based on bag of n-grams and sequential representations. This is followed by experiments and discussion of the results. Further, we investigate the performances of different classes of representations using a synthetic evaluation set which contains different types of sentences. Finally, we look at data augmentation and study the effect it has on the performances of our representations.

6.2 Technical Background

In this section, we provide a technical background to the methods we use in this chapter. We first describe part-of-speech tagging that can be used in the representation of a sentence. We then provide a description of word embeddings. For sequential representations of a sentence, we look at the architecture of recurrent neural networks, first explaining what a simple one-layer neural network entails.

This is followed by a description of LSTMs and bidirectional LSTMs.

6.2.1 Part-of-Speech Information

Part-of-speech (POS) tags provide information about syntactic functions of words in a sentence. POS information can be used alone or in combination with word n-grams to represent a sentence in vector space. Each word in a sentence is assigned a part-of-speech (POS) category based on its syntactic role. Words that have similar syntactic roles in a sentence are assigned the same POS category. In the English language, the main POS categories are nouns, verbs, adjectives, and adverbs. Figure 6.1 shows the POS categories for each word in the sentence “*Location1 is very nice*” where NNP indicates the category of proper noun, VBZ verb (3rd person singular present), RB adverb and JJ adjective.

Location1 (NNP) is (VBZ) very (RB) nice (JJ)

Figure 6.1: POS categories of words in a sentence.

6.2.2 Word Embeddings

Traditionally, in natural language processing systems, words are represented as one-hot vectors and are treated as discrete stand-alone symbols. These encodings do not provide any information regarding the relationships between different words. This means that a model cannot use what it has learned about a word, e.g. “cat” when it is faced with another similar word, e.g. “dog” (the fact that they are both animals, pets, etc.). An example of a one-hot representation of words “cat” and “dog” is illustrated in Figure 6.2.

Representing a word as a one-hot vector causes data sparsity which consequently means that we might require more data to train a statistical model successfully. We can overcome this issue by using dense vector representations. Vector space models (VSMs) represent words in a continuous vector space. Words that are semantically similar are closer to each other in this space. This is illustrated in Figure 6.3.

These VSM approaches are usually inspired by the Distributional Hypoth-

cat	dog
0	0
0	1
.	.
.	.
1	0
.	.
.	.
0	0
0	0

Figure 6.2: Examples of how “dog” and “cat” can be represented using one-hot vector representation.

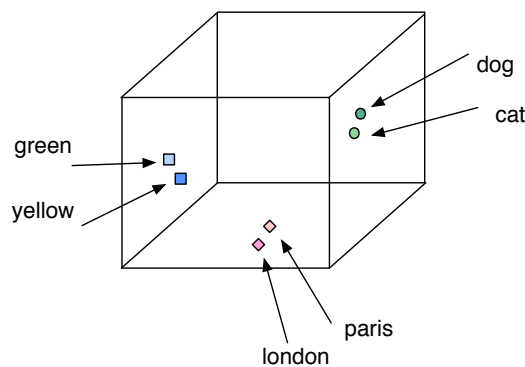


Figure 6.3: Word embeddings in a 3 dimensional space. Words that are semantically similar are closer to each other in this space.

esis. This hypothesis states that the words that appear in the same context are semantically similar.¹ Approaches based on this principle can be divided into two general categories: count-based methods such as Point-wise Mutual Information (PMI) and Latent Semantic Analysis (LSA), and predictive methods such as neural language models.

Count-based methods create a vector of the count statistics of co-occurrences of a word with other words in a large text corpus. These vectors are then projected into lower dimension dense vectors. Predictive models, on the other hand, aim to predict a word given its neighbouring words. In this setting, low dimensional dense embedding vectors are parameters of the predictive model.

¹Reflects the idea of John Rupert Firth that quoted “You shall know a word by the company it keeps”

Word2vec [117] is a predictive model that learns word embeddings from text using a neural network. Embeddings obtained by this model have been used successfully in many recent NLP tasks especially as inputs to many neural models. These embeddings can then be used as they are or they can be tuned further for the downstream task.

6.2.3 Neural Networks

A one-layer neural network consist of an input vector \mathbf{x} of dimension d , a hidden layer \mathbf{h} of dimension k and an output o . Vector \mathbf{x} can be a sparse representation of a sentence s . The hidden layer is calculated as below where $\theta = W$ ($W \in \mathbb{R}^{k \times d}$) is the set of the parameters of the model:

$$\mathbf{h} = g(W\mathbf{x}) \quad (6.1)$$

Here, g is a non-linear function, often a sigmoid that is computed element-wise. The output can be identical to the hidden layer \mathbf{h} or a linear or non-linear projection of the hidden layer. A one-layer neural network is depicted in Figure 6.4.

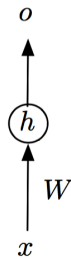


Figure 6.4: The architecture of a one-layer neural network.

We can take \mathbf{h} as a dense lower-dimensional representation, \mathbf{e} , of the sentence s . Numerical optimisation methods such as stochastic gradient descent can be used to find the optimal parameters of the model, θ , based on a loss function such as cross entropy. These methods rely on the gradient of the loss function to be available. Backpropagation [118] is used to calculate the gradient of the loss function with respect to all the parameters in the network.

6.2.4 Recurrent Neural Networks

A recurrent neural network or an RNN is a special type of a neural network with a feed-back connection which allows sequences of arbitrary lengths to be processed. This makes RNNs powerful for learning representations of sequences [119] such as a sentence. The architecture of an RNN is shown in Figure 6.5. Let us assume that the sentence s is a sequence of words, $\{w_1 \dots w_n\}$ and $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ is a sequence of vectors corresponding to the embeddings of the words in s where $\mathbf{x}_i \in \mathbb{R}^d$. An output representation for step t , \mathbf{h}_t , is obtained using an input embedding at step t , \mathbf{x}_t , and the representation of the previous step, \mathbf{h}_{t-1} , as shown in Equation 6.2. In this equation, g is often a non-linear function such as sigmoid.

$$\mathbf{h}_t = g(W_{xh}\mathbf{x}_t + W_{hh}\mathbf{h}_{t-1}) \quad (6.2)$$

Parameters of the RNN model are $\theta = \{\mathbf{h}_0, W_{xh}, W_{hh}\}$ where $\mathbf{h}_0 \in \mathbb{R}^k$, $W_{xh} \in \mathbb{R}^{k \times d}$ and $W_{hh} \in \mathbb{R}^{k \times k}$. The embedding in the last step, \mathbf{h}_n , can be taken as the representation, \mathbf{e} , of the sentence s . In Figure 6.5, we see an abstract view of an RNN on the left, where it has a feed-back connection. On the right, we unroll the network over time. Note that parameters W_{xh} and W_{hh} are the same over all the time-steps.

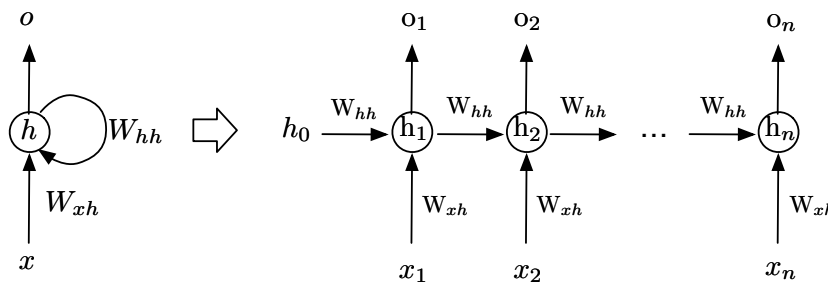


Figure 6.5: An RNN architecture.

Training an RNN is similar to training a traditional neural network. Back-propagation is used to compute the gradient of the loss function with respect to all the parameters, θ , in the network. The gradient at each output not only depends on the current time step, but also on the previous time steps. This is

referred to as Backpropagation Through Time (BPTT) [120]. It is difficult to learn long-range dependencies (dependencies between steps that are far apart) in a sequence with a standard RNN that is trained using BPTT. This is because of a problem called vanishing or exploding gradient.

6.2.5 Long-Short Term Memory Networks

Long-short term memory networks (LSTMs) [85] are variations of RNNs that are specifically designed to deal with the issue of vanishing (exploding) gradient. A long-short term memory (LSTM) network contains a memory cell. The memory cell can store information for a long period of time. This makes an LSTM capable of capturing the long-range dependencies in a sequence. An LSTM also has three types of gates. These gates control the flow of information into and out of the memory cell: input gate — \mathbf{i} , output gate — \mathbf{o} and forget gate — \mathbf{f} . These gates provide a mechanism for optionally letting information through to the memory or clearing the information from the memory. Figure 6.6 [121] shows an LSTM cell at time t . Given an input vector \mathbf{x}_t at the time step t , the previous hidden vector \mathbf{h}_{t-1} and the cell state \mathbf{m}_{t-1} , an LSTM with a hidden size of dimension k computes the next hidden vector \mathbf{h}_t and the cell state \mathbf{m}_t as follows.

$$\mathbf{i}_t = \sigma(W_1\mathbf{x}_t + W_2\mathbf{h}_{t-1}) \quad (6.3) \quad \mathbf{f}_t = \sigma(W_3\mathbf{x}_t + W_4\mathbf{h}_{t-1}) \quad (6.4)$$

$$\mathbf{o}_t = \sigma(W_5\mathbf{x}_t + W_6\mathbf{h}_{t-1}) \quad (6.5) \quad \hat{\mathbf{c}}_t = \tanh(W_7\mathbf{x}_t + W_8\mathbf{h}_{t-1}) \quad (6.6)$$

$$\mathbf{m}_t = \mathbf{m}_{t-1} \odot \mathbf{f}_t + \mathbf{i}_t \odot \hat{\mathbf{c}}_t \quad (6.7) \quad \mathbf{h}_t = \mathbf{m}_t \odot \mathbf{o}_t \quad (6.8)$$

The operator \odot denotes element-wise multiplication. Parameters of the model are $\theta = \{W_1, \dots, W_8, \mathbf{h}_0\}$ where $W_1, W_3, W_5, W_7 \in \mathbb{R}^{k \times d}$ and $W_2, W_4, W_6, W_8 \in \mathbb{R}^{k \times k}$. Further, we have $\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t, \hat{\mathbf{c}}_t, \mathbf{m}_t \in \mathbb{R}^k$. We can take \mathbf{h}_n as the representation, \mathbf{e} , of the sentence s .

6.2.6 Bidirectional LSTMs

At each time step, a single-direction LSTM can only use the contextual information of the previous time steps. Bidirectional LSTMs can generate richer representations by incorporating both the previous and the future context at each time step. They process a sequence in two directions, forward and backward, and gen-

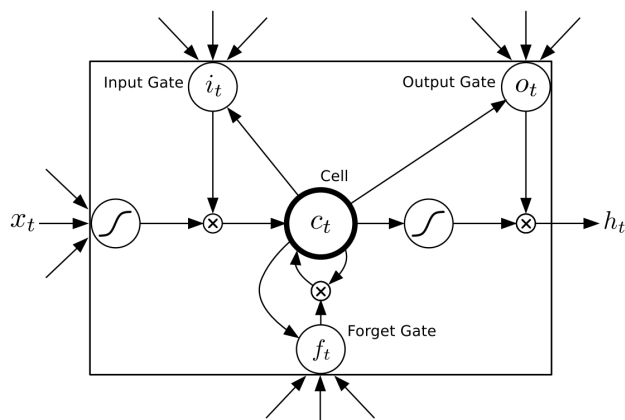


Figure 6.6: Long short term memory cell.

erate two sequences of LSTM hidden vectors. The backward LSTM processes the sequence in the reverse order. The hidden state at each time step, \mathbf{h}_t , is the concatenation of the two hidden vectors from both directions.

$$\mathbf{h}_t = [\vec{\mathbf{h}}_t, \overleftarrow{\mathbf{h}}_t] \quad (6.9)$$

Vectors $\vec{\mathbf{h}}_t$ and $\overleftarrow{\mathbf{h}}_t$ are the embeddings of the forward and the backward LSTMs at time step t , respectively.

6.3 Model

We treat the task of identifying the sentiment expressed for an aspect with respect to a location in a sentence as a three-way classification task. We define a classification task for each aspect separately. We use a MaxEnt model (softmax) to calculate the probability of a sentiment class for the aspect given the representation of the sentence specific to the location, \mathbf{e}_ℓ , as follows:

$$p(y_\ell = c | \mathbf{e}_\ell, w_c, b_c) = \frac{\exp(\mathbf{w}_c \cdot \mathbf{e}_\ell + b_c)}{\sum_{c' \in C} \exp(\mathbf{w}_{c'} \cdot \mathbf{e}_\ell + b_{c'})} \quad (6.10)$$

Here, the dot operator indicates the inner product of two vectors, C is the set of all the sentiment classes (Positive, Negative and None) and y_ℓ is the sentiment label of the given aspect for the location ℓ in the sentence s . $\theta = \{\mathbf{w}_c, b_c\}$ is the set of pa-

parameters representing the weights and the bias specific to each sentiment class c . Note that a sentence can contain more than one location. The embedding of the sentence with regards to different locations can be different. The vector \mathbf{e}_ℓ denotes the representation of the sentence s for the location ℓ . The representation of a sentence with respect to a location should embed all the necessary information to make a prediction for the sentiment of the given aspect towards that location.

6.3.1 Training

To train the above model, for a given aspect, we minimise the negative likelihood of data in the training set. Let's assume that N is the number of sentences in the training set, L is the number of locations in a sentence, C is the set of all sentiment classes and $\mathbf{e}_\ell^{(i)}$ refers to the representation of the sentence i in the dataset with respect to the location ℓ . Further, $I(y_\ell^{(i)} = c)$ indicates whether the true label y_ℓ for the given aspect is the sentiment class c . An L_2 term is added to the loss function for regularisation purposes and λ is a hyperparameter which can be tuned using cross validation. The loss function over all the training instances can be defined using the following:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{\ell \in L} \sum_{c \in C} I(y_\ell^{(i)} = c) \log p(y_\ell^{(i)} = c | \mathbf{e}_\ell^{(i)}, w_c, b_c) + \lambda \|\theta\|^2 \quad (6.11)$$

6.4 Representations

In an aspect-based sentiment analysis task, a representation should contain the information for identifying an aspect and its relevant sentiment. In the *targeted* aspect-based sentiment analysis task, the information regarding the target entity, here a location, should also be present. In other words, the representation of a sentence needs to reflect the context in which a location is discussed in order to identify the correct information specific to that location. A human asked to do this task will selectively focus on the relevant parts of the sentence, and acquire information where it is needed to build up an internal representation towards a

location in their mind. Figure 6.7 shows some examples of sentences with multiple target locations where the relevant context for each location is highlighted with matching colour to the location.²

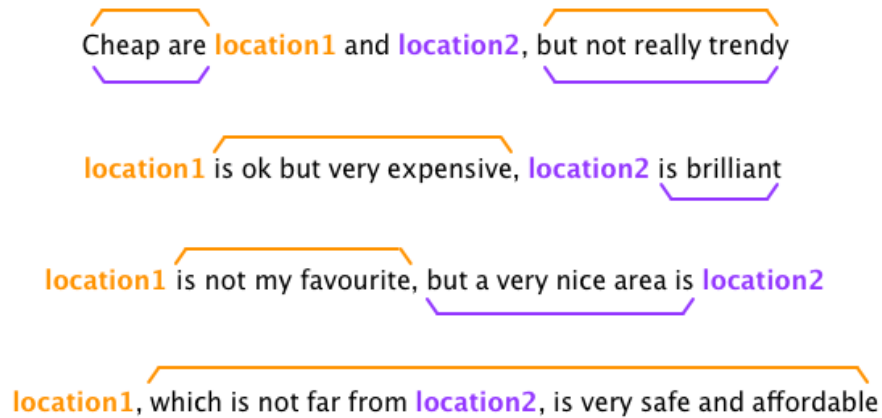


Figure 6.7: Examples of sentences with multiple locations. The context of each location is marked with the same colour as the location name.

As the figure illustrates, two locations sometimes share the same context (the top sentence). They can also have separate contexts with each relevant context appearing before or after the location mention (sentence 2 and 3). Note that in the last sentence, no opinion is expressed towards **location2**. **Location2** does not have any context of its own, but it appears in the middle of the context of **location1**.

For a model to identify the correct sentiment for an aspect of a specific location, it should be presented with a representation that focuses on the appropriate context. Representations that we propose in this section, aim at capturing the information from the relevant context to each location.

6.4.1 Bag of N-grams Representations

In this thesis, we focus on generic features for the bag of n-grams representations that do not require extensive engineering efforts. This includes both the sparse bag of n-grams representations and the bag of dense representations based on word embeddings. Dense representations can be beneficial since they can cap-

²There are more variations of the structure of the context of locations in the dataset.

ture similarities between words even if they have not been observed in the training set. Each representation is then fed into a multi-class logistic regression model to calculate the probabilities of each sentiment class for the given aspect with respect to a location. In the following, we propose variations of sparse and dense representations. We will demonstrate in figures how each representation is defined over a sentence. We take the following sentence as an example for this demonstration: “I heard that **location1** is expensive, I recommend **location2** which is nicer.”

Masked Target Entity For each location, we define a binary bag of n-grams representation over the entire sentence but mask the target location using a special token. This can help to differentiate between the representations of two locations that are present in the same sentence. We experiment with word uni-grams, word bi-grams and POS information. For a uni-gram representation, we have $\mathbf{e}_\ell \in \mathbb{R}^{|V|}$. Figure 6.8 shows how the representation is defined for each of the two locations, showing only a few uni-grams and bi-grams. As we can see, masking the target location entity makes the representations of the sentence distinct for two locations.

I have heard that **target_loc** is expensive, I recommend location2 which is nicer

target_loc_expensive				expensive			recommend_location2				
....	1	0	1	1	0	1
location1_expensive				nicer			recommend_target_loc				

I have heard that location1 is expensive, I recommend **target_loc** which is nicer

target_loc_expensive				expensive			recommend_location2				
....	0	1	1	1	1	0
location1_expensive				nicer			recommend_target_loc				

Figure 6.8: Masked target entity representations of two locations using word uni-grams and bi-grams.

Left-Right Contexts We create a binary bag of n-grams representation separately for each of the right and the left context around each location mention. We then concatenate these two representations to obtain one representation. A model

using this representation can give higher weights to specific terms on the left or the right context of each location. Defining a separate representation of the left and the right contexts of a target entity has been used in the past in targeted sentiment analysis task [2]. Here, we have $\mathbf{e}_\ell \in \mathbb{R}^{2 \times |V|}$ for a word uni-gram representation of the left and the right contexts. Figure 6.9 illustrates the left-right context representations of the sentence for two locations.

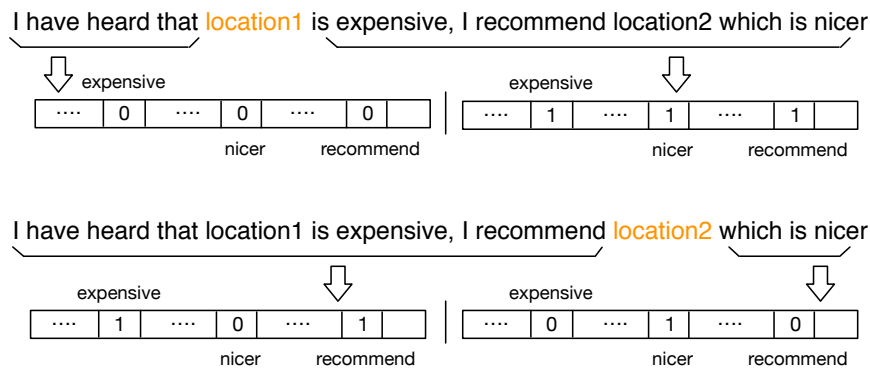


Figure 6.9: Left-right context representations of two locations using word uni-grams.

Context Window To capture the context of each location, a bag of n-grams is defined over the window around the mention of each location. This representation assumes that words that are in closer proximity to the location mention are more relevant to the location. We experiment with windows of various sizes. Here, we have $\mathbf{e}_\ell \in \mathbb{R}^{|V|}$ for a uni-gram representation of the context window. Figure 6.10 shows the representations defined over the context window of each location with a window of size 2 (on either side).

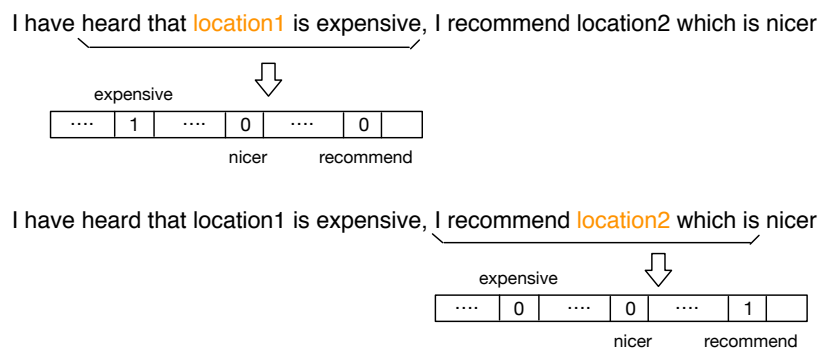


Figure 6.10: Context window representation of two locations using word uni-grams.

Distance-Bucketed Distance buckets are created to represent words that occur in a distance window to the location name. A feature is then the combination of an n-gram and a bucket index, e.g. “good_1” or “dodgy_2” where 1 and 2 indicate the bucket index. A bucket contains words that are at a distance range to the location name, irrelevant of whether they appear on the right side or the left side of the location name. Each bucket is represented using the bag of its word n-grams. The representation of all the buckets are combined to obtain a single representation of the sentence for a specific location. A model using this representation can learn that the closer a specific term is to the location, the more relevant it is when identifying the relevant aspect and sentiment for the location. We experiment with different bucket sizes. The buckets that are close to the location name can be small. Bucket sizes can increase as their distance to the location name increase. Here, we have $\mathbf{e}_\ell \in \mathbb{R}^{b \times |V|}$ for a word uni-gram representation of distance-bucketed, where b is the number of buckets. Figure 6.11 shows an illustration of the distance-bucketed representation for location1 only.

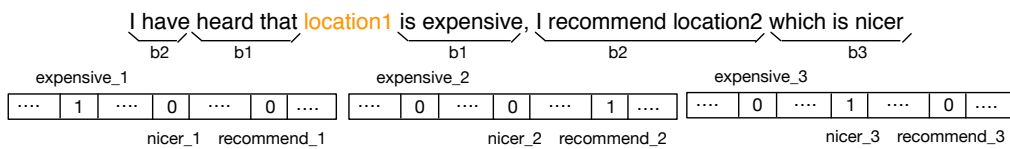


Figure 6.11: Distance-bucketed representation for **location1** using word uni-grams.

Sum of Embeddings The sum of embeddings representation creates a dense encoding of a sentence. For this, we first look up the word2vec embeddings of all the words in the sentence. We then sum these vector embeddings to create a representation for the entire sentence. This is similar to the bag of word uni-grams representation. However, here, dense embeddings of words are used instead of their one-hot vectors. Note that this method will result in the same representation for multiple locations in the same sentence. Therefore, we expect this representation to do poorly in sentences where two location entities are present. Here, we have $\mathbf{e}_\ell \in \mathbb{R}^d$ where d is the dimension of the word embeddings. We can formalise this representation as below where $\text{vec}(w_j)$ is the embedding vector of the

word at index j in the sentence and the *sum* operator is the element-wise sum of vectors.

$$\mathbf{e}_\ell = \text{sum}\{\text{vec}(w_1) \dots \text{vec}(w_j) \dots \text{vec}(w_n)\} \quad (6.12)$$

This representation is illustrated in Figure 6.14. Vectors of embeddings are in grey to show that they are dense embeddings.

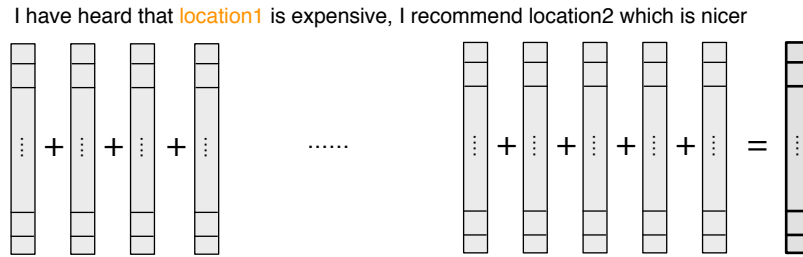


Figure 6.12: Sum of embeddings representation of the sentence for **location1**.

Sum of Left-Right Embeddings We first calculate the sum of embeddings of the left and the right contexts separately around the target location. The left and the right embeddings are then concatenated to obtain a single representation for the target location. This method is similar to the left-right context representation of word uni-grams. However, we use dense embeddings of the words instead of their one hot vectors. Note that this method will result in multiple locations in the same sentence having different embeddings. Here, we have $\mathbf{e}_\ell \in \mathbb{R}^{2 \times d}$ where d is the dimension of the word embeddings. This representation is illustrated in Figure 6.13.

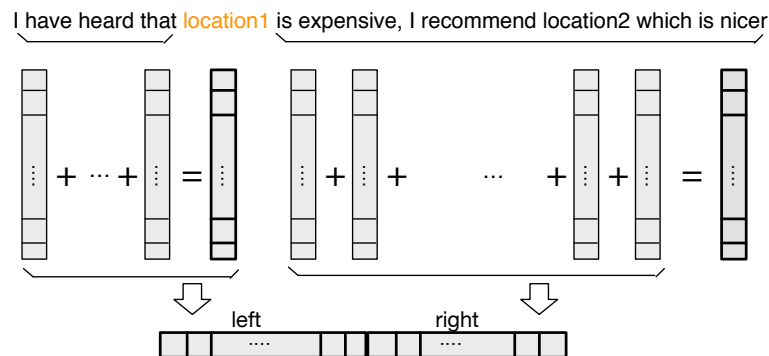


Figure 6.13: Sum of left-right embeddings representation of the sentence for **location1**.

Pooling of Left-Right Embeddings In previous work, pooling of the dense embedding representations over the left and the right contexts has been used for automatic feature detection in the targeted sentiment analysis task [2]. Inspired by this approach, we obtain max, min, average and standard deviation pooling over all the word embeddings for the left and the right context separately. We then combine the pooled embeddings of the left and the right context to obtain a single representation. Here, we have $\mathbf{e}_\ell \in \mathbb{R}^{8 \times d}$ where d is the dimension of the word embeddings.

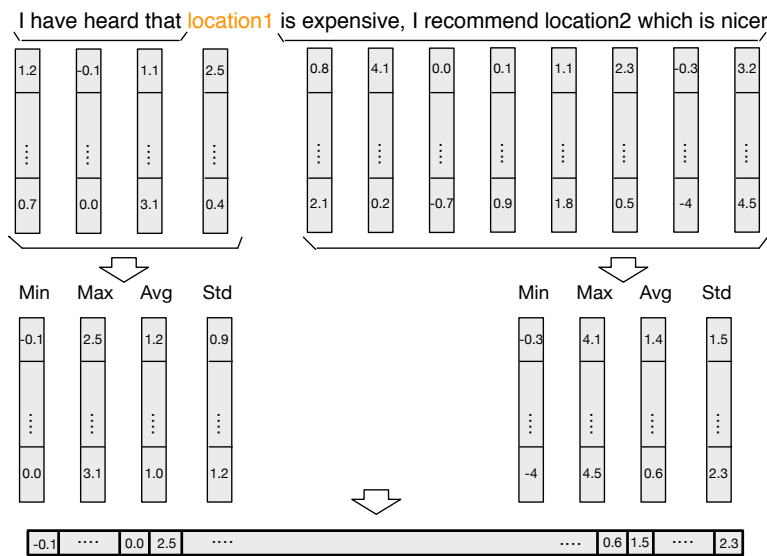


Figure 6.14: Pooling of left-right embeddings representation of the sentence for **location1**.

6.4.2 Sequential Representations

Inspired by the recent successes of applications of sequential learning in NLP tasks [86, 87, 88] and by the use of sequentially learned representations in classification tasks [122], we use *generic* variants of LSTM networks to learn the representations for each target location in a sentence. These representations are obtained using the three models explained in this section. As discussed in Section 6.3, to identify the sentiment class of a given aspect for each location in a sentence, the representations are then fed into a softmax layer to calculate the probabilities of each sentiment class. In a sequential representation setting, this

softmax layer is built at the top of the LSTM model where its parameters are learned jointly with the parameters of the LSTM model.

Unidirectional LSTM - Final (uniLSTM-Final) We use the final representation of a forward LSTM network as the representation of the sentence with respect to the location ℓ , i.e. $\mathbf{e}_\ell = \mathbf{h}_n \in \mathbb{R}^k$ where n is the length of the sentence and k is the dimension of the hidden representations in the LSTM model. To obtain a different representation for each location, we mask the target location.

Bidirectional LSTM - Final (biLSTM-Final) Here, \mathbf{e}_ℓ is the output representation of a bidirectional LSTM. Here, $\mathbf{e}_\ell \in \mathbb{R}^{2 \times k}$ where k is the dimension of the hidden representations in the LSTM model.

Bidirectional LSTM - Index (biLSTM-Index) The sentence representation for a location is the output of the time step j , $\mathbf{e}_\ell = \mathbf{h}_j$, in a bidirectional LSTM where j is the index of the target location in the sentence. This is illustrated in Figure 6.15 where the index of **location1** is 0 and the index of **location2** is 4. Here, $\mathbf{e}_\ell \in \mathbb{R}^{2 \times k}$ where k is the dimension of the hidden representations in the bidirectional LSTM.

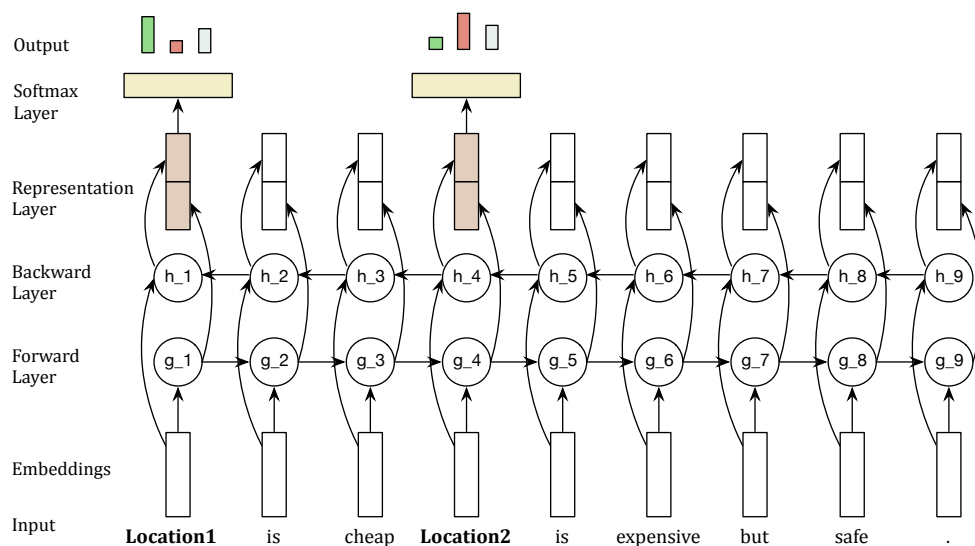


Figure 6.15: Bidirectional LSTM outputs a representation for each token in the sentence. The output at the index of each location is then fed into a softmax layer to identify the sentiment class for the corresponding aspect. In this figure, LSTM is trained to identify the sentiment of the aspect *price*. The model should predict Positive for **location1** and Negative for **location2**.

6.5 Experiments

In this section, we describe our experimental set up.

Aspects

In order to investigate the results of our experiments in great details, we select the four most frequent aspects from the dataset which are *price*, *safety*, *transit-location*, and *general*. The same approach can be applied to the remaining of the aspects. The prediction of each aspect was treated as a separate task.

Instances

The SentiHood dataset consists of sentences that contain one or two location entities. To simplify the set up of our experiments, we make instances that contain one or two locations homogeneous using the following approach. For each sentence that contains two location names, we create two separate instances. Each instance contains labels for only one of the locations, the target location. This means that for each aspect, given an instance, a model only needs to identify the relevant sentiment. Note that the aspect is implicit since a task is defined for each aspect separately. The sentiments reflect the opinion that is expressed for the target location only (for the implicit aspect). We mask the target location with the token “*target_loc*” and replace the other location name with the token “*location1*”.

Evaluation

We report the classification results on two basis, aspect detection and sentiment identification as discussed in the last chapter (Section 5.2.4). The presence of an opinion for an aspect of the target location is identified correctly if the predicted sentiment class is not None. We report both F_1 and AUC for aspect detection. F_1 score is calculated with a threshold that is optimised on the validation set. The correct sentiment is detected if the predicted sentiment class, i.e. Positive or Negative, matches the true value of the sentiment for that aspect of the target location. We report both accuracy and AUC for sentiment identification.

Word Embeddings

Existing work often uses pre-trained word embeddings provided with the

word2vec tool,³ which is trained on Google News dataset (about 100 billion words). The language people use on social media, however, can be different from the language used on the news. This means that the embeddings for some of the words in our dataset might not be present in the pre-trained embeddings. To avoid this, we train word embeddings by running the word2vec model on our corpus. Since the size of the corpora used for training the word embeddings is much larger than our corpus (which is around 9 million words), we combine our corpus with an existing social media data corpus. This corpus is provided with the word2vec tool and contains around 3 billion words.⁴ Embeddings are obtained using the continuous bag of words model (where the word is predicted using its neighbouring words), with the context size set to 5 and the dimensionality to 100.

The number of missing words from our dataset in the pre-trained embeddings is 768. The missing words include the words with a British spelling (e.g. “neighbourhood”, “neighbouring”, “favourite”, “centre”, “theatre”, “characterised”, “colour”, etc.), names of areas or places (e.g. “Belgrave” or “Vauxhall”), terms that are specific to the UK (e.g. “Londoner”, “crossrail”, “Edwardian”), and other words that are more common in the social media data (e.g. “wanna”, “dodgyest”). By training the word embeddings on our corpus, we reduce the number of missing words to 150. These missing words include postcodes (e.g. sw12) and misspelled words.

Location Embeddings

To represent location entities, i.e. **target_loc** and **location1** in the embedding space, we extend the dimension of our word2vec embeddings by two, one per each location entity. Figure 6.16 shows the embeddings of **target_loc** and **location1** and the existing words in the corpora.

Splitting the Dataset

We divide sentences that contain one (Single) or two locations (Multi) separately

³<https://code.google.com/archive/p/word2vec/>

⁴<http://ebiquity.umbc.edu/redirect/to/resource/id/351/UMBC-webbase-corpus>

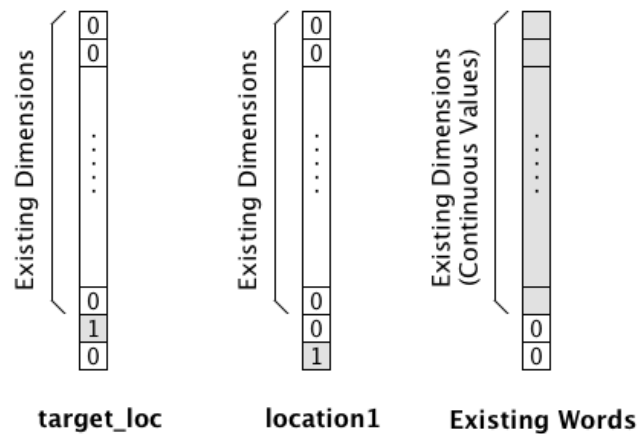


Figure 6.16: Word embeddings are extended by two cells to represent two tokens of “target_loc” and “location1”.

into train, dev and test set, with each having 70%, 10% and 20% of the data respectively. We then combine both subsets to obtain the final sets for train, dev and test. To optimise the hyperparameters of the models (e.g. LSTM parameters, F_1 threshold, logistic regression weights), we cross-validate the performance on the dev set and report the final results on the test set.

Table 6.1 shows the number of sentences from each category (Single, Multi, Multi-Agree and Multi-Disagree) in each of the train, dev and test sets. Multi-Agree and Multi-Disagree refer to sentences that have similar or different sentiments for the same aspect with regard to two target locations. The total number of opinions are also displayed. Note that one sentence can have none, one or a higher number of opinions.

Table 6.1: Statistics for train, dev and test sets.

	# Sentences	#Single	#Multi	# Multi Agree	# Multi Disagree	#Opinions
Train	2977	2202	775	980	294	3401
Dev	747	557	190	260	60	840
Test	1491	1103	388	522	154	1679

Training the LSTM Models

For each aspect in our dataset, we have many instances of sentences with a None class sentiment. A None class sentiment means that there is no opinion ex-

pressed towards a location with respect to that aspect in the sentence. To tackle the problem of having an unbalanced dataset, i.e. many more None class instances than Positive or Negative, we use the re-sampling [123, 124] technique in the following way. We train LSTM models in batches with every batch having the same number of sentences sampled randomly from each sentiment class.

To optimise the value of each hyperparameter of an LSTM model, we perform a greedy search for the values of parameters in the following way. To start with, we heuristically assign all parameters a default value. For each parameter, we then run the model using the values in a specified reasonable range. An optimum value for the hyperparameter is the one that results in the minimum loss on the development (dev) set. After choosing an optimum value for a hyperparameter, we continue with the next one, setting the value of the previously tuned hyperparameter to its optimum value. We continue this procedure until the optimum value for all the hyperparameters are estimated.⁵ We adopt this method instead of a grid search because it is more time-efficient. This allows us to search through a wider range for each hyperparameter in a timely manner. Hyperparameters of our LSTM models are the learning rate, the hidden size, the batch size, and the dropout probability. The lowest loss on the dev set is achieved when a learning rate of 0.01, hidden units of size 50 and batch sizes of 150 are used. Adding dropout did not improve the results of any of the models. The Adam [125] optimiser is used for the optimisation with an initial learning rate of 0.01.

To estimate the parameters of the model, we train the model on the train set and then evaluate the loss function (Equation 6.11) on both the train and the dev set after each iteration. We train for the maximum number of iterations which is 120. We save the best model which has the lowest loss on the dev set across all the iterations. We then run this model on the test set and report the results. Tensorflow [126] is used for the implementation of the proposed LSTM models.

Training Logistic Regression

Logistic regression models were based on the implementation from the scikit-

⁵Note that these values may not be globally optimum as we do not perform a grid search.

learn⁶ library in Python. Since we have an unbalanced dataset, we use a weighted logistic regression. To obtain the best weights, we cross-validate them on the development set. Weights inversely proportional to the size of each class result in the best performances.

6.6 Results

Table 6.2 shows the results averaged over the selected aspects in terms of F_1 and AUC for aspect detection and accuracy and AUC for sentiment identification.⁷

Model/Representations	Aspect (F_1)	Sentiment (Accuracy)	Aspect (AUC)	Sentiment (AUC)
LR-Unigram	0.568	0.840	0.913	0.874
LR-N-gram	0.707	0.853	0.918	0.886
LR-N-gram-POS-Uni	0.393	0.875	0.925	0.905
LR-N-gram-Left/Right	0.693	0.847	0.903	0.871
LR-Window [10]	0.344	0.822	0.859	0.860
Distance-Bucketed [2,4,-]	0.412	0.793	0.831	0.815
LR-Embeddings-Sum	0.362	0.791	0.837	0.797
LR-Embeddings-Sum-L/R	0.392	0.782	0.845	0.785
LR-Embeddings-Pooling-L/R	0.341	0.775	0.862	0.810
uniLSTM-Final	0.268	0.378	0.497	0.479
biLSTM-Index	0.694	0.820	0.898	0.840
biLSTM-Final	0.690	0.825	0.898	0.850

Table 6.2: Aspect and sentiment classification results using different types of representations. Results are shown both in terms of F_1 and accuracy and AUCs for aspect and sentiment classification.

The first six rows in the table, show the results of the sparse bag of n-grams representations which are trained using a logistic regression model. LR-Unigram refers to a word uni-gram representation with masked target location. Similarly, LR-N-gram refers to a combination of word uni-grams and bi-grams with masked target location. LR-N-gram-POS-Uni includes word uni-grams and bi-grams and

⁶<http://scikit-learn.org/>

⁷Even though, F_1 and Accuracy are the common metrics to use for classification tasks, we will report the results using AUC only in the remainder of this chapter. The main reason for this is that we can average the performance of a representation over both aspect and sentiment classification. It is then more convenient to compare two representations. Moreover, when categorising sentences into different aspects and sentiments, it is reasonable to show users a ranked list of sentences that are more probable to describe the desired category and sentiment.

uni-gram POS features (obtained by concatenating a word with its POS tag, e.g. nice_JJ) where target location is masked. LR-N-gram-Left/Right refers to a representation obtained by concatenating word uni-grams and bi-grams of the left and the right contexts. LR-Window defines word uni-grams and bi-grams over the context window around the target location name. We experimented with different context window sizes and report the best performing setting which is 10. Distance-Bucketed representation is defined using word uni-grams and their distance from the name of the target location. Different bucket sizes were examined. They all perform relatively poorly. We report results on bucket distances of size 2, 4 and a bucket containing the remaining words that are further than 4 tokens from the target location mention.

The next three rows in the table show the results of the dense bag of n-grams representations which are trained using logistic regression models. The last three rows show the results of sequential representations obtained by variations of LSTM models.

Sequential vs. Bag of N-grams Representations

It is interesting to observe that sequential representations obtained using LSTM models are not superior to some of the bag of n-grams representations. The performances of the dense representations of bag of n-grams are low in general in comparison with other representations. A forward only LSTM representation (uniLSTM-Final) performs very poorly. The performances of the two variations of biLSTMs are very similar and not significantly different from each other.

One thing to note is that our proposed methods can achieve prediction accuracies on similar levels to the task of aspect-based sentiment analysis on SemEval dataset.⁸ This indicates that the task of targeted aspect-based sentiment analysis on SentiHood dataset is a plausible task, despite the extra level of complexity that the identification of target entity introduces.

Context

Results also indicate that the performance is higher when the entire sentence

⁸Best performing SemEval participants achieve F_1 of 0.73 for aspect detection and accuracy of 0.88 for sentiment classification [39].

is used as the context. The performances of both context-based representations, i.e. LR-Ngram-L/R and LR-Window are lower than the performances of representations that are based on the entire sentence. The best performing representation is the representation that contains word uni-grams, word bi-grams, and POS uni-grams of the entire sentence (LR-Ngram-POS). Even though, the performance of this model in detecting the aspect is very low in terms of F_1 metric. This shows that determining a hard threshold for labeling instances with their predictions is difficult. However, the AUC performance of this representation for both aspect and sentiment classification is superior to other representations. For the rest of this chapter, we make comparisons in terms of AUC between the best performing bag of n-grams representation (LR-Ngram-POS) and one of the best performing sequential (biLSTM-Index) representations which we refer to as BoNgrams and SEQ respectively.

Aspects

We show the performances of SEQ and BoNgrams representations separately on each aspect in Table 6.3. The performance metric is the average AUC on sentiment and aspect detection. The numbers in parenthesis indicate the number of distinct expressions that are annotated for each aspect in the dataset.⁹ Even though, the expressions for aspect *general* are always “null” in the dataset, there are many variations of expressing a *general* aspect for a neighbourhood. It is interesting to note that the performance drops as the number of distinct expressions increases for an aspect. For instance, the aspect *safety* can be predicted more accurately than the aspect *transit-location*. Aspect *general* is the lowest performing aspect. Both representations perform very similarly on the aspect *general* with SEQ having a slightly higher performance.

⁹Please refer to Figure 5.6 in Chapter 5

Model	Safety (31)	Price (56)	Transit-location (116)	General
BoNgrams	0.954	0.944	0.901	0.861
SEQ	0.921	0.910	0.764	0.879

Table 6.3: Performances of the best sequential (SEQ) and bag of n-grams (BoNgrams) representations on each aspect. AUC scores are averaged over aspect and sentiment detection.

Categories

Table 6.4 shows the average AUC using SEQ and BoNgrams representations for different categories of sentences: Single — sentences that contain one location entity and Multi — sentences that contain two location entities. Multi-location sentences are further divided into Multi-Agree and Multi-Disagree. BoNgrams performs better than SEQ on all the categories. As expected, both representations perform better on Single and Multi-Agree categories. The most difficult category of sentences for this task is Multi-Disagree. Consider the following sentence: “*location1 is nice, trendy and central but very expensive, location2 is cheaper*”. In this sentence, Negative and Positive sentiments are expressed for the aspect *price* of location1 and location2, respectively. A word uni-gram or bi-gram feature representation may not capture that the word “expensive” is expressed towards location1. A sequential representation may, in theory, be able to capture this information since it is not based on pre-defined features and it has a high capacity for embedding information. However, it is surprising that BoNgrams representation performs better than SEQ representation on average on this category as well.

Model	Single	Multi	Multi-Agree	Multi-Disagree
BoNgrams	0.918	0.905	0.940	0.814
SEQ	0.889	0.839	0.851	0.761

Table 6.4: Performances of SEQ and BoNgrams representations on different categories of sentences. AUC scores are averaged over aspect and sentiment, for all the aspects.

Categories per Aspects

So far, we have seen that the BoNgrams representation outperforms the SEQ representation overall, on most aspects and all the categories of sentences. In this

section, we look at the results in more details. Figure 6.17 illustrates the performance of BoNgrams and SEQ representations on each selected aspect divided into different categories of sentences. Blue bars indicate the performance of SEQ representation and orange bars indicate the performance of the BoNgrams representation. As we can see, the largest difference in performance is for the aspect *transit-location* where BoNgrams representation performs much better than SEQ. On the other hand, a better performance is achieved by SEQ representation for the aspect *general* on all the categories of sentences. Note that as we have seen so far, the aspect *general* is the hardest aspect to predict.



Figure 6.17: Performances of the best sequential (SEQ) and bag of n-grams (BoNgrams) representations for each aspect and different categories of sentences.

Sentence Length

Here, we look at the performance of each representation as the length of the sentences, in terms of the number of tokens, increases. This can indicate the robustness of the model to noise and to handling the longer range dependencies. Longer sentences are more likely to contain irrelevant information, i.e. noise.

Capturing long-range dependencies is important in sentences where aspect and sentiment related terms appear not very close to the target location mention. For instance, in the sentence “*I live in location1, there are some great pubs, affordable rents, and great access to transport not too far from location2*”, “great access to transport” is expressed for location1 but it is not in a close proximity to it in the sentence.

Figure 6.18 shows the ratio of the correct predictions for different ranges of the sentence length using both SEQ and BoNgrams representations. Length ranges are taken at 5-token intervals. Each range (e.g. [5 – 10]) is represented by a circle with a y-value equal to the ratio of correctly labeled sentences. The size of each circle indicates the number of sentences with a length in the given range. The figure shows that there is a downward trend of the ratio of the correct predictions as the lengths of the sentences increase, for both SEQ and BoNgrams representations. Even though the slope is sharper at the beginning for SEQ for sentences of length 10 to 20, the downward trend changes for longer sentences with a high ratio of correct predictions for the really long sentences. In the case of BoNgrams, the downward trend continues with some exceptions of the very long sentences of around 80 tokens. The number of these sentences, however, is very small.

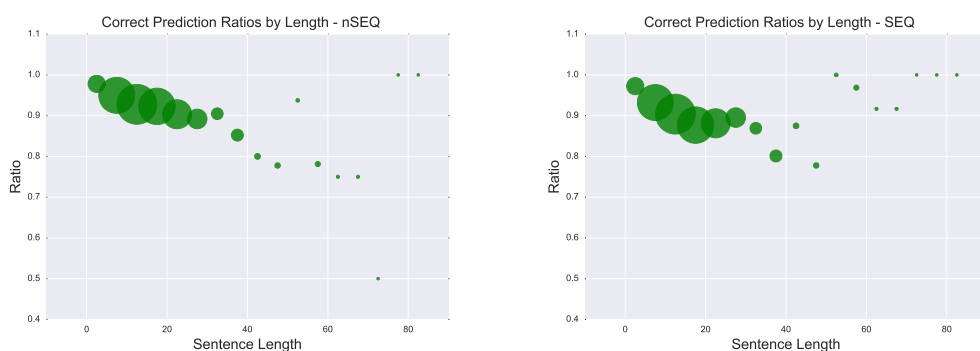


Figure 6.18: The ratio of the correct predictions as the length of sentences increases. The size of each circle indicates the number of sentences in the length range.

Further, Table 6.5 shows the correlation between the length of sentences and the prediction (correct or incorrect) in terms of point-biserial correlation coeffi-

cient. This is calculated over the aspect and the sentiment classification for all the aspects and categories. As we can see, in both representations, there is a negative correlation between a correct prediction and the length of the sentence. This means that longer sentences are harder to label. BoNgrams shows a higher negative correlation meaning that it is worse in making predictions for longer sentences than SEQ representation.

Table 6.5: Correlations ($r < 0.01$) between the length of a sentence and the prediction in terms of point-biserial correlation coefficient.

Representation	Correlation
BoNgrams	-0.10
SEQ	-0.06

Predictions

In this section, we present examples of sentences that are labeled correctly or incorrectly using SEQ and BoNgrams representations. The labels are determined by thresholding the probabilities with the threshold optimised on the development set. Table 6.6 shows examples of predictions made using the BoNgrams (top) and SEQ (bottom) representations.

Examples in the table include both correctly and incorrectly labeled sentences, for both Single and Multi categories. The first sentence “*target_loc is not a nice cheap residential area to live trust me, i was born and raised there*” shows an example of an incorrect prediction for the aspect *price*. This is because there is a case of negation that cannot be captured using bi-grams representation (“not a nice cheap”). The second sentence “*I think you'd find it tough to find something affordable in target_loc*” is labeled incorrectly as Positive for the aspect *price* because according to the weights of the logistic regression model, a word with the lemma “afford” and a POS tag of “JJ” is a strong indicator for a Positive sentiment class. On the other hand, the third sentence “*I can't recommend target_loc for affordability*” is labeled correctly as Negative. This is because a word with the lemma “afford” and the POS tag of “NN” strongly indicates a Negative sentiment class.

Sentence	Aspect	Predicted	Label
target_loc is not a nice cheap residential area to live trust me, i was born and raised there	Price	Positive	Negative
I think you'd find it tough to find something affordable in target_loc	Price	Positive	Negative
I can't recommend target_loc for affordability	Price	Negative	Negative
I would hardly consider target_loc too far out, location1 is not an unsafe area	Safety	None	None
I would hardly consider location1 too far out, target_loc is not an unsafe area	Safety	Positive	Positive
target_loc is a bit of a dump, location1 is slightly nicer, though both have better and worse areas	General	Negative	Negative
location1 is a bit of a dump, target_loc is slightly nicer, though both have better and worse areas	General	Negative	Positive
target_loc is not a nice cheap residential area to live trust me, i was born and raised there	Price	Positive	Negative
I think you'd find it tough to find something affordable in target_loc	Price	Negative	Negative
I can't recommend target_loc for affordability	Price	None	Negative
I would hardly consider target_loc too far out location1 is not an unsafe area	Safety	Positive	None
I would hardly consider location1 too far out target_loc is not an unsafe area	Safety	Positive	Positive
target_loc is a bit of a dump, location1 is slightly nicer, though both have better and worse areas	General	Negative	Negative
location1 is a bit of a dump, target_loc is slightly nicer, though both have better and worse areas	General	Positive	Positive

Table 6.6: Examples of input sentences and their predicted labels using BoNgrams (top) and SEQ (bottom) representations.

Let us now consider an example of a multi-location sentence: “*I would hardly consider target_loc too far out, location2 is not an unsafe area*”. This sentence does not have any indicative word for the aspect *safety* of the `target_loc` and therefore is correctly labeled as None. The second instance of the sentence is correctly labeled Positive for *safety* because the bi-gram feature “`target_loc_not`” has a large coefficient for sentiment class Positive when trained on aspect *safety*.

For comparison, we show the same examples of sentences at the bottom of the table where predictions are made using the SEQ representation. Unlike a logistic regression model which is based on sparse features (e.g. BoNgrams), results of an LSTM model is not easily interpretable.

6.6.1 Synthetic Evaluation Set

So far, our analysis has been based on the prediction results on the test set. To further analyse the capabilities of BoNgrams and SEQ representations, in this section, we create a synthetic evaluation dataset. This dataset is divided into different categories. Each category contains sentences with a type of complexity that is present to some extent in our real dataset. Since the performances of both representations is comparable on the aspect *general*, we create the synthetic set for this aspect. In the following, we present different categories of synthetic data. Each of the categories contains between 50 to 120 instances. We will explain each category and the results of the predictions on that category using both SEQ and BoNgrams.

6.6.1.1 Single Location - Lexical Variation

Description

This category of synthetic sentences focuses on testing the capability of a representation in detecting an aspect and its relevant sentiment when presented with lexical variations. Only one location is present in the sentences in this category. Examples of lexical variations are rare or unseen adjectives. Some examples of the sentences in this category are shown below. We create this set by looking at the simple and often short examples in the real dataset and replacing some of the

terms, especially adjectives by their synonyms.

target_loc is a <u>horrendous</u> area.	(target_loc,general,Negative)
target_loc is <u>enchanting</u> .	(target_loc,general,Positive)

Predictions

Performances of SEQ and BoNgrams representations on this category are illus-



Figure 6.19: Performances of SEQ and BoNgrams representations on the *Lexical Variation* synthetic category on aspect detection, sentiment detection and on average.

trated in Figure 6.19. The figure shows that the performance of SEQ representation is superior to BoNgrams representation with its average AUC higher by 20%. This is what we expect as the dense representations should be able to handle unseen terms better than sparse representations. This is because they are based on word embeddings that are learned from a large corpus.

6.6.1.2 Single Location - Negation

Description

Negation is a simple example of composition in natural language. A representation should be able to capture negation, even without seeing every possible combination (e.g. *not awful*) or when the token indicating the sentiment does not follow the negation token immediately (e.g. *not really that bad*). This category was created by observing the sentences in the real dataset that contain negation. We create samples by using these examples, using different ways of composition (e.g. “not great”, “not that great”, “not really great”) and using different adjectives

(e.g. good, nice, ok). In this category, to test only the capabilities of the representations in capturing the negation, we avoid introducing unseen terms. Similar to the previous category, only a single location is present in each sentence. Note that even though all of the terms have appeared in the training set, not all the combinations of terms and negation tokens have occurred in the training set.

target_loc is <u>not</u> really awful.	(target_loc,general,Positive)
target_loc is <u>not</u> that impressive.	(target_loc,general,Negative)

Predictions

Figure 6.20 illustrates the performances of SEQ and BoNgrams representations in predicting the correct aspects and sentiments on the Negation synthetic category. The figure shows that the SEQ representation performs better in predicting both the aspect and the sentiment. The performance of both representations drops when classifying the correct sentiment. This is understandable because sentiment detection requires understanding the negation. One reason for the poor performance of both of SEQ and BoNgrams representations in sentiment detection can be that there are not many different variations of negation in the training set.

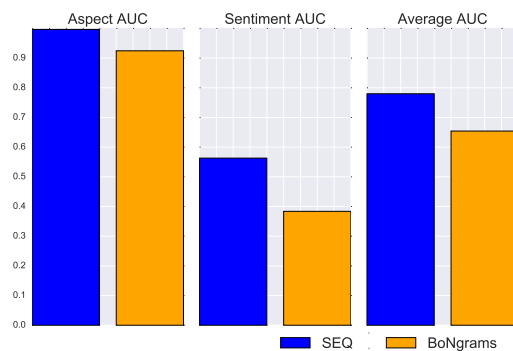


Figure 6.20: Performances of SEQ and BoNgrams representations on the *Negation* synthetic category.

6.6.1.3 Single Location - Noise

Description

In this category, we still focus on sentences with a single location while intro-

ducing noise to each sentence. This is done by adding irrelevant text, mainly between the location name and the words indicating a sentiment for the aspect *general*. The irrelevant text fragments have been taken manually from the sentences in the real dataset. Most sentences are taken from the simple and short sentences in the training set. This category can test the robustness of the model towards the presence of irrelevant text and noise. It should also be able to detect whether a representation can capture longer range dependencies. The irrelevant text fragments are underlined in the examples.

target_loc , <u>where there are many shops,</u> is horrible.	(target_loc,general,Negative)
target_loc , <u>by London standards,</u> is a great area.	(target_loc,general,Positive)
target_loc is in south London, <u>there are bad and good areas everywhere.</u>	(target_loc,general,None)

Predictions

Figure 6.21 shows the performances of SEQ and BoNgrams representations on this synthetic category. Results can suggest that the SEQ representation can be more robust to noise and more capable of capturing the dependencies between sentiment words and the location name that are far apart in the sentence.



Figure 6.21: Performances of SEQ and BoNgrams representations on the *Noise* synthetic category.

6.6.1.4 Multiple Locations - Agreement (Multi-Agree)

Description

In this category, we create examples that contain two locations. Both locations agree on the sentiment towards the aspect *general*. This category therefore, captures the linguistic phenomena of coordination. The sentences are simple, short and not containing unseen words or negation. Examples are shown below. For each sentence, two instances are generated where the name of a location is replaced with *target_loc*.

location1 and location2 are horrible areas.	(location1,general,Negative) (location2,general,Negative)
You might also like to give places like location1 and location2 a try.	(location1,general,Positive) (location2,general,Positive)

Predictions

Results illustrated in Figure 6.22 show that BoNgrams representation outperforms SEQ representation in detecting the presence of the aspect. Note that for each location, the BoNgrams representation uses features defined over the entire sentence. This means that both representations share many features such as word uni-grams and some word bi-grams. Both representations can predict the sentiments for aspect *general* correctly over all the instances in this category. This can be because there exists no unseen words or negation in this set.

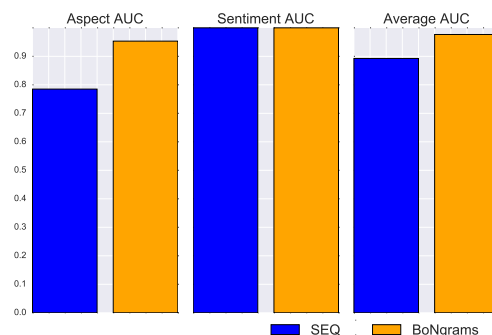


Figure 6.22: Performances of SEQ and BoNgrams representations on the *Multi-Agree* synthetic category.

6.6.1.5 Multiple Locations - Disagreement (Multi-Disagree)

Description

In this category of sentences, two locations are present in each sentence. The sentiments that are expressed for the aspect *general* towards the two locations are not the same. Many of these sentences are generated by combining two single-location sentences in different ways. These single location sentences are based on examples from the real data. For each sentence, two instances are generated where the name of a location is replaced with *target_loc*.

location1 is nice, location2 is bad.	(location1,general,Positive) (location2,general,Negative)
I don't know location1 very well, but location2 is a hole.	(location1,general,None) (location2,general,Negative)
Unlike location1 which is great its neighbouring area location2 is disgusting	(location1,general,Positive) (location2,general,Negative)

Predictions

Figure 6.23 illustrates the performances of SEQ and BoNgrams representations on this category of sentences. As the figure indicates, the SEQ representation can achieve a higher performance than the BoNgrams representation. This means that perhaps SEQ representation can be better in identifying the boundaries of the context of each location.

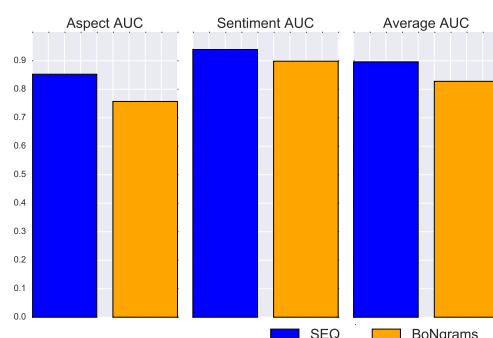


Figure 6.23: Performances of SEQ and BoNgrams representations on the *Multi-Disagree* synthetic category.

6.7 Data Augmentation

In the previous section, we presented the prediction results based on the test set. We further analysed the capabilities of the representations by observing their performances on a synthetic dataset. We have seen that the SEQ representation is superior to the BoNgrams representation on many of the synthetic categories defined for the aspect *general*. The SEQ representation is especially great in dealing with rare or unseen words and in capturing negation. However, overall, the performance of SEQ representation is lower than BoNgrams on the test set. The reason for this may be that in the test set, we do not have many rare or unseen words or many different variations of negation. Moreover, the synthetic dataset is based only on the aspect *general*. The SEQ representation can achieve a comparable performance to the BoNgrams representation when predicting the aspect *general* on the test as well. It is worth remembering that there is a higher number of training examples available for the aspect *general* in the dataset compared to other aspects. Neural models such as LSTMs often need a larger number of training samples to achieve good results. This may explain the good performance of the SEQ representation for the aspect *general*.

Therefore, in this section, we investigate whether generating more training examples through augmenting the existing training data can improve the prediction performance of the SEQ representation. Data generation and augmentation have been used in the past in machine learning [89] and NLP tasks [90] to inject prior knowledge and to improve the performance of the prediction models. We propose two general approaches for data augmentation: automatic augmentation and user-assisted augmentation. In the following, we first explain the two approaches for data augmentation. At the end, we present the prediction results that are obtained by training our models on the combination of the real data and the augmented data.


6.7.1 Automatic Augmentation

In the automatic augmentation approach, the augmentation process is isolated from the annotation process. The augmentation is applied to the annotated sen-

tences. New instances can be generated using one of the methods described below. These methods specifically produce more examples of sentences with multiple location entities. This is because predicting sentiment and aspect classes in sentences with multiple locations is a more difficult task. Moreover, the number of sentences with multiple locations in our training set is far less than the number of sentences with a single location.


Single to Multiple

In this category of augmentation, we take a single-location sentence and add a second location entity where both locations share the same context and therefore sentiment labels. To do this, we choose a single-location sentence randomly from the training data and replace the token “*location1*” with “*location1 and location2*” and make the verb plural if necessary. We also assign all the aspect-sentiment labels of *location1* to *location2*. An example is shown below:

location1 is very expensive	(location1 ,price,Negative)
	
location1 and location2 are very expensive	(location1 ,price,Negative) (location2 ,price,Negative)


Concatenating Two Single-Location Sentences

In this category of augmentation, we concatenate two existing single-location sentences. These sentences can contain opinions about similar or different aspects, with agreeing or disagreeing sentiments. Each sentence is selected randomly. Below, we provide an example:

House prices are very high in location1	(location1 ,price,Negative)
I used to live in location1	(location1 ,price,None)
	
House prices are very high in location1 , I used to live in location2	(location1 ,price,Negative) (location2 ,price,None)

Adding Noise

In this category, we add a fragment of text that contains the name of a second location, where no opinion is expressed for the second location. The added fragment should not affect the opinions that are expressed for the first location. Fragments are randomly selected from a pre-defined list. The list is created by observing the real data. Examples of these fragments are: “which is close to location2”, “in the borough of location2”, “near location2”, “neighbouring location2”, “adjacent to location2”, “south/west/east of location2”. Below is an example:

location1 is great for going out	(location1 ,nightlife,Positive)
	
location1 , north of location2 , is great for going out	(location1 ,nightlife,Positive) (location2 ,nightlife,None)

6.7.2 User-Assisted Augmentation

Many of the sentences in our dataset are complex and long. Often only a small fraction of each sentence suffices in detecting a specific aspect and its relevant sentiment towards a location. However, to identify the relevant fractions of a sentence, the interaction with a user is required. The augmentation procedure can be incorporated into the annotation process or can be carried out separately.

In this approach, we aim to generate lexical and syntactic variations of the existing sentences. For each sentence, we first generate its parse tree using the Stanford parser [127].¹⁰ For syntactic variations, we manipulate the parse tree of the sentence to generate a new sentence. The user input is then needed to confirm whether the new sentence contains the same information with respect to a specific location and a specific aspect. For lexical variations, we ask the user to propose adjectives to replace the identified adjectives in a sentence. These adjectives either keep or reverse the sentiment for a specific aspect of a given location.

¹⁰The implementation is obtained here: <http://nlp.stanford.edu/software/lex-parser.shtml>

We divide augmented sentences into several categories. Sentences in each category are generated by a specific type of manipulation to the parse tree of the original sentences. Approaches for generating these categories of sentences are described below.

Removing Fragments

To simplify a given sentence, we propose to drop specific syntactic subtrees of its parse tree. A candidate sub-tree for dropping can have one of the following syntactic types: subordinate clause (SBAR) e.g. “although parts of it are quite expensive”, conjunction e.g. “and expensive”, prepositional phrases (PP) e.g. “as a general rule”, sentence (S) e.g. “I lived there for 3 months”. For example the sentence “*I love target_location, I lived there for 3 months*” can be simplified to “*I love target_location*” and still preserve its Positive sentiment for the aspect *general* towards the *target_location*.

Inserting Fragments

To make the representations and the prediction models more robust to the noise in the data, we add noise to the sentences through inserting text fragments. This type of augmentation is automatic and does not need the interaction of a user. However, it utilises the simplified sentences from the previous category (Removing Fragments). The inserted fragments should not change the sentiment of a sentence with respect to a given aspect and a location. In order to do this, we first create templates from the simplified instances. A template is a sentence where a dropped subtree is replaced with a placeholder indicating the syntactic category of the subtree. For example, the sentence “*I love target_location, I lived there for 3 months*” is converted into “*I love target_location, [S]*”. The placeholder [S] can be later replaced by any other sentence in the training set.

All the possible replacements for each subtree category such as S, SBAR, PP, etc. are collected using the generated parse trees for all the sentences in the training set. We collect different subtree types for the combination of each aspect and sentiment classes. For instance, we create a list of all PP subtrees from the sentences that have a Positive sentiment towards the aspect *general*.

To generate a new sentence for a given aspect from a template without introducing a conflicting sentiment for `target_loc`, we only replace its placeholders with the subtrees with the same type and the same sentiment (or None) and the aspect. For instance, from the above template we can generate the two following sentences using fragments that carry Positive or None sentiments for aspect *general* towards `target_location`: “I love target_loc, it’s a nice area” or “I love target_loc, my friend lives there”.

Replacing Adjectives

This category of augmentation is designed for introducing lexical variations into our training data. For each sentence, we first identify all the adjectives. For each adjective, we ask the user to provide a list of related adjectives. Each adjective in the proposed list should be able to replace the identified adjective in the sentence while the sentiment of the sentence towards a specific aspect and the target location remains unchanged. Similarly, users are required to provide adjectives that reverse the sentiment for the aspect of the target location. New adjectives provided by the user is then used to generate new instances with the same or the opposite sentiments. Replacements are also applied to the simplified instances and the templates to generate more examples.

Below, we show an example of the procedure of the user-assisted data augmentation for a sentence. As the example shows, we first present the user with an example. We then ask the user to provide the sentiment class with regard to a specific location and an aspect. We then propose different augmentations for the sentence. At each step, user is required to confirm whether the same sentiment is preserved for the specified aspect and the location. Here, we only show the procedure for the aspect *general* and for location1. In reality, for each sentence, we repeat this process for all the aspects and all the mentioned locations. User input is highlighted using a different font.

Algorithm 1 User-assisted data augmentation.

```

1: procedure AUGMENTATION
2:
3:   Sentence: I say location2 is n't the best place but, location1 is very nice but
      expensive
4:   Location: location1
5:   Aspect: general
6:   > Sentiment? Positive
7:
8:   [DropConjunction]
9:   Sentence: I say location2 is n't the best place but, location1 is very nice
10:  Location: location1
11:  Aspect: general
12:  > Sentiment? Positive
13:
14:  [KeepOnlyTree]
15:  Sentence: location2 is n't the best place but, location1 is very nice
16:  Location: location1
17:  Aspect: general
18:  > Sentiment? Positive
19:
20:  [KeepOnlyTree]
21:  Sentence: but , location1 is very nice
22:  Location: location1
23:  Aspect: general
24:  > Sentiment? Positive
25:
26:  [DropConjunction]
27:  Sentence: I say location2 is n't the best place but, location1 is expensive
28:  Location: location1
29:  Aspect: general
30:  > Sentiment? None
31:
32:  [KeepOnlyTree]
33:  Sentence: location2 is n't the best place but, location1 is very nice but ex-
      pensive
34:  Location: location1
35:  Aspect: general
36:  > Sentiment? Positive
37:
38:  [Drop Conjunction]
39:  Sentence: but , location1 is expensive
40:  Location: location1
41:  Aspect: general
42:  > Sentiment? None
43:

```

Algorithm 2 User-Assisted Data Augmentation (cont.)

```

44: [Keep Only Tree]
45: Sentence: but , location1 is very nice but expensive
46: Location: location1
47: Aspect: general
48: > Sentiment? Positive
49:
50: [Replace Adjectives]
51: Replace "nice" without changing the sentiment?
52: > [great , amazing , lush]
53:
54: [Replace Adjectives]
55: Replace "expensive" without changing the sentiment?
56: > [pricey]
57:
58: [Replace Adjectives]
59: Replace "nice" to change sentiment to Negative?
60: > [bad , grim]
61:
62: [Replace Adjectives]
63: Replace "expensive" to change sentiment to Negative?
64: > []

```

User-assisted data augmentation was carried on 1000 sentences randomly selected from the training set. The selected annotator¹¹ of the SentiHood dataset assisted in the augmentation process.

We use a combination of the training data and the augmented data for training models based on SEQ and bag of n-grams representations. We do this for each category of augmented data separately and also on the combined augmented data. When training a logistic regression using the BoNgrams representation, we provide the model with all the sentences. When training the bidirectional LSTM model, at each iteration, we take all the sentences from the training set. We also sample sentences randomly from each of the selected generation category. The number of the selected augmented sentences equals to the number of sentences in the original training set.

¹¹The annotator with the highest inter-annotator agreements

6.7.2.1 Results

Figure 6.24 shows the performances of SEQ and BoNgrams representations on the test set when different categories of augmented data is added for training. SynAuto indicates the syntactic variation introduced through automatic augmentation. SynNoise is the group of user-assisted augmented sentences where fragments are added to a sentence or removed from a sentence. SemAdjectives refers to the category of augmented sentences that introduce lexical variation through a list of adjectives. The performance measure is the average AUC over aspect and sentiment classification. The AUC values are also averaged over all the aspects.

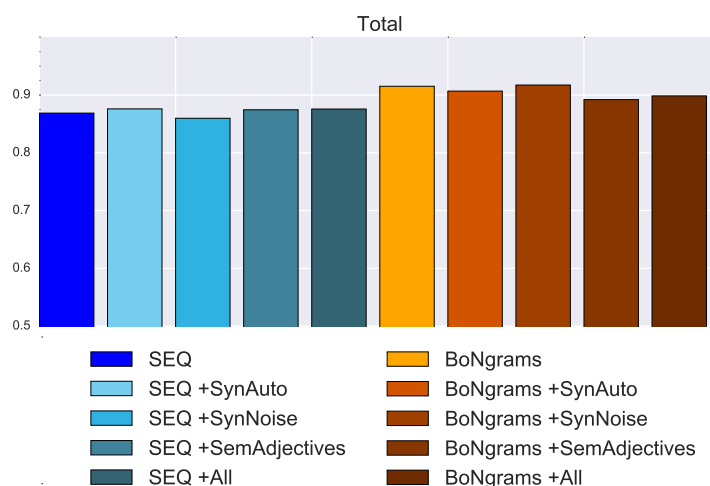


Figure 6.24: The overall performances of SEQ and BoNgrams representations on the test set when different categories of augmented data is added for training. AUC is averaged over sentiment and aspect classification over all the aspects. Note that the y-axis starts with 0.5.

The results indicate that the performance of the BoNgrams representation is superior to the SEQ representation, even after adding more training examples through data augmentation. The performance of SEQ representations can slightly improve (1% average AUC) when SynAuto data is added to the training set. Adding SynAuto data does not improve the overall performance of the BoNgrams representation. Adding SynNoise data improves the performance of BoNgrams representation by 1%. This can be because adding this category of augmented data makes the model more robust to the noise and irrelevant information in a sentence.

Figure 6.25 shows the breakdown of the performance of each representation on different categories of sentences in the test set. The first figure shows the results over the Single sentences. The last two figures show the performances of the representations over Multi-Agree and Multi-Disagree sentences. Adding SynAuto data slightly improves (1.5% and 1.1%) the performance of the SEQ representation on both categories of sentences with two locations. Note that SynAuto only contains sentences with multiple locations. This means that having a higher number of sentences with two locations in the training set can potentially help a model to generalise better on these type of sentences.

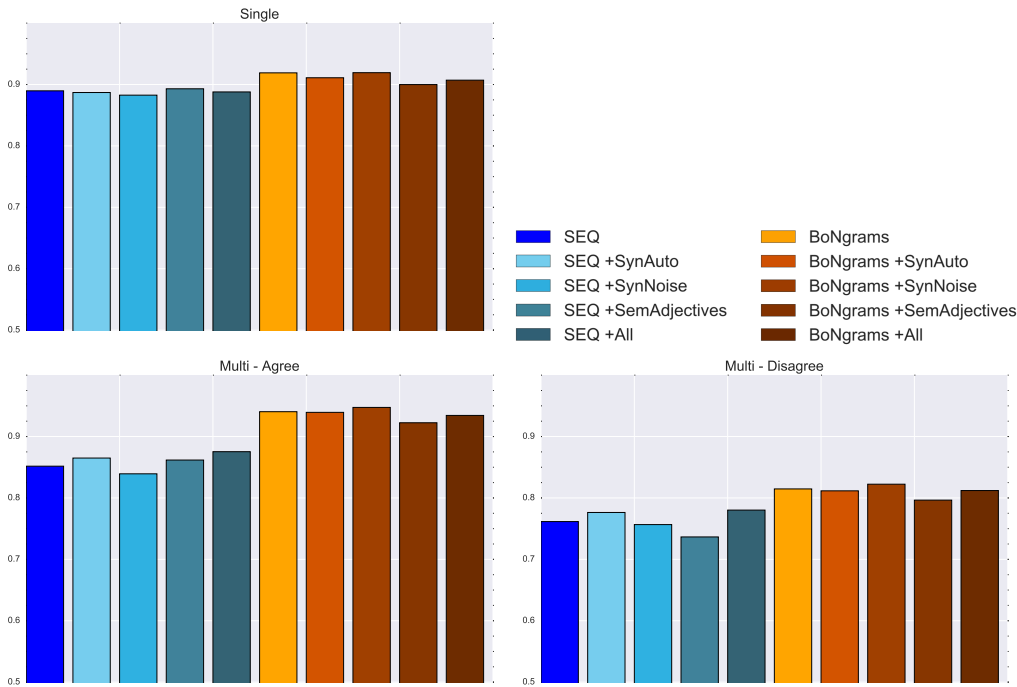


Figure 6.25: Performances of SEQ and BoNgrams representations on different categories of sentences in the test set when different categories of augmented data are added to the training data. Results are averaged over all aspects and over aspect and sentiment classifications. Note that the y-axis starts with 0.5.

We have further investigated whether utilising the augmented data has a positive impact on the prediction performances over the synthetic test categories. Results and graphs are included in Appendix C.2. As results indicate, the use of augmented data for training has a positive effect on the performance of our representations especially SEQ representation on most of the synthetic data

sets. The improvement for the SEQ representation is especially apparent (5%) on the Noise synthetic set. Augmented data increases the performance of the BoNgrams representation by 8% on the Multi-Disagree synthetic test.

The results in this section indicate that the performance of the BoNgrams representation remains superior to the SEQ representation on the test set in the presence of extra augmented data for training. This can be attributed to the two following explanations:

1. The test set does not contain much lexical variations outside of the training set. Moreover, the majority of sentences that contain two locations have agreeing sentiments (refer to Table 6.1) which can be captured in a more straightforward way using the bag of n-grams representation.
2. By simplifying sentences, sometimes, we create examples that are similar to the existing sentences in the training set. Consider the following sentence: “*target_loc is an extremely nice area, very central, but it can be expensive, unless you are fine with living in a match box!*”. By augmenting this sentence, we generate several examples including “target_loc is an extremely nice area” and “target_loc is an extremely beautiful area”. These two sentences are very similar to existing sentences in the training set (e.g. “target_loc is a nice area”, “target_loc is nice” and “target_loc is beautiful”) which do not add much variations on expressing Positive sentiments for the aspect *general*.

6.8 Discussion

In this chapter, we proposed methods for extracting opinion information in a targeted aspect-based sentiment analysis task applied on SentiHood dataset. We mainly focused on comparing the performances of representations that are learned sequentially or those who are defined on isolated parts of a sentence (bag of n-grams). These representations are based on generic architecture models or generic features, both of which do not need extensive task-specific engineering effort. To embed the necessary information, a representation should be able to

identify the context of a location in the sentence, as well as the relevant aspect and sentiment. We hypothesised that representations that are learned sequentially are better in embedding the necessary information that is needed to identify the aspect and its related sentiment for each location entity in the sentence. This is because natural language is expressed in a sequential way and sometimes, to understand the meaning of the sentence, the sequence of all the words in the sentence matters; something that in theory should be captured better using a sequential representation.

The results of our experiments show that the best performing proposed bag of n-grams representation is superior to the best performing proposed sequential representation when evaluated on the test set. However, the results on a synthetic set indicate that the sequential representation is better in capturing the meaning of the rare or unseen words and composition phenomena such as negation. The reason for sequential representations being able to handle the rare or unseen words is that they use dense word embeddings that are based on large corpora. The bag of n-grams representation can perform better on sentences where there is coordination structure (e.g. "*Location1 and location2 are safe places to live*"). We assume the reason for this inconsistency between the results on the test set and the synthetic data set is that our test set does not include many unseen words or many variations of negation. Moreover, the majority of the multiple location sentences in our test set contain similar opinions for both locations (Multi-Agree); something that a bag of n-gram representation can embed in a more straightforward way.

We further proposed data augmentation to help the models with learning more lexical and syntactic variations in the data. This can lead to models that generalise more and perform better on unseen data. Moreover, the success of models such as recurrent neural networks often heavily depend on the availability of a large amount of training data. Data augmentation is a cost-effective way of generating more examples. The performance of our representations on the test set using the additional augmented data, however, shows that data augmen-

tation improves the overall performance of the sequential representation by a small margin. This improved performance is still lower than the performance of the bag of n-grams representation. This can be attributed to the type of data available in the test set, as explained above.

We can now answer all the questions that we have raised in this chapter:

Q1: *Are sequential representations superior to the traditional bag of n-grams representations for addressing the task of targeted aspect-based sentiment analysis on SentiHood dataset?*

A1: Prediction results on the test set indicate that a bag of n-grams representation consisting of word uni-grams, word bi-grams and POS uni-grams can overall outperforms our proposed sequential representations.

Q2: *Which type of sentences are more suitable to be addressed using sequential representations compared to the bag of n-grams representations?*

A2: The results of our experiments on a synthetic dataset show that the sequential representations are better in capturing the meaning of rare or unseen words and composition phenomena such as negation. Moreover, the results on the test set indicate that the sequential representation (SEQ) can perform better on longer sentences, compared to the bag of n-grams representation (BoNgrams).

Q3: *Can generating more training examples through data augmentation improve the performances of representations, especially the sequential representations?*

A3: The performance of our representations on the test set using the additional augmented data shows that data augmentation improves the overall performance of the sequential representation SEQ by a small margin, i.e. 1%. This performance is still lower than the performance of the bag of n-grams representation, BoNgrams, when only trained using the training set.

In summary, the results of our experiments show that the hypothesis that we raised at the beginning of this chapter does not hold for the SentiHood dataset and our proposed sequential representations. This is somehow surprising as sequential representations and neural based models in general have shown significant improvements over the models that are based on bag of ngrams and hand-engineered feature representations in many NLP tasks. But the results of our intensive evaluations in this chapter indicate that the traditional bag of ngrams representations are superior to the sequential representations in targeted aspect-based sentiment analysis task. This can indicate that knowing the sequence of the entire sentence may not be necessary to solve some of the tasks in the field of NLP.

Chapter 7

Conclusion

This thesis was inspired by the need for understanding opinions of public expressed on social media platforms. We specifically aimed to extract information from the opinions expressed on community question answering platforms about neighbourhoods in a city. Community question answering platforms have not been used in the past for predicting real-world values such as characteristics of neighbourhoods. Further, extracting fine-grained opinion information has only been investigated from review-specific platforms. These platforms are not available for many entities such as neighbourhoods of cities. Therefore, in this thesis, we studied the strengths and the weaknesses of QA data in predicting characteristics and in extracting fine-grained opinion information for neighbourhoods. We focused on the QA platform of Yahoo! Answers¹ and neighbourhoods of London.

Throughout this thesis, we have shown that the language people use in Yahoo! Answers discussions reflects many characteristics of neighbourhoods. Moreover, we have demonstrated that fine-grained opinion information can be extracted for neighbourhoods using text from QA discussions. In this chapter, we first critically evaluate the limitations of our work. We then propose the future directions that can directly extend the research in this thesis. At the end, we provide a longer term research vision.

¹In the recent years, other QA sites such as Quora (<https://www.quora.com/>) have also become very popular. While QA sites share similar characteristics, whether they can all be used to predict characteristics of neighbourhoods with similar accuracies remain to be investigated.

7.1 Critical Evaluation

Availability of Yahoo! Answers Data In this thesis, we looked at extracting information for neighbourhoods, focusing on the city of London. London is a big cosmopolitan city and a popular destination for tourists and immigrants. This means that there are many discussions about its neighbourhoods on the QA platform of Yahoo! Answers. This is not always the case as we have shown for cities of Manchester and Birmingham.

Twitter Data Collection For this research, we collect Twitter data for a period of six months. Twitter related analysis and findings in this thesis are based on this data. Predictions or correlations using Twitter data might improve if the data is collected for a longer period of time.

Unification of Geographical Units To study the relations between demographic attributes and discussions on Yahoo! Answers, we were required to unify the geographical units in which both of these sources are available for. Demographics data is collected for geographical units with boundaries. Twitter data is geotagged which can be mapped into these geographical boundaries. However, Yahoo! Answers discussions do not include geographical information. We proposed a heuristics method based on the locality assumption which uses the distance between neighbourhoods and the units in which census data is aggregated for. The accuracy of this method could not be validated against the existing work since no previous research has used non-geotagged data with relation to the demographic attributes of neighbourhoods. We have, however, experimented with other heuristic methods which resulted in a less consistent outcome.

Time-Independency of Study In this thesis, the data that is collected from Yahoo! Answers spans over around five years. Twitter data was collected for the second half of 2015 and the demographics data is obtained from the last UK census in 2011. Our analysis does not consider that the nature of data on the platforms of Twitter or Yahoo! Answers may change in time. Equally neighbourhoods in cities evolve over time. Changes of characteristics of data and neighbourhoods over time has not been considered in this thesis.

Ethical Aspects of Using People’s Opinions for Predictions The work in this thesis is based on the assumption that the opinions collected from the community question answering platforms such as Yahoo! Answers are unbiased representations of the opinions of the people in those communities. Recently, there has been a rise in fake news and alternative facts targeting social media platforms and in the extreme case influencing the election results [128]. Twitter, for instance, has a large number of users that are bots. These bots can send spams, affect the opinion of the public, and contaminate the Twitter stream API [16]. While Twitter has been more prone to be affected by this phenomena in comparison to QA platforms, this trend can also affect these platforms in the future. Apart from the risk of getting influenced by fake or alternative news or bots, opinions expressed by people can also be uninformed, prejudiced or wrong. Identifying genuine and unbiased opinions has been out of the scope of this thesis. However, we encourage readers interested in applying the work introduced in this thesis to implement strategies for identifying such issues.

7.2 Future Work

We divide the future work of this research into the two following categories.

7.2.1 Opinion Aggregation

Leveraging Non-Textual Features In this thesis, we focused on using language in predicting aspects and attributes of neighbourhoods. As we observed, aspects such as *Quiet* cannot be predicted well using textual features. However, such an aspect can be predicted using metadata such as the number of tweets or users in an area. We also suspect that some other aspects can be best predicted using alternative sources of data. For instance, one can assume that the aspect *Eating Out* is correlated with the number of eateries and restaurants in an area.

Using Both Yahoo! Answers and Twitter In this thesis, we showed that Yahoo! Answers and Twitter are capable of predicting different types of characteristics which can be complementary to each other. The combination of text features from Yahoo! Answers and Twitter can be used to achieve more accurate predic-

tions on a wide range of aspects.

Spatial Prediction with Feature Labeling In predicting attributes and aspects of neighbourhoods, we explore methods for spatial prediction. We have shown that spatial prediction improves the accuracy of prediction for many attributes and aspects significantly. However, we have only done so when learning from labeled instances. Incorporating spatial properties of neighbourhoods into learning from labeled features can be explored.

7.2.2 Opinion Mining

Engineering More Sophisticated Features In this thesis, we have only considered bag of n-grams representations that are based on *generic* features that do not need any task-specific engineering efforts. As we have shown, these features outperform the sequential representations that are learned using generic LSTM architectures on the SentiHood test set. Adding more carefully designed features inspired by the existing work for the task of aspect-based sentiment analysis may improve the performance of the aspect and the sentiment predictions further.

Engineering More Sophisticated Architectures In this thesis, we have looked at a single-direction LSTM and two variations of bidirectional LSTM architecture. Instead of engineering more sophisticated features, we can engineer a more sophisticated model. For instance, we can add an attention mechanism that attends to specific parts of the sentence that is more likely to be the context of the target location.

Multiple Entities In this thesis, we looked at extracting aspect-based sentiment information from sentences that contain one or two locations. We have also observed that predicting the correct sentiment for the aspects of multi-location sentences is a harder task. Many sentences that appear in Yahoo! Answers discussions contain more than two location entities. The task of identifying the correct sentiment class for aspects of several locations present in a sentence can be more challenging.

Twitter While in this thesis, we only looked at extracting fine-grained opinion information from QA data, it was beyond the scope of this thesis to extract such

information from Twitter data. Recognising sentences that contain an opinion on aspects of neighbourhoods is one of the challenges of the process. Applying targeted-aspect based sentiment analysis to data from Twitter can be an extension of the work in this thesis.

7.3 Research Vision

Utilising Data from Community Question Answering Platforms Question answering platforms contain discussions from users in the community on a wide range of topics. These discussions are rich sources of information and opinions. Currently, the studies that utilise the QA data aim to provide improvements for QA platforms, such as finding the best answer or responder [9, 8]. QA discussions have not got much attention for making predictions about real-world phenomena and events. In this thesis, we have shown that such discussions can be used for predicting characteristics of neighbourhoods. Data from these platforms can be used by the research community for analysis and prediction in many other domains; especially domains that do not have a central review platform. An example of such a domain is the domain of automobiles. Even though, there are many sites that offer expert advice and reviews,² there is not a central platform for people to express their views and experiences with different cars. Other examples of domains that do not have their own centralised review platforms are service providers such as insurance companies and energy providers.

Decision Making for Neighbourhoods This thesis was inspired by the need for understanding neighbourhoods. To help a user with choosing the right neighbourhood, we proposed approaches for predicting characteristics of neighbourhoods from users' opinions and for fine-grained opinion mining. Even though knowing about characteristics of neighbourhoods can help with making decisions on suitable neighbourhoods, the burden of the decision making is still on the user. An intelligent agent can integrate more sources of information to make personalised recommendations. These sources can include information about

²For example <http://www.autoexpress.co.uk/car-reviews>

the places that user has lived and liked in the past, the needs and hobbies of the user, the presence of amenities (e.g. church, school, gym) in different neighbourhoods and more. This information can be used to find other neighbourhoods in the same city or in other cities across the world that are most suitable for the user.

Appendices

A Perceived Characteristics

A.1 Learning from Labeled Instances

Learning curves of the prediction performances using Twitter data when only labeled instances are used for supervision are provided in Figures 7.1 and 7.2. As figures show, aspects *Well Connected*, *Posh* and *Cultured* can reach high performances in the presence of around 5 positively and negatively labeled instances. Other aspects, especially *Multicultural* and *Eating Out* perform poorly even when we have the maximum number of labeled instances.

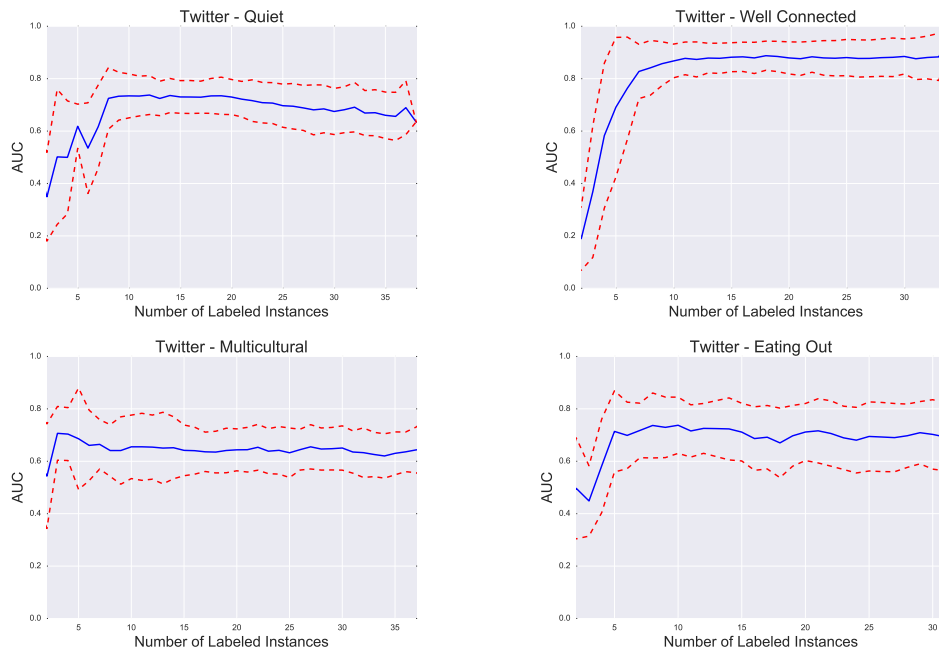


Figure 7.1: Learning curves of the performance in terms of AUC for selected aspects using **Twitter** data when we train a model using labeled instances.

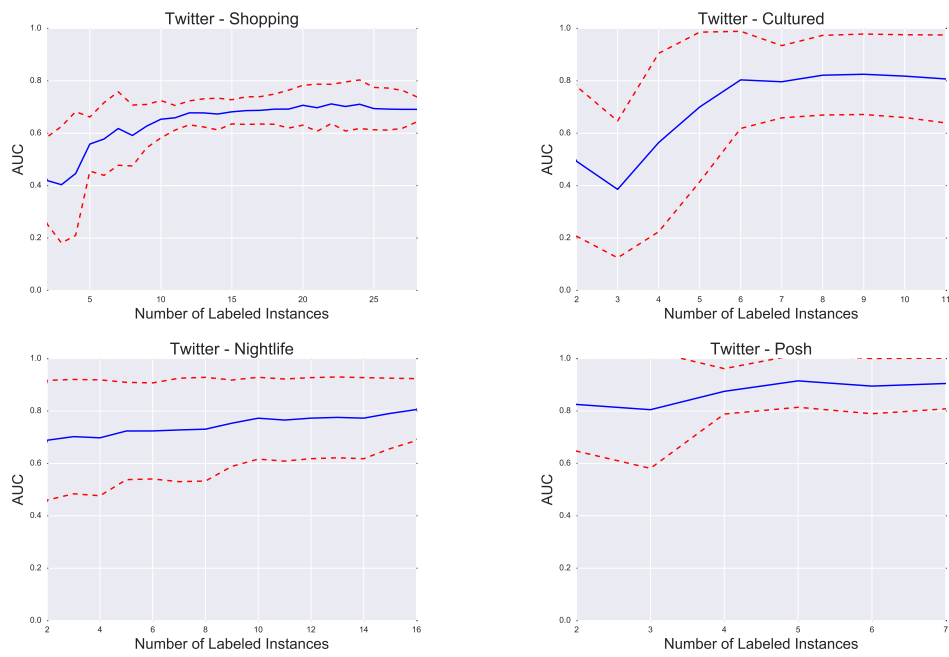


Figure 7.2: Learning curves of the performance in terms of AUC for selected aspects using **Twitter** data when we train a model using labeled instances (cont.)

A.2 Learning from Labeled Features

Feature Cost Analysis Figures 7.3 and 7.4 show the learning curves of the prediction performances in terms of AUC as the number of labeled features increases. This is for both methods of frequency score and GE on Yahoo! Answers data. AUC is reported on the test set only.

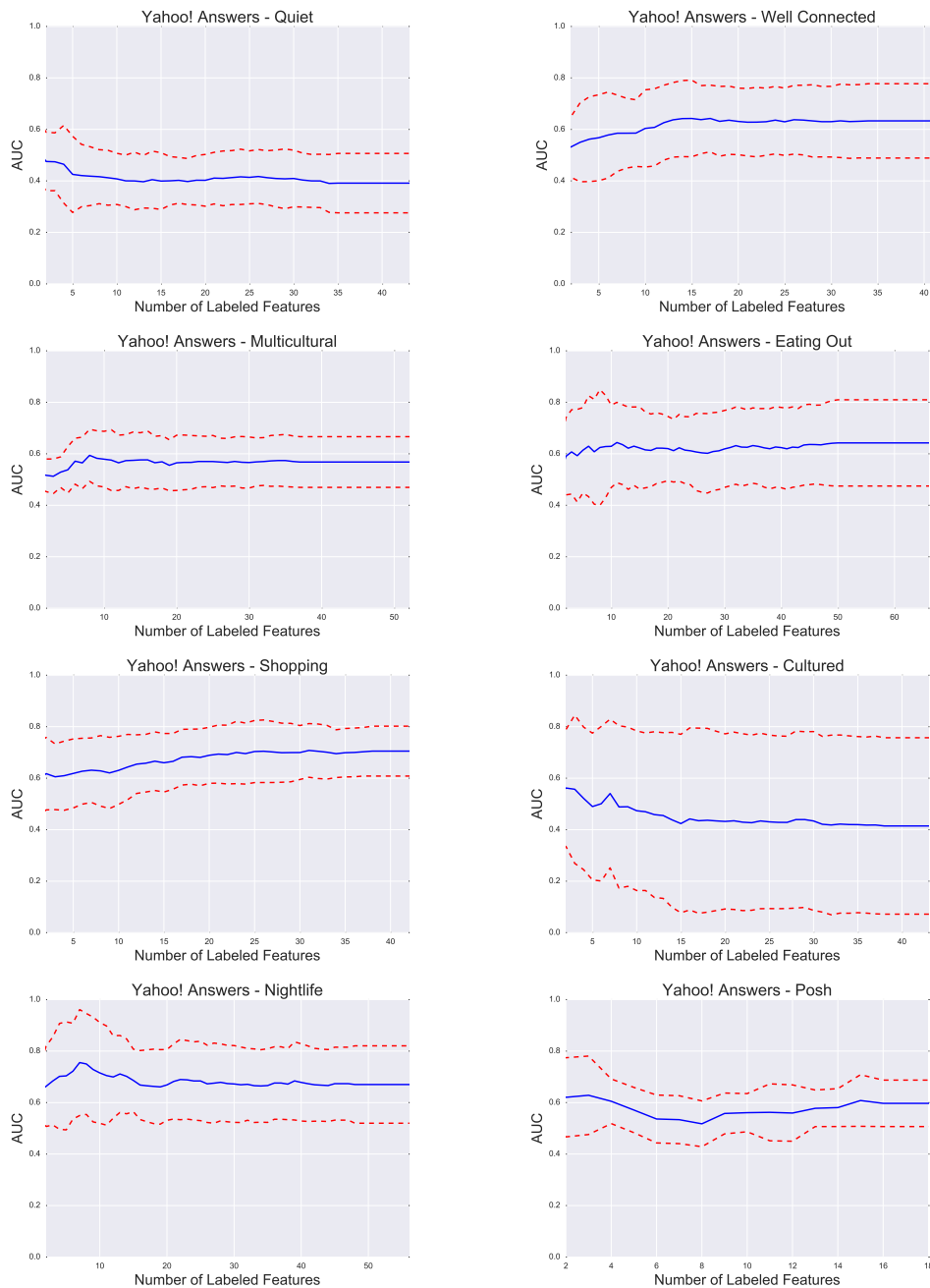


Figure 7.3: Learning curves of the performance in terms of AUC using **Yahoo! Answers** data and the frequency score (Test set only).

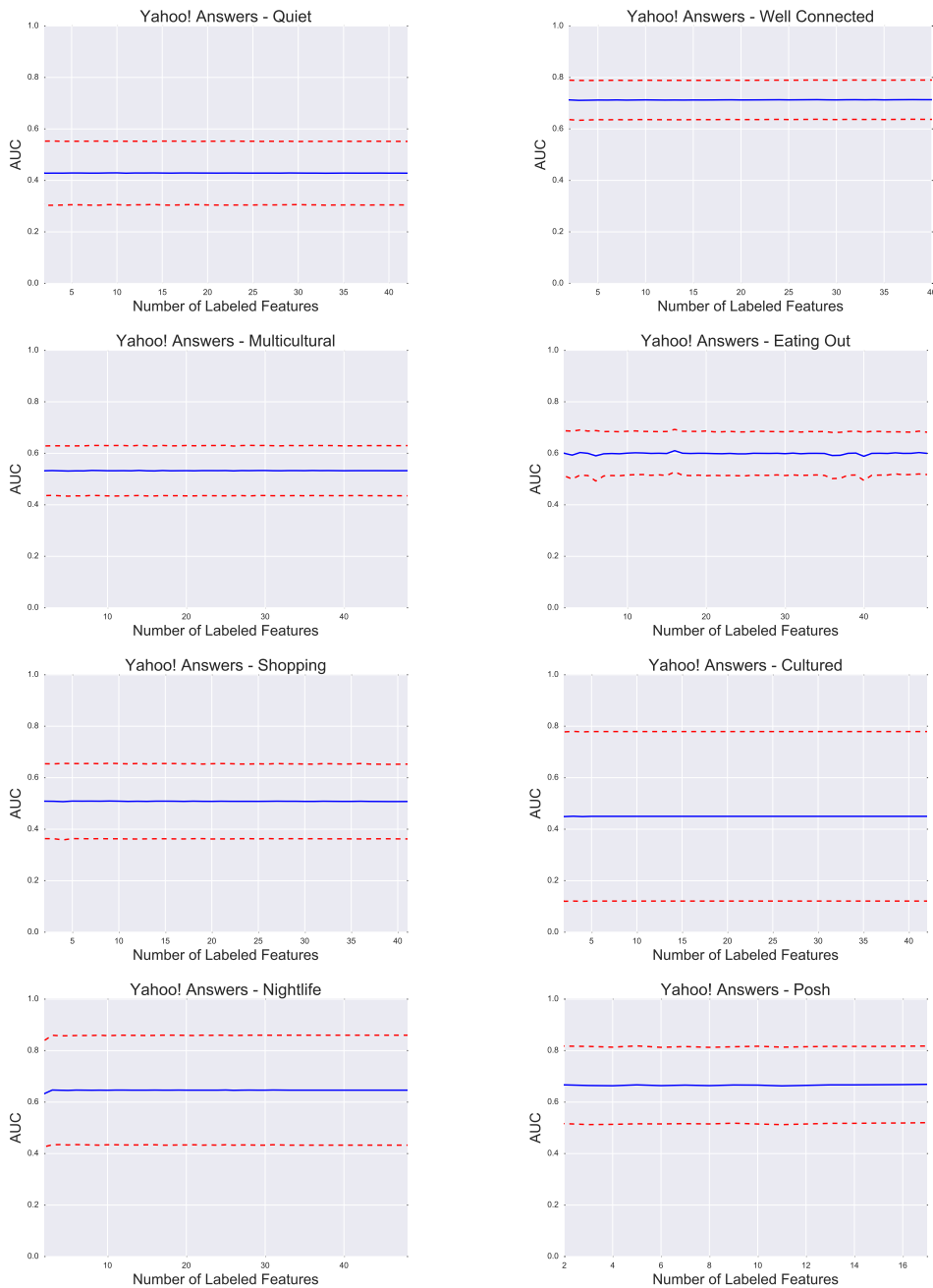


Figure 7.4: Learning curves of the performance in terms of AUC using **Yahoo! Answers** data and the GE model (Test set only).

In general, text features from Yahoo! Answers cannot reach a good performance using labeled features. We can observe that unlike the frequency score, GE can reach its optimum performance right away.

A.2.1 Prediction Results Over All Areas

In Chapter 4, we looked at the prediction performances using labeled features over the test set only. When learning from labeled features, we do not need any instances for training. Therefore, we can calculate the ranking metric of AUC over the entire dataset of neighbourhoods. We show the performance of both frequency score and the GE model on both data sources in Table 7.1.

Table 7.1: Prediction performances using labeled features applied on data from **Yahoo! Answers** and **Twitter**. The AUC is reported on *all areas*.

Aspect	Y! A Frequency	Twitter Frequency	Y! A GE	Twitter GE
Quiet	0.40 (0.00)	0.32 (0.00)	0.42 (0.00)	0.30 (0.00)
Well Connected	0.68 (0.00)	0.82 (0.00)	0.71 (0.00)	0.82 (0.00)
Multicultural	0.51 (0.00)	0.36 (0.00)	0.53 (0.00)	0.37 (0.00)
Good For Eating Out	0.63 (0.00)	0.60 (0.00)	0.65 (0.00)	0.63 (0.00)
Good For Shopping	0.67 (0.00)	0.69 (0.00)	0.54 (0.00)	0.69 (0.00)
Cultured	0.50 (0.00)	0.86 (0.00)	0.48 (0.00)	0.88 (0.00)
Good For Nightlife	0.72 (0.00)	0.85 (0.00)	0.67 (0.00)	0.87 (0.00)
Posh	0.57 (0.00)	0.67 (0.00)	0.56 (0.00)	0.79 (0.00)
Total	0.59 (0.00)	0.65 (0.00)	0.57 (0.00)	0.67 (0.00)

By comparing these results with the results in Table 4.9, we can see that the performance of Twitter is lower by 3% using the GE model and by 2% using the frequency score method when all the instances are used for evaluation. The performance of Yahoo! Answers remains the same. Overall, the performance of Twitter remains higher than Yahoo! Answers when all the instances are used for evaluation.

A.3 Learning From Labeled Instances and Features

Feature labeling is used on its own using only the domain knowledge or in combination with labeled instances. This is referred to as dual supervision in the literature [129, 130, 131] and is useful when we have access to very few labeled instances and want to improve the accuracy of the predictions.

A.3.1 Method

To use labeled features as supervision in a classification setting using logistic regression, we convert each labeled feature into a pseudo-instance as done in [131]. We then train a classifier on a combination of labeled and pseudo instances. A pseudo document has only one term, i.e. the labeled feature. The representation of the pseudo document for a labeled feature is shown in Figure 7.5. By using pseudo-instances, we encourage the parameters of the model to give higher weights to the labeled features as well as learning other terms that are relevant to the target aspect. The aspect label for each pseudo instance is positive. Note that *ind* is the index of the labeled feature in the vocabulary.

1	2	3	4	...	ind	...	0	0	0	...	$ V $
0	0	0	0	...	1	...	0	0	0	...	0

Figure 7.5: Representation of a pseudo-instance using a labeled feature.

A.3.2 Results

Classification results using labeled instances and labeled features are displayed in Table 7.2. We compare these performances with the results in Table 4.7 where only labeled instances are used for training. As we can see, adding domain knowledge does not increase the overall performance of Yahoo! Answers data. However, using labeled features increases the performance of Twitter by 2%. The main performance gains using Twitter data in this case are for aspects *Well Connected* (19%) and *Posh* (18%). On the other hand, predictions for some aspects such as *Multicultural* suffer when we add information regarding the labeled features.

Table 7.2: The results of the predictions using Yahoo! Answers and Twitter data when both labeled instances and features are used for training.

Aspect	Yahoo! Answers		Twitter	
	Instances	+ Features	Instances	+ Features
Quiet	0.65 (0.11)	0.64 (0.10)	0.63 (0.12)	0.66 (0.07) ↑
Well Connected	0.80 (0.11)	0.76 (0.11)	0.65 (0.31)	0.84 (0.11) ↑
Multicultural	0.86 (0.10)	0.74 (0.10)	0.81 (0.10)	0.64 (0.09)
Eating Out	0.80 (0.12)	0.74 (0.13)	0.80 (0.17)	0.61 (0.13)
Shopping	0.65 (0.15)	0.71 (0.11) ↑	0.62 (0.15)	0.68 (0.11) ↑
Cultured	0.81 (0.08)	0.77 (0.17)	0.65 (0.12)	0.67 (0.28) ↑
Nightlife	0.69 (0.27)	0.76 (0.16) ↑	0.84 (0.09)	0.76 (0.17)
Posh	0.70 (0.15)	0.67 (0.32)	0.71 (0.22)	0.89 (0.09) ↑
Average	0.74 (0.16)	0.72 (0.15)	0.70 (0.19)	0.72 (0.13) ↑

Cost Analysis We study the performance of predictions as the number of instances and features increases. This is to observe whether using labeled features can help when we have fewer labeled instances. In the case of Yahoo! Answers as the Figure 7.6 shows, improvements mainly occur as the number of instances increases (and not as the number of labeled features increases simultaneously). Labeled features help when no labeled instances are available (*Shopping* and *Nightlife*). For some other aspects, the performance decreases as the number of labeled features increases (*Posh*). Similar patterns can be observed when using Twitter data as illustrated in Figure 7.7. Overall, adding labeled features does not improve the performance when we have access to a few labeled instances. This is consistent for both Yahoo! Answers and Twitter.

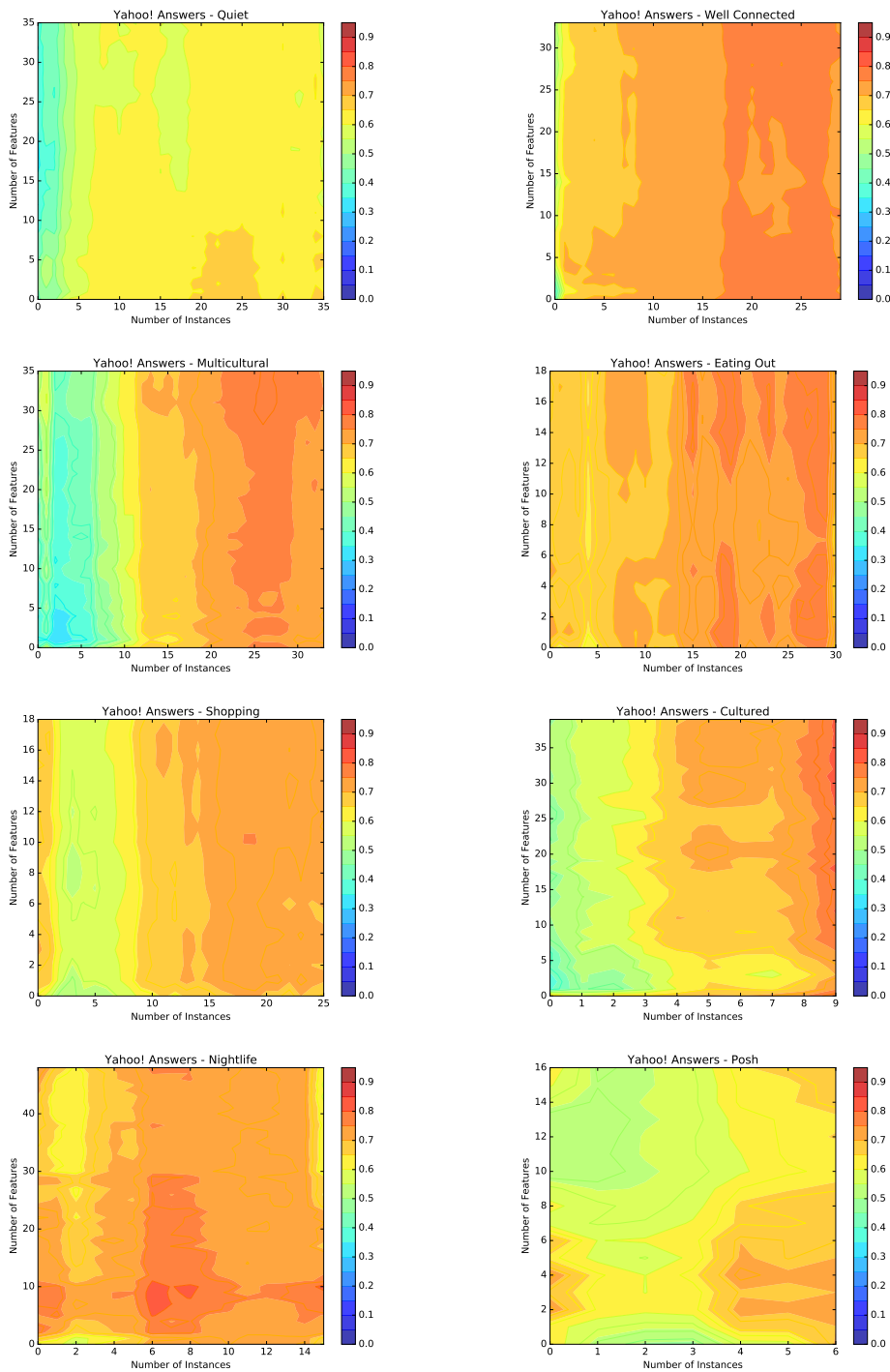


Figure 7.6: Contours of performance in terms of AUC for selected aspects using **Yahoo! Answers** data when we train a model using labeled features and instances.

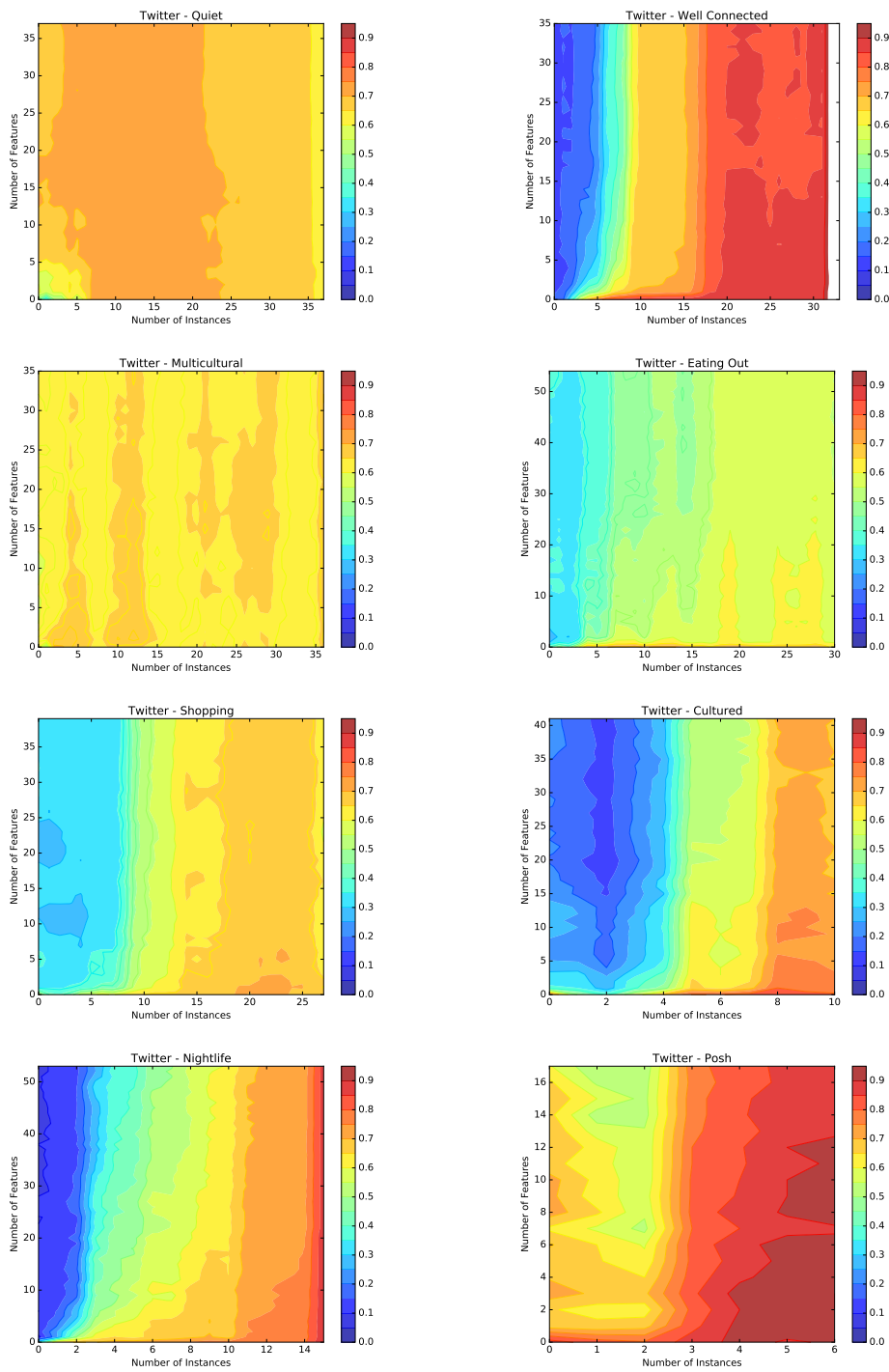


Figure 7.7: Contours of performance in terms of AUC for selected aspects using **Twitter** data when we train a model using labeled features and instances.

Twitter Metadata Table 7.3 shows the prediction performances of two metrics defined over Twitter data. These two metrics are the number of tweets associated with each neighbourhood and the number of unique users tweeted from a neighbourhood. Using both of these metrics, we can achieve the best performance so far for the aspect “Quiet” with AUCs of over 70%. These results show that for some aspects, metadata can be used complementary to text features. The number of users is also a good indicator for aspects such as “Well Connected”, “Cultured” and “Nightlife”. This can be because people log their activities on Twitter when they go out or take part in cultural events. This is also true for big commute hubs. People often use their smartphones and social media platforms when they are waiting for their bus, tube or train to arrive. On the other hand, aspects such as “Posh” and “Eating Out” cannot be predicted well using any of these metrics. The low performance is expected for the aspect *Posh*. This aspect is not related to activities of people and therefore it is harder to conclude whether an area is more likely to be *Posh* if there are more users visiting the area or tweeting from the area. The low performance for the aspect *Eating Out* is surprising. We expect people to tweet when they are out for dining or talk about what they are eating.

Table 7.3: Predicting aspects using metadata from **Twitter**

Aspect	Number of Tweets	Number of Users
Quiet	0.74(0.04)	0.75(0.07)
Cultured	0.87(0.11)	0.89(0.12)
Shopping	0.7(0.13)	0.69(0.12)
Eating Out	0.45(0.24)	0.44(0.23)
Multicultural	0.59(0.11)	0.59(0.11)
Well Connected	0.88(0.07)	0.89(0.07)
Nightlife	0.85(0.06)	0.85(0.06)
Posh	0.17(0.14)	0.33(0.32)
Total	0.67(0.24)	0.69(0.24)

A.4 Beyond London

In this section, we investigate how well we can predict aspects for areas beyond the city of London when having access to a number of labeled instances. Since Spareroom only provides labels for London areas, we take labels from AirBnB as explained below.

Dataset

AirBnB³ provides labels for aspects of areas for over 542 areas in 20 cities across 13 countries. These cities include London, Paris, Los Angeles, Barcelona, Tokyo, etc. Note that the dataset covers mainly the big metropolitan cities of the countries around the world. The labels provided by AirBnB have some intersection with labels from Spareroom that we studied so far. These labels are *Quiet*, *Nightlife*, *Shopping*, *Eating Out* and *Well Connected*.⁴

We query Yahoo! Answers (similar to the existing dataset) for all the areas labeled by AirBnB. For each area, we make a document by combining all the retrieved QAs. We split each document into sentences and filter out all the areas that have less than 40 sentences. This leaves us with 427 areas, including 43 areas within London. Table 7.4 shows the common aspects and the number of areas labeled with these aspects, across all the cities.

Table 7.4: Shared aspects of AirBnB and Spareroom and the number of areas that are labeled with each aspect in AirBnB dataset.

Aspect	Number of Areas
Eating Out	260
Shopping	217
Nightlife	199
Well Connected	198
Quiet	187

Extending Labels to New Areas

In the first experiment, we include the areas of all the cities in both training and the test set. We cross-validate results by randomly choosing training and test sets

³<https://www.airbnb.co.uk/locations>

⁴In AirBnB they are referred to as *Peace and Quiet*, *Nightlife*, *Shopping*, *Dining* and *Great Transit*.

in each fold from all the areas. Results are shown in table 7.5. As we can see, for the given aspects, we can make predictions with an average AUC of 72%. This performance is comparable to the performance of predictions when applied to the areas of the city of London only. This can be an indicator that the language people use on online QA platforms has enough similarity for the model to generalise to new areas.

Table 7.5: Prediction performance in terms of AUC, cross validated over areas of several cities around the world.

Aspect	AUC
Eating Out	0.71 (0.04)
Shopping	0.63(0.05)
Nightlife	0.71 (0.06)
Well Connected	0.75 (0.04)
Quiet	0.81 (0.04)
Total	0.72 (0.07)

Extending Labels to New Cities (zero-shot learning)

Consider a scenario where we have annotated areas in several cities as in AirBnB. The aim is to label areas of a new city where no annotations or expert knowledge is available. This can be challenging as the language used to describe aspects of areas can vary across different countries and cities. We study this task by training the prediction models on areas of all the cities but one (leave one city out). We then report the prediction performance on the areas of the remaining city. We repeat this process for all the cities. We do this only for cities that have at least 20 areas in our dataset and five positive areas for the given aspects. Average predictions are displayed in table 7.6.

Table 7.6: Prediction performance in terms of AUC for areas of a new city (zero-shot learning), averaged over all cities.

Aspect	AUC
Eating Out	0.6 (0.14)
Shopping	0.56 (0.11)
Nightlife	0.65 (0.17)
Well Connected	0.64 (0.14)
Quiet	0.72 (0.14)
Total	0.64 (0.15)

As we can see, prediction performances decrease in a zero-shot learning setting where no areas of the city used in test have appeared in the training set. This is to be expected. Looking at the results for individual cities, we observe that the lowest results are for the areas of the city of Barcelona and the best predictions are for the areas of Tokyo. Prediction results for the aspects that are available for the areas of Tokyo are displayed in Table 7.7.

Table 7.7: Prediction performance in terms of AUC for aspects of areas of *Tokyo*.

Aspect	AUC
Quiet	0.89
Well Connected	0.87
Good for Nighlife	0.88

These results show that we can extend aspect labels to new areas and new cities with reasonable accuracies.

B Guidelines for SentiHood Annotations

To start the annotation procedure for the SentiHood dataset, the annotators were presented with a written overall guidelines. The guidelines are provided below.

The goal of this annotation task is to identify opinions expressed in a given sentence for specific entities (neighbourhoods) and their aspects. An aspect should be chosen from a predefined list. In particular, given a sentence, the task of the annotator is to identify the following types of information:

Location: Location is the entity (E) that the opinion is expressed for. Locations are surrounded by “**” symbol.

- ****Camden Town**** and ****Islington**** are both desirable areas to live.

Aspect: Aspect (A) is a label that is given to the opinion expressed for a location. Aspects will be chosen from a provided list. For each identified entity location, one or more (or none) aspects can be identified based on the context of the sentence they appear in. The aspect can be chosen from the following list: *general, live, safe, price, quiet, dining, nightlife, transit/location, touristy, artsy, shopping, studenty, green/nature, and multicultural*.

The aspect *general* describes a general feeling about an area without a specific aspect. Examples are:

- I love ****Balham****.
- ****Lewisham**** is pretty dreadful.

Aspect Term Expression: An aspect term expression (ATE) is an explicit reference (mention) to an aspect A of entity E. This reference can be one or more words, however shorter spans are preferred. This reference is uniquely identified by its starting and ending offsets. Examples are below:

- House prices are so high in ****Camden town****. {ATE: prices, A: price}
- Parts of ****Hackney**** is quite rough. {ATE: rough, A: safety}
- I recommend ****Balham**** for its great range of restaurants. {ATE: food, A: restaurants}

- I wouldn't walk alone at night in **Streatham**. {ATE: alone at night, A: safety}
- I really recommend **Islington**. {ATE: null, A:general}

Polarity: Each identified E#A pair in a sentence has to be given a polarity, from a set $P = \{\text{positive, negative}\}$.

- I love **Camden town**. {positive}
- Avoid **Peckham** at all costs! {negative}
- I recommend **Balham** for food. {positive}

C Targeted Aspect-Based Sentiment Analysis

C.1 Synthetic Evaluation Test - Predictions

Table 7.8 shows examples of sentences from the Lexical Variation synthetic set where correct or incorrect predictions have been made for the relevant aspect or sentiment categories. The table includes rows for Positive, None and Negative ground truth labels.

We hypothesise that the SEQ representation gives a high weight to the term “is” for the sentiment class Positive and therefore predicting many sentences as Positive where “is” is present in the sentence. This means that it cannot correctly label many Negative sentences.

Table 7.8: Examples of labeled sentences in the *Lexical Variation* synthetic test.

Lexical Variation			
Sentence	Label	SEQ	BoNgrams
target_loc is spectacular	Positive	☑	☑
target_loc is breathtaking	Positive	☑	☑
target_loc is enchanting	Positive	☑	☑
For great Indian food, go to target_loc	None	☑	☑
target_loc is great for touristy stuff	None	☑	☒
If you want delicious food, i recommend target_loc	None	☒	☒
target_loc is a disgusting area	Negative	☑	☑
target_loc is an absolute toilet	Negative	☑	☑
target_loc is ghastly	Negative	☑	☒
target_loc is hideous	Negative	☑	☒
target_loc is revolting	Negative	☑	☒
target_loc is a ghastly area	Negative	☑	☒
People in target_loc are very snobby	Negative	☒	☒

Table 7.9 shows examples of correctly or incorrectly labeled sentences from the Negation synthetic set. It seems that the SEQ representation gives a high weight to the term “not” for the sentiment class Negative and therefore predicting many Positive sentences as Negative (top).

Table 7.9: Examples of labeled sentences in the *Negation* synthetic test.

Negation			
Sentence	Label	SEQ	BoNgrams
target_loc is not that bad	Positive	✓	✓
target_loc is not too bad	Positive	✓	✓
target_loc is not bad	Positive	✓	✗
target_loc is not dirty	Positive	✓	✗
target_loc is not really ugly	Positive	✗	✗
target_loc is not appalling	Positive	✗	✗
target_loc is not too awful	Positive	✗	✗
target_loc is not safe	None	✓	✓
target_loc is not too expensive	None	✓	✓
target_loc is not great for shopping	None	✓	✗
target_loc is not very far	None	✓	✗
target_loc is not nice	Negative	✓	✓
target_loc is not a good area	Negative	✓	✓
target_loc is not really okay	Negative	✓	✓
target_loc is not the best area	Negative	✓	✓
target_loc is not really beautiful	Negative	✓	✗
target_loc is not that clean	Negative	✗	✓

Table 7.9 shows examples of correctly or incorrectly labeled sentences from the Noise synthetic set. The SEQ representation is in general more robust to the added noise.

Table 7.10: Examples of labeled sentences in the *Noise* synthetic test.

Noise			
Sentence	Label	SEQ	BoNgrams
target_loc by London standards is a great area	Positive	✓	✓
target_loc for example around the corner from St Pauls is gorgeous	Positive	✓	✓
In South London target_loc mentioned above is safe and spectacular	Positive	✗	✓
target_loc it is right along The Thames so it can make good use of the river	None	✓	✓
It's cool working in target_loc as I like being on the Victoria line	None	✓	✗
target_loc is busy because of the cricket ground but it's very pleasant	None	✗	✗
target_loc , as a general rule, is nasty	Negative	✓	✓
target_loc , where i shop, is horrible	Negative	✓	✓
target_loc on the hill, is ugly	Negative	✓	✗
target_loc is quite far out and horrible	Negative	✓	✗
target_loc , in the South, is dirty	Negative	✗	✓
I used to live in target_loc (London and before it changed its name) - it's now a slum	Negative	✗	✗

Table 7.11 shows examples of correctly or incorrectly labeled sentences from Multi-Agree synthetic set. The BoNgrams representation outperforms the SEQ representation in this category.

Table 7.11: Examples of labeled sentences in the *Multi-Agree* synthetic test.

Multi-Agree			
Sentence	Label	SEQ	BoNgrams
Location1 and target_loc areas are good alternatives as well.	Positive	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
target_loc and location1 areas are good alternatives as well.	Positive	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Location1 and target_loc isn't as bad as everyone thinks and the houses are beautiful.	Positive	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
target_loc and location1 isn't as bad as everyone thinks and the houses are beautiful	Positive	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
The safest boroughs across all crime categories are target_loc and location1	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
The safest boroughs across all crime categories are location1 and target_loc	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
target_loc and location1 are considered the 'nice' area's of London to live	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
location1 and target_loc are considered the 'nice' area's of London to live	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Avoid target_loc and location1, the rest is ok to good	Negative	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Avoid location1 and target_loc , the rest is ok to good	Negative	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
People are saying location1 and target_loc etc, but we warned that ain't all good	Negative	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
People are saying target_loc and location1 etc, but we warned that ain't all good	Negative	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
The areas of target_loc and location1 are full of crimes, I wouldn't recommend any of the two	Negative	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
The areas of location1 and target_loc are full of crimes, I wouldn't recommend any of the two	Negative	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Table 7.12 shows examples of correctly or incorrectly labeled sentences from the Multi-Disagree synthetic set where the SEQ representation outperforms the BoNgrams representation.

Table 7.12: Examples of labeled sentences in the *Multi-Disagree* synthetic test.

Multi-Disagree			
Sentence	Label	SEQ	BoNgrams
target_loc is nice, location1 is bad.	Positive	☑	☑
location1 is nice, target_loc is bad.	Positive	☑	☑
location1 is a bit of a dump, target_loc is slightly nicer	Positive	☑	☒
target_loc is a bit of a dump, location1 is slightly nicer	Negative	☑	☑
where I stayed in target_loc I thought it was nice however in south east river location1 can be nasty	Positive	☑	☒
where I stayed in location1 I thought it was nice however in south east river target_loc can be nasty	Negative	☑	☒
target_loc is great and location1 is also rather more cosmopolitan	Positive	☑	☑
location1 is great and target_loc is also rather more cosmopolitan	None	☑	☒
target_loc is a nice area which is very near to the nasty area of location1	Positive	☑	☑
Location1 is a nice area which is very near to the nasty area of target_loc	Negative	☒	☒
Mikey target_loc is a disgusting area and location1 isn't bad	Negative	☑	☑
Mikey location1 is a disgusting area and target_loc isn't bad	Positive	☒	☒

C.2 Synthetic Dataset with Augmented Data

In this section, we look at the performances of SEQ and BoNgrams representation on different categories in the synthetic dataset when adding different types of augmented data. The performance is measured using AUC, averaged over the aspect and sentiment detection. Note that the synthetic dataset is based on the aspect *general* only.

Single Location - Lexical Variation

This category of sentences contains rare or unseen words for expressing the presence of aspects and their sentiments. Figure 7.8 shows the performances of our representations using different categories of augmented data on the synthetic dataset with lexical variations. There is only one target location in the sentences in this set. As the figure shows, the performance of the SEQ representation improves, especially when SynNoise category of augmented data is used for training. The performance of the BoNgrams improves when the category SemAdjectives are used for training. This is because this category of data contains new unseen words, the type of information that cannot be captured using features such as word n-grams or POS tags in a bag of n-grams representation.

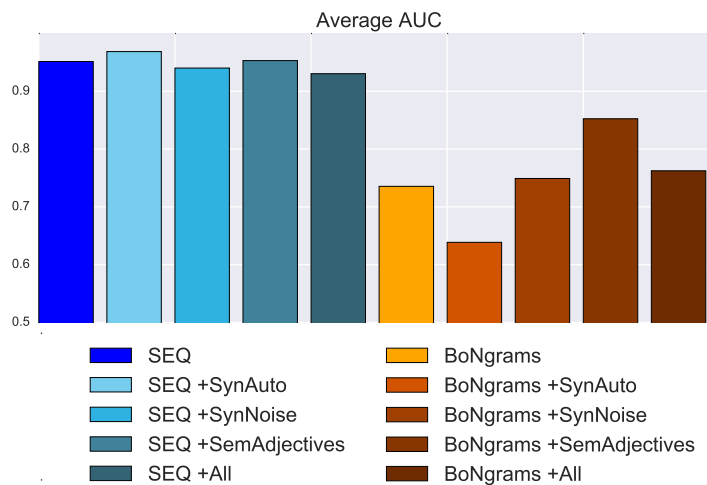


Figure 7.8: Results in terms of the average AUC (aspect and sentiment) on the synthetic set of *Lexical Variation* using SEQ and BoNgrams representations.

Single Location - Negation

This category of synthetic data has been created using variants of negation com-

position. Figure 7.9 shows the results of our representations using different categories of augmented data on the synthetic dataset representing negation. Results show that the performance of the SEQ representation can improve only slightly when all the categories of augmented data is utilised in training. It is surprising to note that some categories of augmented data result in a decrease in performances of both representations.

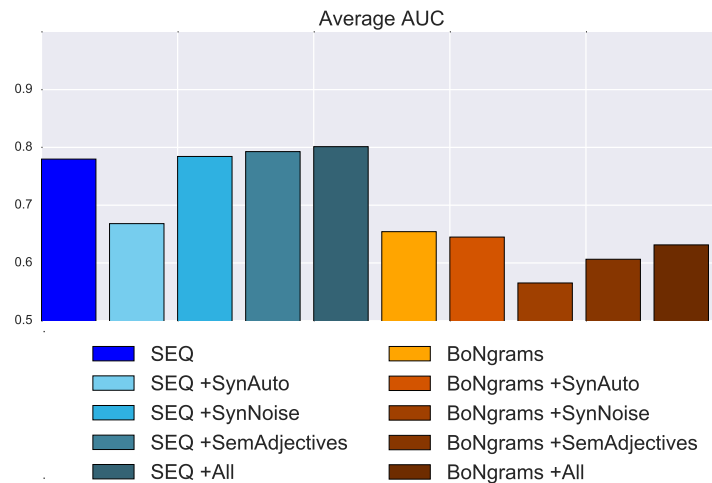


Figure 7.9: Results in terms of the average AUC (aspect and sentiment) on the synthetic set of *Negation* using SEQ and BoNgrams representations.

Single Location - Noise This category of synthetic data has been created by adding fragments of text to a sentence without affecting the opinions expressed towards the target location. As Figure 7.10 shows the performance of the SEQ representation improves by over 4% using the augmented dataset of SynNoise. This makes sense as training a model with more examples of noise (i.e. irrelevant information) in sentences makes the model more robust to noise. However, this effect is not present for the BoNgrams representation.

Multiple Locations - Agreement (Multi-Agree) In this category of synthetic sentences, there are two locations present in a sentence where both locations share the same sentiments towards the same aspects. It is often the case where two locations share the same context. As Figure 7.11 shows the performance of the SEQ representation improves when the category of SynAuto is used for training. This category contains sentences that are created automatically by concatenat-

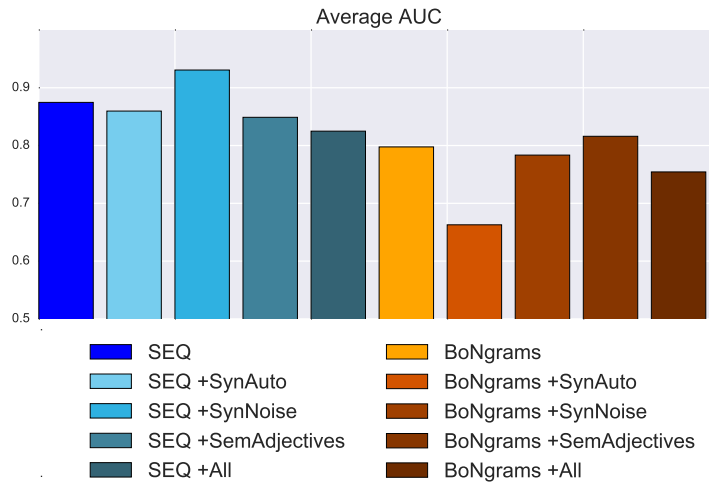


Figure 7.10: Results in terms of the average AUC (aspect and sentiment) on the synthetic set of *Noise* using SEQ and BoNgrams representations.

ing two existing sentences. The performance of the BoNgrams representation remains generally higher than the performance of the SEQ representation. Adding augmented data does not have much effect on its performance.

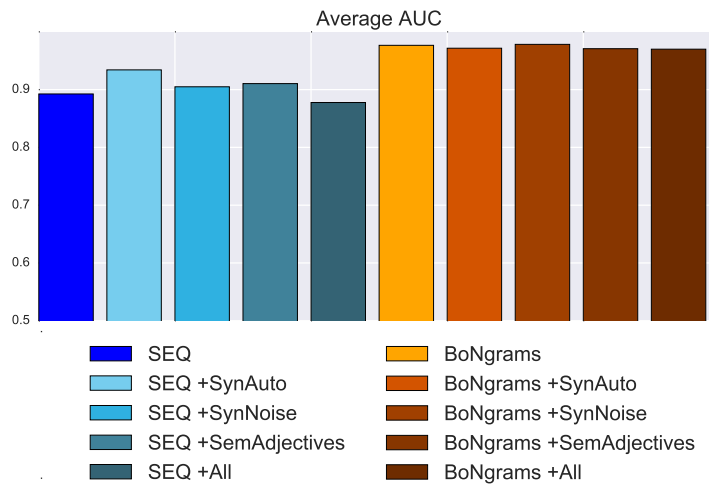


Figure 7.11: Results in terms of the average AUC (aspect and sentiment) on the synthetic set of *Multi-Agree* using SEQ and BoNgrams representations.

Multiple Locations - Disagreement (Multi-Disagree) This category of synthetic data contains sentences that express opinions for two location entities. The aspects and the related sentiments expressed towards these two location entities are not in agreement. Figure 7.12 shows the performances of our representations

on this synthetic dataset using different categories of augmented data. As results show, the BoNgrams representation benefits from adding the SynAuto dataset (and consequently all the augmented data). SynAuto contains sentences that are generated automatically by concatenating two existing sentences. This type of sentence resembles the examples in this synthetic dataset and therefore this improvement is to be expected. However, it is surprising that performance of the SEQ representation does not improve when this category of augmented data is used in training.

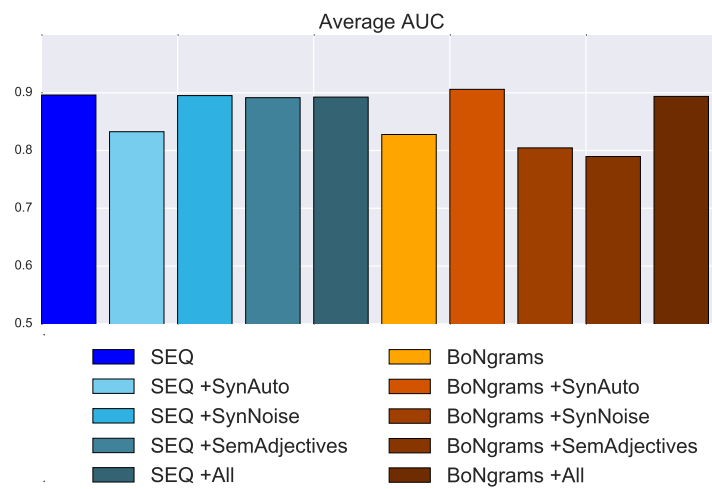


Figure 7.12: Results in terms of the average AUC (aspect and sentiment) on the synthetic set of *Multi-Disagree* using SEQ and BoNgrams representations.

Bibliography

- [1] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011.
- [2] Duy-Tin Vo and Yue Zhang. Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.
- [3] Himabindu Lakkaraju, Richard Socher, and Chris Manning. Aspect specific sentiment analysis using hierarchical deep learning. In *Proceedings of the Deep Learning and Representation Learning Workshop: Neural Information Processing Systems (NIPS)*, 2014.
- [4] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the AAAI International Conference on Knowledge Discovery and Data Mining (KDD)*, 2004.
- [5] Gayatree Ganu, Noemie Elhadad, and Amélie Marian. Beyond the stars: Improving rating predictions using review text content. In *International Workshop on the Web and Databases (WebDB): ACM SIGMOD/PODS Conference*, 2009.
- [6] Ting Liu, Wei-Nan Zhang, Liujuan Cao, and Yu Zhang. Question popularity analysis and prediction in community question answering services. *PloS one*, 2014.

- [7] Tong Zhao, Chunping Li, Mengya Li, Siyang Wang, Qiang Ding, and Li Li. Predicting best responder in community question answering using topic model method. In *Proceedings of the IEEE International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 2012.
- [8] Yuan Tian, Pavneet Singh Kochhar, Ee-Peng Lim, Feida Zhu, and David Lo. Predicting best answerers for new questions: An approach leveraging topic modeling and collaborative voting. In *Proceedings of the Workshops at the International Conference on Social Informatics (SocInfo)*, 2013.
- [9] Mingrong Liu, Yicen Liu, and Qing Yang. Predicting best answerers for new questions in community question answering. In *Proceedings of the International Conference on Web-Age Information Management (WAIM)*, 2010.
- [10] Shimon Kogan, Dimitry Levin, Bryan R Routledge, Jacob S Sagi, and Noah A Smith. Predicting risk from financial reports with regression. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2009.
- [11] Mahesh Joshi, Dipanjan Das, Kevin Gimpel, and Noah A Smith. Movie reviews and revenues: An experiment in text regression. In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2010.
- [12] Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welp. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the AAI International Conference on Web and Social Media (ICWSM)*, 2010.
- [13] Dean P Foster and Mark Liberman Robert A Stine. Featurizing text: Converting text into predictors for regression analysis.
- [14] Sitaram Asur, Bernardo Huberman, et al. Predicting the future with social media. In *Proceedings of the IEEE International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 2010.

- [15] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2011.
- [16] Juan Echeverria and Shi Zhou. The ‘star wars’ botnet with > 350k twitter bots. *arXiv preprint arXiv:1701.02405*, 2017.
- [17] Baichuan Li, Tan Jin, Michael R Lyu, Irwin King, and Barley Mak. Analyzing and predicting question quality in community question answering services. In *Proceedings of the International Conference on World Wide Web (WWW)*, 2012.
- [18] Dong Nguyen, Noah A Smith, and Carolyn P Rosé. Author age prediction from text using linear regression. In *Proceedings of the 5th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities: Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011.
- [19] Jacob Eisenstein, Noah A Smith, and Eric P Xing. Discovering sociolinguistic associations with structured sparsity. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011.
- [20] Daniel Preoțiuc-Pietro, Vasileios Lampos, and Nikolaos Aletras. An analysis of the user occupational class through twitter content. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.
- [21] Daniele Quercia, Jonathan Ellis, Licia Capra, and Jon Crowcroft. Tracking gross community happiness from tweets. In *Proceedings of the ACM International Conference on Computer Supported Cooperative Work and Social Computing (CSCW)*, 2012.
- [22] Dongsheng Wang, Abdelilah Khiati, Jongsoo Sohn, Bok-Gyu Joo, and In-Jeong Chung. An improved method for measurement of gross national happiness using social network services. In *Advanced Technologies, Embedded and Multimedia for Human-centric Computing*. 2014.

- [23] Daniel Preotiuc-Pietro and Trevor Cohn. A temporal model of text periodicities using gaussian processes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.
- [24] Shoko Wakamiya, Ryong Lee, and Kazutoshi Sumiya. Urban area characterization based on semantics of crowd activities in twitter. In *GeoSpatial Semantics*. 2011.
- [25] Anastasios Noulas, Cecilia Mascolo, and Enrique Frias-Martinez. Exploiting foursquare and cellular data to infer user activity in urban environments. In *Proceedings of the IEEE International Conference on Mobile Data Management (MDM)*, 2013.
- [26] Daniele Quercia, Diarmuid Ó Séaghdha, and Jon Crowcroft. Talk of the city: Our tweets, our community happiness. In *Proceedings of the AAAI International Conference on Web and Social Media (ICWSM)*, 2012.
- [27] Christopher Smith-Clarke, Afra Mashhadi, and Licia Capra. Poverty on the cheap: Estimating poverty maps using aggregated mobile communication networks. In *Proceedings of ACM International Conference on Human Factors in Computing Systems (CHI)*, 2014.
- [28] Neal Lathia, Daniele Quercia, and Jon Crowcroft. The hidden image of the city: sensing community well-being from urban mobility. In *Pervasive computing*. 2012.
- [29] Alessandro Venerandi, Giovanni Quattrone, Licia Capra, Daniele Quercia, and Diego Saez-Trumper. Measuring urban deprivation from user generated content. In *Proceedings of the ACM International Conference on Computer Supported Cooperative Work and Social Computing (CSCW)*, 2015.
- [30] Sanjiv Das and Mike Chen. Yahoo! for amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific Finance Association Annual Conference (APFA)*, 2001.

- [31] Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. Mining product reputations on the web. In *Proceedings of the AAAI International Conference on Knowledge Discovery and Data Mining (KDD)*, 2002.
- [32] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.
- [33] Richard M Tong. An operational system for detecting and tracking opinions in on-line discussion. In *Working Notes of the Workshop on Operational Text Classification: SIGIR Conference on Research and Development in Information Retrieval*, 2001.
- [34] Peter D Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.
- [35] Sentiment Analyzer. Extracting sentiments about a given topic using natural language processing techniques; jeonghee yi et al; ibm. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2003.
- [36] Kushal Dave, Steve Lawrence, and David M Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the International Conference on World Wide Web (WWW)*, 2003.
- [37] Bin Lu, Myle Ott, Claire Cardie, and Benjamin K Tsou. Multi-aspect sentiment analysis with topic models. In *Proceedings of the IEEE International Conference on Data Mining series (ICDM)*, 2011.
- [38] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. Semeval-2014 task 4: As-

- pect based sentiment analysis. In *Proceedings of the International Workshop on Semantic Evaluation: Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014.
- [39] Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeny Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the International Workshop on Semantic Evaluation: Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.
- [40] Margaret Mitchell, Jacqueline Aguilar, Theresa Wilson, and Benjamin Van Durme. Open domain targeted sentiment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.
- [41] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014.
- [42] Meishan Zhang, Yue Zhang, and Duy-Tin Vo. Gated neural networks for targeted sentiment analysis. In *Proceedings of the Association for the Advancement of Artificial Intelligence Conference (AAAI)*, 2016.
- [43] Yang Liu, Xiangji Huang, Aijun An, and Xiaohui Yu. Arsa: a sentiment-aware model for predicting sales performance using blogs. In *Proceedings of SIGIR Conference on Research and Development in Information Retrieval*, 2007.
- [44] Eldar Sadikov, Aditya Parameswaran, and Petros Venetis. Blogs as predictors of movie success. *Stanford InfoLab*, 2009.

- [45] Yancheng Hong and Steven Skiena. The wisdom of bookies? sentiment analysis vs. the nfl point spread. In *Proceedings of the AAAI International Conference on Web and Social Media (ICWSM)*, 2010.
- [46] Wenbin Zhang and Steven Skiena. Trading strategies to exploit blog and news sentiment. In *Proceedings of the AAAI International Conference on Web and Social Media (ICWSM)*, 2010.
- [47] Kin-Yip Ho, Yanlin Shi, and Zhaoyong Zhang. How does news sentiment impact asset volatility? evidence from long memory and regime-switching approaches. *The North American Journal of Economics and Finance*, 2013.
- [48] Saif M Mohammad and Tony Wenda Yang. Tracking sentiment in mail: how genders differ on emotional axes. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis: Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011.
- [49] Roy Bar-Haim, Elad Dinur, Ronen Feldman, Moshe Fresko, and Guy Goldstein. Identifying and following expert investors in stock microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2011.
- [50] Ronen Feldman, Benjamin Rosenfeld, Roy Bar-Haim, and Moshe Fresko. The stock sonar—sentiment analysis of stocks based on a hybrid approach. In *Proceedings of the International Conference on Innovative Applications of Artificial Intelligence (IAAI)*, 2011.
- [51] Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the International Conference on World Wide Web (WWW)*, 2008.
- [52] Abdulaziz Alghunaim. *A Vector Space Approach for Aspect-Based Sentiment Analysis*. PhD thesis, Massachusetts Institute of Technology, 2015.

- [53] Samuel Brody and Noemie Elhadad. An unsupervised aspect-sentiment model for online reviews. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2010.
- [54] Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the International Workshop on Semantic Evaluation: Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.
- [55] Gamgarn Somprasertsri and Pattarachai Lalitrojwong. Automatic product feature extraction from online product reviews using maximum entropy with lexical and syntactic features. In *Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI)*, 2008.
- [56] Ayoub Bagheri, Mohamad Saraee, and Franciska De Jong. Care more about customers: unsupervised domain-independent aspect detection for sentiment analysis of customer reviews. *Knowledge-Based Systems*, 2013.
- [57] Mary McGlohon, Natalie S Glance, and Zach Reiter. Star quality: Aggregating reviews to rank products and merchants. In *Proceedings of the AAAI International Conference on Web and Social Media (ICWSM)*, 2010.
- [58] Bing Liu. Sentiment analysis and opinion mining. *Journal of Synthesis Lectures on Human Language Technologies*, 2012.
- [59] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 2011.
- [60] Xiaowen Ding, Bing Liu, and Philip S Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, 2008.

- [61] Yorick Wilks and Mark Stevenson. The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation. *Journal of Natural Language Engineering*, 1998.
- [62] Taku Kudo and Yuji Matsumoto. A boosting algorithm for classification of semi-structured text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2004.
- [63] Alistair Kennedy and Diana Inkpen. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 2006.
- [64] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.
- [65] Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016.
- [66] Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archana Bhatia, Chris Dyer, and Noah A Smith. A dependency parser for tweets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
- [67] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011.
- [68] Tomáš Brychcín, Michal Konkol, and Josef Steinberger. Uwb: Machine learning approach to aspect-based sentiment analysis. In *Proceedings of the International Workshop on Semantic Evaluation: Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014.

- [69] Zhiqiang Toh and Jian Su. Nlangp at semeval-2016 task 5: Improving aspect based sentiment analysis using neural network features. In *Proceedings of the International Workshop on Semantic Evaluation: Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.
- [70] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the International Workshop on Semantic Evaluation: Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014.
- [71] Sebastian Ruder, Parsa Ghaffari, and John G Breslin. A hierarchical model of reviews for aspect-based sentiment analysis. *arXiv preprint arXiv:1609.02745*, 2016.
- [72] Yequan Wang, Minlie Huang, Li Zhao, and Xiaoyan Zhu. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.
- [73] Duyu Tang, Bing Qin, and Ting Liu. Aspect level sentiment classification with deep memory network. *arXiv preprint arXiv:1605.08900*, 2016.
- [74] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2001.
- [75] Bishan Yang and Claire Cardie. Extracting opinion expressions with semi-markov conditional random fields. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing (EMNLP) and Computational Natural Language Learning (CoNLL)*, 2012.
- [76] Ioannis Pavlopoulos. *Aspect based sentiment analysis*. PhD thesis, Athens University of Economics and Business, 2014.

- [77] Giuseppe Carenini, Raymond T Ng, and Ed Zwart. Extracting knowledge from evaluative text. In *Proceedings of the International Conference on Knowledge Capture (K-Cap)*, 2005.
- [78] Yejin Choi and Claire Cardie. Hierarchical sequential learning for extracting opinions and their attributes. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2010.
- [79] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, 2009.
- [80] Yohan Jo and Alice H Oh. Aspect and sentiment unification model for on-line review analysis. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, 2011.
- [81] Bishan Yang and Claire Cardie. Joint inference for fine-grained opinion extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2013.
- [82] Ivan Titov and Ryan T McDonald. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2008.
- [83] Zhiqiang Toh and Jian Su. Nlangp: Supervised machine learning system for aspect category classification and opinion target extraction. 2015.
- [84] José Saias. Sentiue: Target and aspect based sentiment analysis in semeval-2015 task 12. In *Proceedings of the International Workshop on Semantic Evaluation: Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.
- [85] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

- [86] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2014.
- [87] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Proceedings of the ISCA Conference on Interspeech*, 2012.
- [88] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and Phil Blunsom. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*, 2015.
- [89] Partha Niyogi, Federico Girosi, and Tomaso Poggio. Incorporating prior information in machine learning by creating virtual examples. *Proceedings of the IEEE*, 1998.
- [90] Robin Jia and Percy Liang. Data recombination for neural semantic parsing. *arXiv preprint arXiv:1606.03622*, 2016.
- [91] Noah A Smith and Jason Eisner. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the International Workshop on Semantic Evaluation: Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005.
- [92] Carl Edward Rasmussen. *Gaussian processes for machine learning*. Cite-seer, 2006.
- [93] Evelina Gabasova. GP. <http://evelinag.com/Ariadne/img/gp.png>, 2017.
- [94] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, 2013.

- [95] Adrian Benton, Margaret Mitchell, and Dirk Hovy. Multi-task learning for mental health using social media text.
- [96] Gerard Salton, Edward A Fox, and Harry Wu. Extended boolean information retrieval. *Communications of the ACM*, 1983.
- [97] David L Streiner and Geoffrey R Norman. Correction for multiple testing: is there a resolution? *CHEST Journal*, 2011.
- [98] Eric W Weisstein. Bonferroni correction. <http://mathworld.wolfram.com/BonferroniCorrection.html>.
- [99] Noel Cressie. The origins of kriging. *Mathematical geology*, 1990.
- [100] BP Shumaker and RW Sinnott. Astronomical computing: 1. computing under the open sky. 2. virtues of the haversine. *Sky and telescope*, 1984.
- [101] Daniel Preoțiuc-Pietro, Svitlana Volkova, Vasileios Lampos, Yoram Bachrach, and Nikolaos Aletras. Studying user income through language, behaviour and affect in social media. *PLoS ONE*, 2015.
- [102] Saharon Rosset, Claudia Perlich, and Bianca Zadrozny. Ranking-based evaluation of regression models. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2005.
- [103] Guido W Imbens and Tony Lancaster. Efficient estimation and stratified sampling. *Journal of Econometrics*, 1996.
- [104] GPy. GPy: A gaussian process framework in python. <http://github.com/SheffieldML/GPy>, since 2012.
- [105] Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2016.

- [106] Gregory Druck, Gideon Mann, and Andrew McCallum. Learning from labeled features using generalized expectation criteria. In *Proceedings of SIGIR Conference on Research and Development in Information Retrieval*, 2008.
- [107] Robert F Tate. Correlation between a discrete and a continuous variable. point-biserial correlation. *The Annals of mathematical statistics*, 1954.
- [108] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the European Chapter of the Association for Computational Linguistics Conference (EACL)*, 2012.
- [109] Mohammad Al-Smadi, Omar Qawasmeh, Bashar Talafha, and Muhannad Quwaider. Human annotated arabic dataset of book reviews for aspect based sentiment analysis. In *Proceedings of the IEEE International Conference on Future Internet of Things and Cloud (FiCloud)*, 2015.
- [110] Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William A Baumgartner, K Bretonnel Cohen, Karin Verspoor, Judith A Blake, et al. Concept annotation in the craft corpus. *BMC bioinformatics*, 2012.
- [111] Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Dan Sullivan, Chunhong Mao, Chunxia Wang, Bruno Sobral, Jun'ichi Tsujii, and Sophia Ananiadou. Overview of the id, epi and rel tasks of bionlp shared task. *BMC bioinformatics*, 2012.
- [112] Maria Liakata, Shyamasree Saha, Simon Dobnik, Colin Batchelor, and Dietrich Rebholz-Schuhmann. Automatic recognition of conceptualization zones in scientific articles and two life science applications. *Bioinformatics*, 2012.
- [113] Maria Liakata, Simone Teufel, Advait Siddharthan, Colin R Batchelor, et al. Corpora for the conceptualisation and zoning of scientific papers.

- In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2010.
- [114] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 1960.
- [115] Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. Pulse: Mining customer opinions from free text. In *Advances in Intelligent Data Analysis VI*. 2005.
- [116] Ayush Kumar, Sarah Kohail, Amit Kumar, Asif Ekbal, and Chris Biemann. Iit-tuda: Beyond sentiment lexicon: Combining domain dependency and distributional semantics features for aspect based sentiment analysis. In *Proceedings of the International Workshop on Semantic Evaluation: Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.
- [117] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2013.
- [118] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 1988.
- [119] Alex Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. PhD thesis, Technical University of Munich, 2012.
- [120] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 1990.
- [121] David Ha. LSTM. <http://blog.otoro.net/2015/05/14/long-short-term-memory/LSTM.png>, 2017.
- [122] Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. Stance detection with bidirectional conditional encoding. *arXiv preprint arXiv:1606.05464*, 2016.

- [123] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [124] Paul H Lee. Resampling methods improve the predictive power of modeling in class-imbalanced datasets. *International journal of environmental research and public health*, 2014.
- [125] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [126] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.
- [127] Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
- [128] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. Technical report, National Bureau of Economic Research, 2017.
- [129] Josh Attenberg, Prem Melville, and Foster Provost. A unified approach to active dual supervision for labeling features and examples. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*. 2010.
- [130] Hema Raghavan, Omid Madani, and Rosie Jones. Active learning with feedback on features and instances. *The Journal of Machine Learning Research*, 2006.

- [131] Vikas Sindhwani, Prem Melville, and Richard D Lawrence. Uncertainty sampling and transductive experimental design for active dual supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2009.