Technische Universität Kaiserslautern
Department of Computer Science

DISSERTATION

Assessment, Semantification and Applications of Sensor Data

Thesis approved by
the Department of Computer Science
of the Technische Universität Kaiserslautern
for the award of the Doctoral Degree

Doctor of Engineering
(Dr.-Ing.)

to

Hassan Issa

Date of Defense: November 30, 2018
Dean: Prof. Dr.-Ing. Stefan Deßloch

Reviewers:
Prof. Dr. Prof. h.c. Andreas Dengel
Prof Dr. Paul Lukowicz

D386

Abstract

The usage of sensors in modern technical systems and consumer products is in a rapid increase. This advancement can be characterized by two major factors, namely, the mass introduction of consumer oriented sensing devices to the market and the sheer amount of sensor data being generated. These characteristics raise subsequent challenges regarding both the consumer sensing devices' reliability and the management and utilization of the generated sensor data. This thesis addresses these challenges through two main contributions. It presents a novel framework that leverages sentiment analysis techniques in order to assess the quality of consumer sensing devices. It also couples semantic technologies with big data technologies to present a new optimized approach for realization and management of semantic sensor data, hence providing a robust means of integration, analysis, and reuse of the generated data. The thesis also presents several applications that show the potential of the contributions in real-life scenarios.

Due to the broad range, growing feature set and fast release pace of new sensor-based products, evaluating these products is very challenging as standard product testing is not practical. As an alternative, an end-to-end aspect-based sentiment summarizer pipeline for evaluation of consumer sensing devices is presented. The pipeline uses product reviews to extract the sentiment at the aspect level and includes several components namely, product name extractor, aspects extractor and a lexicon-based sentiment extractor which handles multiple sentiment analysis challenges such as sentiment shifters, negations, and comparative sentences among others. The proposed summarizer's components generally outperform the state-of-the-art approaches. As a use case, features of the market leading fitness trackers are evaluated and a dynamic visual summarizer is presented to display the evaluation results and to provide personalized product recommendations for potential customers.

The increased usage of sensing devices in the consumer market is accompanied with increased deployment of sensors in various other fields such as industry, agriculture, and energy production systems. This necessitates using efficient and scalable methods for storing and processing of sensor data. Coupling big data technologies with semantic techniques not only helps to achieve the desired storage and processing goals, but also facilitates data integration, data analysis, and the utilization of data in unforeseen future applications through preserving the data generation context. This thesis proposes an efficient and scalable solution for semantification, storage and processing of raw sensor data through ontological modelling of sensor data and a novel encoding scheme that harnesses the split between the statements of the conceptual model of an ontology (TBox) and the individual facts (ABox) along with in-memory processing capabilities of modern big data systems. A sample use case is further introduced where a smartphone is deployed in a transportation bus to collect various sensor data which is then utilized in detecting street anomalies.

In addition to the aforementioned contributions, and to highlight the potential use cases of sensor data publicly available, a recommender system is developed using running route data, used for proximity-based retrieval, to provide personalized suggestions for new routes considering the runner's performance, visual and nature of route preferences.

This thesis aims at enhancing the integration of sensing devices in daily life applications through facilitating the public acquisition of consumer sensing devices. It also aims at achieving better integration and processing of sensor data in order to enable new potential usage scenarios of the raw generated data.

to my parents

Acknowledgments

My sincere thanks go to my supervisor, Prof. Andreas Dengel, for giving me the chance to be part of the vibrant research environment in the Smart Data & Knowledge Services group of DFKI. Your ongoing guidance and feedback were vital at every stage leading to the completion of this thesis. I would like to also thank my friendly advisors Dr. Stefan Agen, Dr. Stephan Baumann and Mr. Ludger van Elst for their valuable advice, feedback and support which continued past the scopes of the projects we collaborated on.

I would like to further thank all of my colleagues and friends at DFKI whom I had the pleasure to share this dynamic working space with. Special thanks go to my office mates throughout the years: Michael, Bahaa, Takumi, Martin, Riaz and Gagan. Thank you all for making my working days more enjoyable and for the many fruitful discussions and fun conversations and activities we shared. During my PhD, I was part of three projects namely, MyCustomer, Anakonda and ServiceFactory. In these projects, I got to collaborate with several colleagues who shared their ideas, expertise and knowledge with me and provided me with much appreciated feedback and support. Thank you Andreas, Sara, Sebastian, Alexander, Christian and Tej. To the students I had the honour to supervise in their bachelor and master theses, seminar topics and projects, thank you so much for your valuable input and contributions. I particularly like to thank Alaa, Shary and Amir for their dedication and collaboration that remarkably enriched this work.

Outside of DFKI, I would like to thank all my friends in Kaiserslautern and Saarbrücken for being there for me and for the fun times and the endless activities we shared in the past few years.

To my lovely fiancée, Maryam, whom I was lucky to meet within my PhD years, your positivity and unmatched support kept me going through the roughest of times and I am looking forward for a life-time of adventures and success with you by my side. To my extended family, my uncles, aunts cousins and especially to my wonderful grandparents, thank you for your encouraging words, consistent motivation and hopeful prayers.

Last but not least, to my beloved family, I am so grateful for having you in my life, for enduring with me every difficult moment and enjoying with me every glimpse of success. Your support has always been key to me in my journey abroad so I would like to dedicate this thesis, in which I conclude my PhD journey, to you Mohammad, Zainab, Ali, Fatima, Mariam and mostly to my inspiring parents as a humble token of my appreciation and gratitude.

—Hassan

Contents

List of Figures

List of Tables

Introduction 1

In the recent years, technological advancements have led to an unprecedented adoption of sensors in both commercial systems and consumer products. For commercial systems, sensor integration nowadays is spanning a wide range of application sectors. In agriculture, data from wireless sensors is used to ensure optimal crop growth along with saving scarce resources like water [67]. In farming, the well-being of livestock is ensured using monitoring sensors [16]. In industry, modern factory machines collaborate to automate the industrial process [57]. In transportation, sensors are deployed in order to optimized the traffic flow reducing city traffic [2]. A more general application is seen with the holistic trend towards achieving smart urbanism with the so-called smart city concept [52]. Expectedly, such applications generate vast amounts of sensor data which, despite its benefits, impose serious usability, storage and management challenges. For example, 30 minutes of flight time in a commercial jet reportedly produces 10 TB of sensor data [21].

Similarly, consumer-oriented sensor devices cover a wide range of applications in various domains including health, sports and fitness, home automation and security to name a few. At the core of all of these devices lie the embedded sensors providing the main input to ensure the functionality of the device. Figure 1.1 includes examples from multiple categories of market sensor devices. The number of such connected devices is rapidly increasing following the technological trend labeled as the Internet of Things(IoT). This number is expected to rise from about 15 billion devices in 2015 to reach over 75 billion devices in 2025 according to IHS Markit[1] and in a less conservative forecast by Intel, 200 billion devices as soon as 2020[2]. Following this fast pace of product

[1]https://www.ihs.com/Info/0416/internet-of-things.html

[2]https://www.intel.com/content/dam/www/public/us/en/images/iot/guide-to-iot-infographic.png

FIGURE 1.1: Consumer-oriented sensor devices.

release, growing feature sets of the products and continuous updates of their firmware comes the huge challenge of providing a robust, dynamic and reliable means of evaluating these products. Assessing the features of these products facilitate their acquisition by consumers through aiding them to make educated personalized purchase decisions.

In this thesis, the two aforementioned challenges are addressed as follows:

- Rather than relying on standard product testing, this thesis introduces an end-to-end pipeline that applies sentiment analysis techniques to opinion rich customer reviews of market sensing devices and provides potential consumers with an easy to grasp summary of other customer's experiences with the products at the aspect level.

- By coupling semantic ontologies which enhance the usability of sensor data with big data technologies that enable storage and processing scalability, an approach for semantification and management for sensor data is presented to address both the usability and the management demands.

In what follows, I state the objective of this research in Section 1.1, list the thesis contributions in Section 1.2 and introduce the thesis organization in Section 1.3.

1.1 Objective

In this thesis, I try to answer the two following major questions regarding the expected advancements in sensor device usage and sensor data generation:

1. How to feasibly assess the quality of consumer sensor devices?
2. How can we manage and utilize vast amounts of generated sensor data?

Answering these questions enable consumers to make educated purchase decisions in order to acquire reliable consumer sensor devices and ensure scalable solutions for handling and using huge amounts of sensor data. In the following chapters I try to answer these questions by validating the following two hypotheses:

1. By using aspect-based sentiment analysis over customer reviews of market sensing devices, the quality of each feature of a device is feasibly assessed.
2. Coupling semantic technologies with big data technologies ensure the reusability of sensor data and the scalability of sensor data management systems.

1.2 Thesis Contributions

This thesis includes contributions on three levels, namely, conceptual level, implementation/practical level and application level. The contributions span aspect-based sentiment evaluation of sensor devices, semantification and management of sensor data, as well as applications of sensor data. An overview of the thesis contributions are given below:

1.2.1 Conceptual Contributions

C1 The first conceptual contribution is the usage of aspect-based sentiment analysis over customer reviews to evaluate market sensing devices. A complete aspect-based sentiment analysis pipeline is proposed (Chapter 4). The proposed unsupervised lexicon-based sentiment extraction outperformed both Bayesian and SVM sentiment classifiers trained on unigrams following [69].

C2 The second conceptual contribution is proposing an ontology-based encoding scheme for semantification and management of sensor data. This scheme leverages the in-memory capabilities of big data frameworks by persisting encoded TBox statements of a sensor network ontology in memory to achieve scalability in semantification and querying of raw sensor data. This scheme enhances reusability of sensor data through semantification and ensures scalability through big data storage and processing. (Section 6.4)

1.2.2 Practical Contributions

Practical contributions of this thesis include contributions that support the conceptual contributions listed above in addition to contributions needed to enable the application use cases.

C3 A novel combined sentiment lexicon of manually compiled and probabilistic lexicons. This combined lexicon is designed to utilize both the accuracy of the widely used manually combined lexicon in [42] and the completeness of the SentiWordNet probabilistic lexicon [34]. (Section 4.6.1)

C4 A new approach for handling negations. This approach uses several NLP techniques to achieve an accurate detection of the negation scope. This contribution enhances the overall performance of the sentiment analysis system proposed as negations denote the most occurring sentiment shifters. (Section 4.6.2)

C5 A smartphone-based prototyping approach for semantic sensor analysis systems. This approach uses an app for sensor data collection on a smartphone attached to an object of interest. The collected data is transformed into semantic sensor data using an ontology generated to describe the sensor data generation context. This approach requires very minimal setup efforts and specialized sensors, and the prototype can easily evolve into a reliable sensor network. (Section 6.5)

C6 A novel running track detection approach that uses a Bayesian classifier trained on GPS running-route data to identify standard running tracks. The classification relies on the convex shape of standard running tracks. This is reflected through features that determine the route's proximity to its convex hull. (Section 7.4)

1.2.3 Application Contributions

Application contributions of this thesis represent use cases, in which the above conceptual and practical contributions are applied to approach real-life challenges.

C7 A comprehensive evaluation of market leading fitness trackers using customer reviews. This evaluation utilizes the proposed sentiment analysis pipeline for sensing devices to process 3241 customer reviews and provide aspect-level insights and analysis for each device. (Section 7.2)

C8 A street quality assessment application. This application uses over 14 million semantic sensor data records collected through a smartphone embedded in a public transport bus to detect street anomalies in a city. Anomalies are detected using semantic rules on acceleration data followed by spatial clustering of the results. This application demonstrates the enhanced usability of sensor data following the proposed semantic sensor analysis systems prototyping approach. (Section 7.3)

C9 A preference-based running route recommender application. This application harnesses publicly available running sensor data to introduce a novel route filtering approach based on performance, visual, and nature of route features. A content-based route recommender is built on top of this filtering and is evaluated using feedback from active runners. (Section 7.4)

1.3 Thesis Organization

This thesis consists of eight chapters. Following the introduction, the organization of the thesis proceeds as follows:

- **Chapter 2** and **Chapter 3** introduce the technological background needed throughout this thesis.
  - **Chapter 2** lays the foundations for understanding aspect-based sentiment analysis systems. It introduces the different components comprising sentiment analysis systems and discusses the challenges handled by each of them. It further discusses state-of-the-art approaches used for each component. Additionally, basic concepts for methods and technologies used later in Chapter 4 and Chapter 5 are introduced for completeness and for the reference of the reader.
  - **Chapter 3** discusses background information on the realization and the management of semantic sensor data. An overview for related research is first introduced. Additionally, basic overviews for semantic and big data techniques and systems are presented. These are vital for understanding the approach for semantification and processing of sensor data presented in Chapter 6, and hence are included for reference.
- **Chapter 4** presents a complete pipeline for evaluating consumer sensor devices based on the aspect-based sentiment analysis of customer reviews of these devices. The pipeline features different components that enable

transforming the online reviews of sensor products into a user-friendly sentiment summary that conveys the insights of the sentiment analysis results to the user.

- **Chapter 5** presents an evaluation of the different components of the pipeline introduced in Chapter 4. The evaluation uses a dataset of customer reviews on three sensor devices.
- **Chapter 6** discusses the importance of semantic sensor data and introduces an approach that leverages big data technologies for better processing and management of semantic sensor data. It further introduces a prototyping approach for achieving semantic sensor data systems harnessing the multiple sensors incorporated in smartphones.
- **Chapter 7** presents multiple applications that display real-life use cases of the formerly presented approaches. It includes an evaluation of three market leading fitness trackers following the approach presents in Chapter 4. It also includes a street quality assessment application using the prototyping approach presented in Chapter 6. And finally, it includes a running route recommender application that shows the potential of harnessing openly available sensor data.
- **Chapter 8** presents an outlook for possible future work and concludes this thesis.

Sentiment Summarizer Background 2

By using aspect-based sentiment analysis over customer reviews of market sensing devices, this thesis addresses the problem of assessing the quality of market sensor devices. In this chapter, the foundations for understanding aspect-based sentiment analysis systems are presented. The different components comprising sentiment analysis systems are introduced and their challenges are highlighted. Additionally, this chapter discusses state-of-the-art approaches used for each component. For the sake of completeness, basic descriptions for baseline methods and technologies relevant to the proposed sentiment summarizer and its evaluation are included.

2.1 Introduction to Sentiment Analysis

Sentiment analysis, also commonly known as opinion mining, is an interdisciplinary research field that aims at determining the attitude of a person with respect to a certain topic, product or idea. It is a means of quantifying the general public's opinion, or in some cases a selected class of people's opinion, on that topic. Numerous subjective data sources are considered for this purpose including social media posts, surveys as well as written customer reviews which are used in Chapter 4 to evaluate market sensing devices.

Sentiment analysis research relies on and benefits from several other research areas such as natural language processing, information retrieval, statistics, data mining and machine learning. The insights provided by sentiment analysis proved to be valuable in many fields including political, commercial, social and health care applications.

Sentiment analysis can be applied to text on different levels, namely document, sentence and aspect/feature level. At document-level, the whole document's polarity is expressed as positive, neutral or negative. For larger documents with detailed discussions about different aspects of a product or mentions of different products for example, this level of granularity fails to convey the entirety of the author's opinion expressed through the document. Nevertheless, document-level sentiment analysis is often sufficient for use cases where the analyzed text is too small to discuss multiple opinions on different topics as in the case of microblogs and SMS messages. A more refined level of sentiment analysis is the sentence level where each sentence in the document is analyzed and its own polarity is computed. Though fairly fine, sentence-level sentiment analysis is not sufficient for the cases where multiple aspects of a product or multiple products are mentioned in a single sentence. In this work, sentiment analysis is applied on the most refined level i.e. the aspect level, which means, for every mention of an aspect of a product, the sentiment is extracted at this level. As an example, consider the sentence *"The Fitbit is very accurate in recording my steps but its battery drains extremely fast!"*. In this sentence, the author expresses a positive opinion about the step-counter aspect of the Fitbit product and a negative opinion about its battery-life aspect. Both mentions are considered when applying sentiment analysis at the aspect level.

The subjective nature of the data for which sentiment analysis is applied makes it a complex problem and performing sentiment analysis at the aspect level only adds to the complexity of this problem. Aspect-based sentiment analysis and its challenges has been addressed in related research including [72], [42], [12], [18] and in the bachelor thesis of Alaa Shafaee who contributed to this work [77]. In what follows, the main components comprising sentiment analysis systems are introduced in Section 2.2 through Section 2.4. Additionally, challenges that are needed to be overcome by these components are introduced, research related to these challenges is discussed, and brief introductions to technologies and approaches used in later chapters to handle these challenges are also included for the reader's reference. Finally, in Section 2.5, a brief introduction is given for the market-leading fitness tracker devices evaluated as a use-case for the proposed sentiment summarizer.

2.2 Product Name Extraction

An assumption often taken when applying sentiment analysis to text reviews of certain products is that every sentiment-baring sentence in the review is talking about this product. This assumption does not always hold as in many cases people tend to mention other competing products in order to compare them to the product under review. This necessitates the proper identification of the entity to which the sentiment should be assigned. Consider for example the following sentence where the negative sentiment should be attached to a mentioned competing product(Nike Fuelband) rather than the product in review(Jawbone Up): ”Jawbone Up is a stylish product unlike the annoyingly bulky Nike Fuelband.”

Extracting competing product names and their mentions in text is necessary to ensure assigning sentiments to the correct products. Product name extraction, though not widely discussed within the scope of sentiment analysis research, resembles the named entity extraction problem which is studied extensively in research.

Distributional similarity ([58] and [71]) is the classic approach of solving this problem. Given a list of seed entities and another of candidate entities for expansion, the approach ranks the candidate entities according to their likelihoods of belonging to the class by comparing the similarity of words surrounding the candidate entities with those of the seed entities. It is very possible that similar words occur around both the seed entities and candidate entities that do not belong to the class of interest such as ”syncing” around the ”iPhone” and fitness trackers like ”Fitbit One” which makes this approach inaccurate.

In [59], this problem is modeled as an entity-set expansion problem. The system is fed with a set of seed entities that share a certain class. Using a text corpus, the system expands the set to include more entities that belong to the same class. The entities selected as potential candidates for expansion are ranked based on their respective probabilities of belonging to the class of seed entities.

[58] and [71] present another approach for entity set expansion. The inputs for this approach are two lists of seed entities and of entities representing expansion candidates. Using word similarity measures, the similarities of words surrounding the candidate entities to the words surrounding the seed entities are computed, and each candidate entity is then assigned respective scores denoting the probabilities of it belonging to the same class of all seed entities. The limitations of this approach were discussed in [59] where an alternative approach is presented.

[59] introduces a machine learning model called positive and unlabeled learning (PU learning) to address the entity set expansion problem. This approach relies on the S-EM algorithm introduced in [61]. Based on the test results of 10 different text corpora, this approach is shown to significantly outperform the distributional similarity approach discussed above in addition to the Bayesian sets approach [40] also addresses the set expansion problem.

Since in the scope of this work, competing product names extraction relies on customer reviews. The available review meta-data in addition to NLP techniques are utilized to achieve high accuracy in product name extraction as shown in 5.3.

2.3 Product Aspect Extraction

One of the most challenging problems imposed by performing sentiment analysis at aspect level is to automatically extract product features or aspects. In many use cases, the product features are manually fed into the sentiment summarizer. However, considering the case where the summarizer would be applied to products with varying feature sets such as the broad category of market sensing devices used in this work, relying solely on manually fed product features is not enough. In what follows, Section 2.3.1 discusses state-of-the-art approaches for aspect extraction, Section 2.3.2 discusses the need for clustering words or expressions referring to equivalent aspects and Section 2.3.3 introduces WordNet, a lexical database used in the process of aspect clustering.

2.3.1 Aspect Extraction Related Work

Many related research such as [42] and some commercial tools consider explicit aspects to be frequent nouns and noun phrases. Nouns and noun phrases occurring beyond an experimentally determined threshold frequency are considered to be aspects. This approach relies on the assumption that the vocabulary used by people when commenting on different product aspects tend to converge. This requires a large number of reviews in order to cover all the product aspects which is not always the case. Additionally, although extracting the frequent nouns returns the features that are discussed frequently by people, it also returns many words that are not aspects. This is independent of the volume of reviews, even after applying some filters. The main reason noticed is that some of the general domain-related vocabulary also tend to converge. For example, for fitness trackers, many customers buy fitness trackers to get motivated to lose weight or to keep fit. Thus, words like ”weight” and ”fitness” sometimes occur more than some words referring to aspects.

[72] used filtering of irrelevant noun phrases to enhance the results. In particular, they computed the association degree between common noun phrases and some meronymy discriminators (e.g., ”of printer”, ”printer comes with”, ”printer has”, etc. for the Printer class) associated with the product class through their point-wise mutual information score (PMI) on the web. The PMI in natural language processing measures how two entities x and y tend to occur together within certain distance. In addition to the possibility of getting many irrelevant words, querying the web is very time consuming.

By adding a filtering step for irrelevant noun phrases, [72] enhanced the aforementioned approach. The degrees of association among common noun phrases and some meronymy discriminators such as ”of device”, ”device comes with”, and ”device has” for a certain ”device” class are measured. The measurement is based on their PMI score on the web (point-wise mutual information score).

Although effective, this approach has performance limitations due to the need for PMI computations.

Similarly, [5] enhanced this algorithm by removing stop words, sentiment-bearing words that are detected by using a sentiment lexicon, namely, Sentistrength [82] as well as removing any words included in a manually compiled dictionary for the application context. In this thesis, a similar approach is applied, however, pruning competing product names by utilizing the product name extraction component further increased the accuracy of the aspect extractor.

[12] presents an alternative approach for aspect extraction identifying aspects as nouns or noun groups occurring frequently in sentiment-bearing sentences or in certain syntactic patterns. Experiments has shown that the most successful pattern was that of nouns following adjectives as reported in [12]. The resulting list of candidate aspects is then filtered to exclude stop words and candidates occurring beneath a threshold frequency. Finally, the candidates aspects are stemmed and then ranked according to a manually tuned weighting scheme.

2.3.2 Clustering Equivalent Aspects

Clustering words referring to the same aspect of a product is a vital task for the sentiment analysis system. Because even when words are correctly identified as aspects, people often use different terminology or expressions to refer to a single aspect. Dealing with each such word as a different aspect heavily undermines the outcome of the system as the results which are referring to the same aspect are not properly aggregated. For example, all of the following three phrases are discussing the "price", however, none of them uses the same words to express the feature they all discuss:

- "The phone is quite *expensive*."
- "The *price* of this phone is high."
- "It *costs* twice as much of any other phone!"

In such cases, grouping equivalent aspects under the same category is vital for the correctness and the usability of the insights provided by the sentiment summarizer. In this work, lexical knowledge is utilized in order to group all the terms related to a single aspect under one cluster. WordNet, introduced next, is the lexical database used for this purpose.

2.3.3 WordNet:A large lexical database of English

WordNet [66] is a large lexical database for the English language. It groups words that function as different parts of speech into sets of synonyms called *synsets*(or synonym rings). A synset essentially is a group of semantically equivalent data elements and is widely used in the field of information retrieval. WordNet attaches a short definition of each word in a synset and maintains the semantic relations connecting each element in the synset.

Synonymy is the main relation among words in WordNet because it is the basis on which the words are grouped into synsets. It refers to words that can be used interchangeably in the same context. Synonymy is a symmetric semantic relation which means that if A is a synonym to B, then B is a synonym to A. The notion of Synonymy can be applied to all the different parts of speech as for the nouns ”car” and ”automobile”, the verbs ”hop” and ”jump”, the adjectives ”fast” and ”quick” and the adverbs ”happily” and ”gladly”.

Antonymy is another symmetric semantic relation in WordNet. Antonymy is the notion of opposing meanings between two words, especially useful with adjectives and adverbs since the concept of an opposite does not always apply in the case of nouns and verbs. Some examples of antonymy are the adjectives ”quick” and ”slow” and the adverbs ”happily” and ”sadly”.

Some further semantic relations included in WordNet are listed below:

1. **Nouns**
   - **hypernyms**: B is a hypernym of A if every A *is a* B (”vehicle” is a hypernym of ”car”)
   - **hyponyms**: Opposite to hypernyms, B is a hyponym of A if every B *is a* A (”car” is a hyponym of ”vehicle”)
   - **coordinate terms**: This relation denotes that the two words shared a common parent word. Formally, B is a coordinate term of A if A and B share a hypernym (”motorcycle” is a coordinate term of ”car”, as they both share the common hypernym, ”vehicle”)
   - **meronym**: B is a meronym of A if B is a part of A. For example, ”engine” is a meronym of ”car” as an engine is considered a part of the car)
   - **holonym**: Opposite to meronym, B is a holonym of A if A is a part of B (”car” is a holonym of ”engine”)
2. **Verbs**
   - **hypernym**: The verb B is a hypernym of the verb A if the activity A is a type of B (to perceive is an hypernym of to listen)
   - **troponym**: The verb B is a troponym of the verb A if the activity B is doing A in some manner (”to nap” is a troponym of ”to sleep”)

- **entailment**: The verb B is entailed by A if by doing the verb A you definitely must be doing the verb B ("to drive" is entailed by "to steer")
- **coordinate terms**: When two verbs share a common hypernym such as "to see" and "to hear" share the hypernym "to perceive"

The version of WordNet referenced in this thesis is WordNet 3.1[1] released on 09.11.2012. It contains 155,287 words organized in 117,659 synsets for a total of 206,941 word-sense pairs. In this work, WordNet database is used to facilitate grouping similar words under one category which is vital in the clustering of product aspects as explained in Section 4.5.2.

It is also worth mentioning that although WordNet only supports the English language, several other similar projects provide similar lexical databases for different languages and language groups. Some notable projects are:

- EuroWordNet [86], which includes support for many European languages.
- The Arabic WordNet [11], which supports the Arabic language.
- IndoWordNet [9], which supports the 18 scheduled languages of India.

2.4 Sentiment Extraction

Opinion words are vital in identifying the sentiment of a sentence or a phrase. The usage of "beautiful" and "ugly" in Example 2.4.1 demonstrates how such opinion words can express the polarity of the sentences in which they are used. However, in most of the cases, identifying the polarity of a phrase is much more complex. Examples of that are the subjective sentences written without the use of typical opinion words such as *" I will not buy this product again"* which clearly holds a negative sentiment despite the lack of opinion words.

Example 2.4.1 *The use of opinion words to express the sentiment of a sentence.*

- *This camera takes* ***beautiful*** *photos.*
- *Photos taken by this camera are* ***ugly****.*

In order to increase the accuracy of sentiment detection, a sentiment analyzer must handle many challenges that add to the complexity of the problem.

[1]http://wordnet.princeton.edu/wordnet/download/

Section 2.4.1 introduces a list sentiment extraction challenges. These challenges are later addressed by the proposed system as detailed in Section 4.6.

Sentiment extraction from text is generally approached either by supervised approaches, by unsupervised approaches, or by hybrid approaches. Supervised approaches require labeled training data which often need extensive manual labeling efforts. In addition, they are dependent on the domain of their training data which makes applying the classifier to a different domain than the training domain less effective. Unsupervised approaches, however, do not require any training data and can be applied to different domains with minimal tuning in order to achieve reliable results. Section 2.4.2 and Section 2.4.3 discuss unsupervised and supervised sentiment extraction methods respectively.

2.4.1 Sentiment Extraction Challenges

Below is a list of sentiment analysis challenges with clarifying examples that highlight possible occurrences in text. The examples used are taken from the field of fitness tracking for which the presented sentiment summarizer system is applied in Section 7.2.

1. Different part-of-speech (POS)

   Some words have different meanings and therefore different impact on the polarity of a sentence based on the different parts of speech they can take. To illustrate, consider the following examples where the words "like" and "pretty" clearly lose their positive sentiment influence with the change of their parts of speech.

   Example 2.4.2 *The effect of part-of-speech on the sentiment of a sentence.*

   - *I **like** the Nike Fuelband. [POS tagged: Verb]*
   - *This is useless for people **like** me. [POS tagged: Preposition or subordinating conjunction]*
   - *The Jawbone Up is **pretty**. [POS tagged: Adjective]*
   - *The Jawbone Up is **pretty** inaccurate. [POS tagged: Adverb]*

2. Position in sentence

   Even keeping the same part-of-speech, some words lose their impact on the polarity of the sentence just by taking a different position in the sentence. Example 2.4.3 shows such a case for the word *"well"*.

   Example 2.4.3 *The effect of word position on the sentiment of a sentence.*

- *Fitbit One does its job **well**.*
- ***Well**, I am not satisfied with the Fitbit.*

3. Word Sense Disambiguation

   Some words can carry different meanings even if while having the same part of speech and sentence position. Usually, the only way to differentiate the usage of such words is the context in which they are used. The word "questioned" in Example 2.4.4 highlights such differences wherein the first case it refers to asking a question which says nothing about the author's opinion while in the second case it signifies raising doubt about the accuracy of the device which in turn carries a negative sentiment about the device.

   Example 2.4.4 *The effect of word sense on the sentiment of a sentence.*

   - *I **questioned** the customer service.*
   - *I **questioned** the accuracy of the device.*

4. Verbs disabling the effect of sentiment words

   The use of sentiment-bearing words in a sentence with clear polarity does not always signify that the sentiment those words carry transfers to the sentence or the phrase as a whole. The use of phrases expressing wishes, suggestions, past thoughts and wondering serve as sentiment inverters and sometimes disable the sentiment carried by other words in the phrase. In Example 2.4.5, the usage of "thought" disables the negative sentiment imposed by the word "problem". Moreover, the usage of "wish" shifts the positive sentiment imposed by the word "reliable" to give a negative sentiment to the phrase as a whole.

   Example 2.4.5 *The effect of word sense on the sentiment of a sentence.*

   - *I **thought** there was a **problem** with my tracker.*
   - *I **wish** they had a **reliable** customer service.*

5. Domain context

   In different domains, the same word can be used to express a sentiment with different polarity. This is totally dependent on the context in which the word is used which leads to it being perceived differently by the reader in different contexts. In Example 2.4.6, the word "lose" is used in three different contexts carrying a positive, negative and neutral sentiments in the three different contexts.

   Example 2.4.6 *The effect of domain context on the sentiment of a sentence.*

- *With the Fitbit One, I could easily* ***lose*** *weight. [positive]*
- *You can easily* ***lose*** *the Fitbit One. [negative]*
- *I sync the Fitbit One with* ***Lose*** *It*[2]*. [neutral]*

6. Sentiment shifters

   Sentiment shifters (also called valence shifters)) are a class of words that invert the polarity of a phrase when used or intensify the sentiment carried by other words in the phrase. Below are some subclasses of sentiment shifters:

   a) Negations

      The usage of negations is very common in text and speech. Therefore, properly handling negations is a vital task for any sentiment analysis system to perform accurately. Phrases in which the negation word 'not' is used can vary in complexity making identifying the final polarity of the phrase a challenging task. Consider the two sentences in Example 2.4.7. In the first sentence, "not" shifts the positive polarity of the word "like" to mark the whole sentence as negative. Whereas in the more complex example in the second sentence, "not" shifts the positive polarity imposed by "succeed" but still the sentence as a whole holds a positive sentiment nevertheless. A novel approach for handling such complex cases with negations is introduced in Section 4.6.

      Example 2.4.7 *The effect of negation on shifting the sentiment of a sentence.*

      - *I do* ***not like*** *this product.*
      - *The man did* ***not succeed*** *in stealing my motivating fitness tracker.*

   b) Modal auxiliary verbs

      Modal auxiliary verbs are another class of words serving as sentiment shifters. In the English language, there are nine modal auxiliary verbs which are commonly used in text. These words are can, could, may, might, must, shall, should, will and would. In Example 2.4.8, the usage of "should" shifts the positive sentiment carried by "durable" to give a negative sentiment to the sentence as whole.

      Example 2.4.8 *The effect of modal verbs on shifting the sentiment of a sentence.*

      - *The device* ***should*** *have been more* ***durable****.*

[2]https://www.loseit.com/

c) Sentiment intensifiers and diminishers

Unlike the use of negations and modal auxiliary verbs that only shift the sentiment from positive to negative and vice versa, the use of sentiment intensifiers and diminishers also serve at imposing a stronger level of negativity or positivity on an already subjective phrase. In Example 2.4.9, three sentences are presented where in the first sentence the word ”extremely” intensifies the positive polarity carried by ”accurate”. In the second sentence ”barely” shifts the positive polarity carried by ”works” and conversely in the third sentence ”less” shifts the negative polarity carried by ”groggy”.

Example 2.4.9 *The effect of sentiment intensifiers and shifters on the sentiment of a sentence.*

- *The steps-counter is* ***extremely accurate!***
- *This device* ***barely works*** *for me.*
- *The silent alarm makes me feel* ***less groggy*** *in the morning.*

7. Sentence semantics and facts

Some sentences are perceived as subjective with a clear polarity even though they might not contain any opinion words. The polarity, in this case, relies on the background knowledge people have about the topic discussed. To illustrate, consider the two sentences in Example 2.4.10. Both sentences are written in an objective style listing facts that happened with the authors. However, knowing how the devices in both cases must properly perform, the facts listed in these sentences lead to perceiving the sentences as clearly negative.

Example 2.4.10 *The effect of sentence semantics on the sentiment of a sentence.*

- *The Fitbit did not count the five stairs I climbed up home.*
- *Black dots formed on the surface of my wristband.*

8. Conditional sentences and questions

Sentences with conditionals as well as interrogative sentences can be very challenging to handle when it comes to identifying the sentence polarity. This is due to the ability of disabling or inverting a sentiment imposed by words with a clear sentiment in those sentences. Example 2.4.11 shows several examples where the sentiment of opinion words is disabled or even shifted through the use of conditionals and interrogative forms.

Example 2.4.11 *The effect of conditionals on the sentiment of a sentence.*

- *If anyone has suggestions for me, that would be great! (Conditional, disables sentiment)*
- *If it had kept working, I would have recommended it. (Conditional, inverses sentiment)*
- *Can you tell me which fitness tracker is the best? (Question, disables sentiment)*

9. Sarcasm

   People often use sarcasm to express opinions in informal settings such as when writing a product review or through social media posts. Sarcasm entitles the use of exaggerated positive opinion words to express negative opinions and vice versa. Example 2.4.12 displays two such cases.

   Example 2.4.12 *The effect of sarcasm on the sentiment of a sentence.*

   - *Broken on its first day.. such a great product!*
   - *It would be awesome to charge us the double for nothing but their logo.*

2.4.2 Unsupervised Sentiment Classification

Two main approaches are usually considered to address the problem of extracting sentiment automatically [68]. The lexicon-based approach, introduced in this section, is an unsupervised approach that involves calculating the sentiment for a document or a sentence by aggregating the sentiment of words or phrases comprising the document. And the supervised sentiment classification approach, which involves building classifiers from labeled instances of documents or sentences as indicated next in Section 2.4.3.

For the lexicon-based approaches, the lexicon, which is a dictionary of positive and negative words, can be created manually or automatically. Manually crafted lexicons have human accuracy, however they are incomplete and do not consider cases where a word can reflect different sentiments depending on its position in the sentence. Manually crafted sentiment lexicons can be domain specific such as [26], which is used for classifying stock postings on an investor bulletin, and general, such as the Bing Liu opinion lexicon [60], which contains 6789 words and is used in this work in conjunction with an automatically generated sentiment lexicon.

Different methods are used to automatically generate sentiment lexicons. [42] and [51] developed a sentiment lexicon starting with a list of positive and negative seed words and expanding this list using WordNet, a lexical database for the English language. Synonyms and antonyms of the seed words were appended to the developed lexicon.

In [42], the sentence polarity was then determined by summing over the sentiment scores of the words comprising the sentence. In case the summation is zero, the sentence was assigned the sentiment score of the sentiment word that is the closest to the aspect mentioned in the sentence.

[51] computed the overall polarity of a sentence by multiplying the sentiment signs of sentiment-bearing words in the sentence. This approach is based on the simplifying assumption that ”negatives cancel each other out”. [51] also listed alternative computations for the overall sentiment using the harmonic and the geometric means of the sentiment words comprising the sentence.

[29] uses the same lexicon introduced by [42]. However, it presents a holistic approach that exploits linguistic conventions and external evidences of natural language expressions. This enables handling context dependent opinion words.

In this work, SentiWordNet [34] [4] which is also built using WordNet is used. SentiWordNet includes a total of 38,182 non-neutral words and is part-of-speech aware.

An alternative unsupervised sentiment classification approach is introduced in [84]. The polarity of a phrase is considered positive (resp. negative) if it has positive (resp. negative) associations. The association is computed as the difference between the point-wise mutual information (PMI) of the phrase and the words ”excellent” and ”poor” as shown in Formula 2.1. The algorithm achieved a reported average accuracy of 74% when applied to reviews in several domains. As mentioned earlier, the PMI computation impose performance limitations on this approach.

$$SO(phrase) = PMI(phrase, excellent) - PMI(phrase, poor) \tag{2.1}$$

[90] introduces a similar approach to the one in [84]. It detects the polarity of a phrase using a significantly larger list of seed words instead of only relying on the words ”poor” and ”excellent”. It also replaces the choice of PMI as an association metric and uses a modified log-likelihood ratio measure.

2.4.3 Supervised Sentiment Classification

Sentiment extraction can be approached as a supervised classification problem using machine learning techniques. Hence, several supervised machine learning techniques can be utilized to address this problem. The general supervised classification problem can be described as the problem of assigning unknown data items to categories(classes) based on a model developed through learning from labeled training data (data items for which the class is known). The algorithm used for the learning and classification is known as a classifier.

In the case of sentiment analysis, a data item can be a whole text passage, a sentence, a phrase or a single word depending on the desired granularity of sentiment extraction. A labeled data item usually comprises a polarity (positive, negative or neutral) or a sentiment score expressing the intensity of the polarity assigned to each item in the training dataset. However, in some applications, the sentiment classes take the form of actual expressions of emotions such as happiness, anger and fear.

In [69], three classification algorithms, namely, Naive Bayes, maximum entropy, and support vector machines (SVMs) were tested to identify sentiments using movie reviews as input data. The experiments including features based on unigrams and bigrams and the results showed that SVMs trained on unigrams outperformed the other two approaches.

In [5], a hybrid approach is presented where a lexicon-based sentiment score assigned to a sentence, specifically a Twitter post, is used as a training feature, among 15 other features, for an SVM classifier. The hybrid approach achieved a better result compared to both strictly supervised and lexicon-based methods. This approach assigns the sentiment for a sentence/document as whole, as is the case with most supervised approaches which is not favored for our intended use case where multiple aspects in a sentence are handled.

Using supervised methods usually assigns the sentiment for a sentence as a whole. However, matching the sentiment in the sentence to an aspect is possible by using parsing data in the feature vector to better determine the sentiment scope as shown in [13] and [48]).

In Section 5.5, the proposed sentiment extraction system is compared to two widely used classifiers, namely Naive Bayes classification and support vector machines. In both classifiers, the data items are represented using the bag-of-features approach. The selected features are commonly chosen to be single words(unigrams) such as "happy" or bigrams such as "really bad". Formally, let $F = \{f_1, ..., f_m\}$ be a set of m features, and let $n_i(d)$ denote the frequency of occurrence of f_i in data item d, Formula 2.2 presents the feature vector representation $\overrightarrow{d}$ of d.

$$\overrightarrow{d} := (n_1(d), ..., n_m(d)) \tag{2.2}$$

These two classifiers are widely used in sentiment analysis literature and previous work has shown that svms trained on unigrams can achieve high accuracy sentiment classification as shown in [69]. For the reference of the reader, brief introductions to Naive Bayes and SVM classifiers are given below.

2.4.3.1 Naive Bayes Classifier

The Naive Bayes classifier [39] is a probabilistic classification model based on Bayes theorem. Given a set C of classes such as positive, negative and neutral classes in the context of sentiment analysis, it assigns the most probable class $c \in C$ for each data item d. Naive Bayes assumes complete independence of the features of a data item. Formula 2.3 presents Bayes theorem in the classification problem context. $P(c|d)$ denotes the probability of d being in class c, $P(d|c)$ denotes the probability of generating item d given class c, and $P(c)$ and $P(d)$ denote the probability of occurrence of class c and item d respectively.

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} \tag{2.3}$$

Since $P(d)$ does not play a role in selecting the class, it is ignored. Substituting d by its feature vector representation and using the independence of feature assumption of the Naive Bayes classifier, $P(d|c)$ is defined in Formula 2.4.

$$\begin{aligned} P(d|c) &= P(f_1, ..., f_m|c) \\ &= P(f_1|c)...P(f_m|c) \\ &= \prod_{f_i \in F} P(f_i|c) \end{aligned} \tag{2.4}$$

In the training step, the values of $P(c)$ and $P(f_i|c)$ are estimated based on the labeled data items. And in the classification of each data item d, the probabilities $P(c_k|d)$ for all classes $c_k \in C$are considered and d is assigned to the most probable class C_{MAP} according to the *maximum aposteriori* or MAP decision rule as shown in Formula 2.5.

$$\begin{aligned} C_{MAP}(d) &= argmax_{c_k \in C} P(c_k|d) \\ &= argmax_{c_k \in C} P(c_k) \prod_{f_i \in F} P(f_i|c_k) \end{aligned} \tag{2.5}$$

Despite its simplicity and the assumption of strongly independent features which does not often hold in real-life scenarios, Naive Bayes classification performs well in classification problems and is often used as a baseline approach for such problems.

2.4.3.2 Support Vector Machines

Support vector machines [22] or SVMs are supervised learning models that are used to solve both regression and classification problems. Unlike Naive Bayes, SVMs are non-probabilistic linear classifiers that aim at finding a hyperplane that maximizes the separation between labeled data items belonging to two separate classes. SVM is hence considered a large margin classification approach. The relative position of any unlabeled data item with respect to this hyperplane determines its predicted class according to svms.

The term support vectors refer to the data items that are closest to the hyperplane. A simple illustration of a 2-d space support vector machine is shown in Figure 2.1.

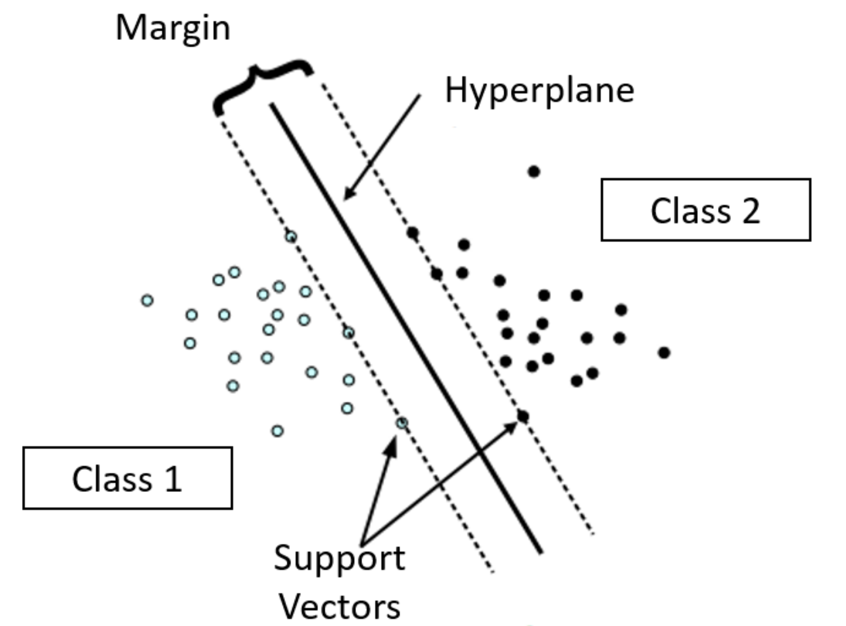

Figure 2.1: An illustration of a support vector machine in 2d-space

In the case of two classes such as a positive sentiment class and a negative sentiment class($c_i \in \{-1, +1\}$), the search for a hyperplane $\overrightarrow{w}$ is done by solving for α_i in order to optimize $\overrightarrow{w}$ in the dual optimization problem represented in Formula 2.6 where $\overrightarrow{d_i}$ represent the labeled data items.

$$w := \sum_i \alpha_i c_i \overrightarrow{d_i}; \quad \alpha_i \geq 0 \tag{2.6}$$

Even though the original svm is a linear classifier, the usage of kerneling [15] allows applying svms to problems where no clear hyperplanes are available. Kerneling is basically using a mapping function to map all the feature vectors to a higher dimensional space where separating the classes using a hyperplane becomes possible.

Using svms as sentiment classifiers is fairly obvious when considering only a positive and a negative class. However, when considering more than two classes, a modification for svm is needed in order to be able to classify unknown data items into more than two classes such as classifying a data item as "neutral" in the case of sentiment analysis. The approach used to enable multi-class classification is called the one-versus-all approach. Using this approach, a separate classifier is trained for each class. Each classifier aims at finding the whether an item belongs or not to a single specific class. In sentiment analysis context, this translates into training a positive classifier, a negative classifier and a neutral classifier. For each data item, each classifier computes whether this item belongs to its class. In case of an item belonging to multiple classes, the item is assigned the class whose classifier has the highest confidence score.

2.5 Fitness Sensor Trackers

Developing an aspect-based sentiment summarizer for the purpose of assessing features of market sensor devices is a novel contribution of this thesis. In Chapter 5, Amazon reviews of the top three market fitness trackers are used to evaluate the aspect-based sentiment analysis pipeline introduced in Chapter 4. What follows is an overview of the fitness tracking market and a brief description of the three devices used in the evaluation namely; Fitbit One[3], Jawbone Up[4] and Nike+ Fuelband[5]. Together, these devices dominated over 97% of the fitness trackers market in 2013 according to NPD Group[6].

Fitness trackers are always-on devices that provide basic pedometer functionalities including estimating steps taken, distance traveled and calories burned. In addition, these devices connect and sync their data easily to smartphone apps and web portals and enable social sharing and friendly competitions. Fitness and health trackers are constantly incorporating new sensors and algorithms to provide further health and fitness insights and functionalities for users. Samsung Simband 2[7] notably collects real-time biometric data including heart rate, blood flow and pressure, skin temperature, CO2 and oxygen levels as well as EKG[8] levels. Likewise, machine learning techniques are employed by some trackers like AMIIGO[9] fitness tracker which can detect over 100 exercises along with the number of sets and repetitions and the ability to learn new exercises by the user. The market size for wearable computing devices has

[3] http://www.fitbit.com/one

[4] https://jawbone.com/up

[5] http://www.nike.com/us/en_us/c/nikeplus-fuelband

[6] https://www.npd.com/latest-reports/consumer-technology-reports/

[7] http://www.samsung.com/us/globalinnovation/

[8] Electrocardiogram; spelled with a 'K' because in German it is spelled *Elektrokardiogramm*

[9] https://amiigo.com/

been rapidly growing and expected to attain a market size of $30 billion as forecasted by [8]. Accordingly, fitness and health trackers, which correspond to a big part of this growth, are expecting a 96 million unit shipment in 2018.

Figure 2.2 shows the three fitness tracker devices and below is a brief introduction to each one of them.

Figure 2.2: From left to right: Fitbit One, Jawbone Up and Nike+ Fuelband

2.5.0.1 Fitbit One

Fitbit one is a clip-on tracker that was announced on September 17, 2012. It features a digital display and it is the first tracker to sync through Bluetooth 4.0. Fitbit One supports Android, iOS and Windows Phone smartphone operating systems. It can record several daily activities including the number of steps taken, the distance travelled on foot, the number of floors climbed, calories burned, vigorously active minutes, as well as the quality of sleep. It also incorporates a "silent alarm" that wakes the user up through gentle vibrations. The launch price for Fitbit One was 99.99 U.S. dollar.

2.5.0.2 Jawbone Up

Jawbone up was initially announced in November 2011. Due to manufacturing problems, the product was relaunched in November 2012. The Up comes in the form of a waterproof wristband. It can track the number of steps taken, the distance travelled on foot, calories burned as well as the quality of sleep. It incorporates a vibration alarm feature and communicates to many 3rd party lifestyle and fitness apps and services. Jawbone Up supports Android and iOS through a native smartphone app. The launch price for Jawbone Up was 129.99 U.S. dollar.

2.5.0.3 Nike+ Fuelband

Nike+ Fuelband is a wristband tracker that was announced on January 19, 2012. Fuelband uses a set of LEDs as a screen. It tracks the number of steps taken and calories burned, it displays time and computes a proprietary measure of fitness activity, NikeFuel. Users set their daily goal in terms of NikeFuel, and the band displays the progress of the user in achieving their goal through an array of colored LEDs. The Fuelband syncs wirelessly with smartphones and both Android and iOS are supported through a native app. The launch price for Nike+ Fuelband was 149.99 U.S. dollar.

2.6 Summary

In this chapter, an overview of aspect-based sentiment analysis systems is presented. The different components of sentiment analysis systems, namely, product name extraction, product aspects extraction and sentiment extraction are introduced. For each component, an overview of its related work is presented and introductions to relevant technologies and approaches are given when needed.

The product name extraction problem is discussed in the context of named entity recognition in the related work. Most approaches use a group of entities as seed entities and use different methods to expand in order to include similar entities. Alternatively, in this thesis, the available review meta-data in addition to NLP techniques are utilized for product name extraction as discussed later in Section 4.4.

The product aspect extraction problem is generally approached through selecting the most common nouns in the dataset and applying different heuristics for pruning invalid words. A main challenge that follows aspect extraction is the grouping of equivalent aspects. This challenge is handled by utilizing lexical knowledge. For this purpose, WordNet, which is a lexical database for the English language is introduced.

As for the sentiment extraction component, a detailed list of its challenges is presented. The sentiment extraction problem is usually approached using either lexicon-based unsupervised methods or using supervised classifiers trained on labeled data. In this work, the approach chosen is lexicon-based to ensure usability for different device categories and additionally because supervised methods extract the sentiment for a sentence as a whole and not on the aspect level as is the aim of this work. Two state-of-the-art supervised approaches, namely, Naive Bayes and SVM Classifiers are compared to the proposed lexicon-based approach in Section 5.5 and hence they are introduced here for completeness.

Finally, this chapter provides an overview of fitness tracking devices, that are evaluated in a use-case for the proposed sentiment summarizer in Section 7.2.

Semantic Sensor Data Background 3

Coupling semantic technologies with big data technologies, this thesis addresses the need for scalable management and utility of vast amounts of sensor data generated in commercial and consumer contexts. This chapter discusses background information on the realization and the management of semantic sensor data. An overview for related research is first introduced. Additionally, basic overviews for semantic and big data techniques and systems are presented. These are vital for understanding the approach for semantification and processing of sensor data presented in Chapter 6, and are therefore included in this chapter.

3.1 Introduction

Semantification of raw sensor data has been addressed frequently in the past decade due to its significance. Realization of semantic sensor data helps achieving autonomous processing and reasoning over raw sensor data generated by complex sensor networks. It facilitates data integration through incorporating ontologies that describe other forms of semantic data. It also enables the reuse of sensor data for new applications as the complete generation context of the semantic sensor network is captured. These benefits, among others, have lead to the development of different ontologies that aim to model semantic sensor networks such as in [75] and [20].

Alongside the efforts towards achieving semantic sensor data, new technologies have been developed to enable scalable storage and processing of vast amounts of data by utilizing arrays of cheap commodity hardware. These technologies are often referred to as "Big Data" technologies and they have enabled handling huge amounts of data surpassing the capabilities of traditional data management systems.

In Chapter 6, a novel approach for realization and processing of semantic sensor data that harnesses the two aforementioned advancements is presented. This approach relies on the foundations of the ontological modeling of sensor networks and on the data management foundations of big data technologies as well.

In this chapter, research related to the semantification of raw sensor data and the processing of semantic data is first presented in Section 3.2. The technological backgrounds for both semantic technologies and big data technologies required for Chapter 6 are introduced in Section 3.3 and Section 3.4 respectively. Specifically, SSN ontology is used for the semantic modelling of sensor data, and Apache Spark is used for the management and processing of the data.

3.2 Related Work

This section discusses research related to semantic sensor data realization and processing. In Section 3.2.1, research describing ontologies built to model sensor networks is discussed. In Section 3.2.2, methods used for processing semantic sensor data are discussed. Finally, Section 3.2.3 introduces methods used for street quality assessment, which is the problem showcasing the usability of the smartphone-based prototyping approach for semantic sensor networks proposed in Section 6.5.

3.2.1 Ontologies for Sensors and Sensor Data

Modeling sensor networks and sensor data using ontologies has been addressed often in the recent years. Some ontologies describing wireless sensors appeared as early as 2004 [3], [31]. These ontologies however, did not focus on describing the sensor observations and only focused on describing sensor meta-data. Additionally, these ontologies did not comply with ontology design conventions that facilitate reuse and compatibility with reference ontologies.

More advanced models such as the OntoSensor ontology [75] and the SWAMO ontology [88] were based on the concepts of the Sensor Model Language (SensorML) Open Geospatial Consortium[1] (OGC). The OntoSensor is considered one of the most complete ontologies that describe sensor network meta-data. It contains concepts and properties such as "platform", "sensor", "sensor capabilities", "observable" and "measurement". Although included in SensorML concepts, OntoSensor does not describe the observation data. [50] introduced

[1] http://www.opengeospatial.org/standards/sensorml

several service extensions for the OntoSensor Ontology. Following the same OGC standards, [7] presents an ontology that describes the sensor observation. It however, does not describe the deployment and configurations of sensors.

Based on the advances in all these ontologies, the W3C Semantic Sensor Network Incubator Group[2], developed the Semantic Sensor Network Ontology [20] based on the stimulus-sensor-observation design pattern introduced in [46] and complying with the OGC standards. The SSN ontology is used as the basis for semantification of sensor data in this work and is further discussed in Section 3.3.

3.2.2 Processing of Sensor Data

Using manual user description of sensor deployment, [94] presents an approach that helps in transforming raw sensor data into RDF conforming to SSN ontology. This method is usable for general sensor data however for the proposed smartphone based prototyping approach introduced in Section 6.5 a fully automated transformation of sensor data is feasible as all the possible deployed sensors are known.

Apache Spark [93], which is the general cluster computing framework used in this work does not support semantic data natively. [23] presents an evaluation of RDF distribution algorithms using Apache Spark. In [24], an encoding scheme for ontology classes and properties using Spark is presented. This scheme however, imposes a restriction for single-inheritance ontologies. This is due to using an implicit hierarchy representation that uses binary prefixes to represent a super-class. This scheme, although efficient, cannot be applied in the case of the SSN ontology, used in this work, as it allows multiple inheritances.

3.2.3 Street Quality Assessment

In Section 7.3, we introduce a street quality assessment application that aims at detecting street imperfections such as potholes. This application is built to showcase the significance and reliability of the proposed smartphone-based prototyping approach for utilizing sensor networks. Although in this work, this use case is approached from a new perspective enabled by semantification of sensor data, a brief discussion of related work, from an application perspective, is presented.

Quality assessment has generally been approached through three major methods using sensor data, laser scanning and street imagery as well. In [65] and [32], similar approaches relying on analyzing accelerometer data through attaching accelerometer-equipped smartphones or dedicated acceleration sensors to vehicles. Another approach such as the one introduced in [91] uses laser

[2]https://www.w3.org/2005/Incubator/ssn/

street imaging to detect potholes and their shapes and classify them based on severity. Although this approach gave very accurate results, however it required laser scans to be performed from a still position on the pavement which is not very practical. In a similar approach but using still images rather than laser scans, [54] presents an approach that detects potholes by comparing potential defect shapes in images and comparing them to their surroundings.

3.3 Semantic Data and Ontologies

An ontology [37] represents a formal explicit description of a certain domain of knowledge. Ontologies aim at facilitating knowledge understanding and sharing among different users including human users and software agents. Typically, ontologies are developed by experts in a certain domain and the resulting ontology is then used and further extended by different users.

The basic components of an ontology are:

- **Classes**: A class denotes a concept or a kind that represents different instances or other sub-classes. An example class is "Movement Sensor" which represents subclasses such as "Accelerometer" and "Gyroscope" and also can represent specific instances of the class such as "Movement Sensor 001".
- **Instances**: An instance denotes a specific individual item belonging to one or more of the ontological classes. It is the base unit that can be described using the ontological model. Instances can represent concrete objects such as people or buildings and also abstract objects such as nationality or job.
- **Relations**: Relations describe the ways classes and individuals are related to each other. For example, the relation "measures_Acceleration" describes how the class "Accelerometer" is related to the concept class "Acceleration". For each relation, its domain and range classes are defined in the ontology. Subsumption which denotes the "subClassOf" relation is the relation by which the class hierarchy in an ontology is defined.

Ontologies also include additional components such as class attributes, restrictions, rules and events.

When concrete data is integrated into the basic model of the ontology, the resulting ontology is known as a knowledge base. In a knowledge base, there exists a logical distinction between the intentional knowledge or the general domain knowledge modeled in the ontology and between the extensional knowledge which is gained by adding concrete data instances to the core ontology model. Following this distinction, two types of statements in ontologies are defined:

- **Terminological Statements (TBox)**: statements that describe general concepts as well as properties and relations between these concepts.
- **Assertion Statements (ABox)**: statements that describe properties and relations of specific individuals and assert the class memberships of each instance.

The TBox/ABox separation can be thought of as an analogy to the separation between classes and objects within the frame of object oriented programming. In Chapter 6, this distinction is utilized in order to facilitate the semantification and processing of large amounts of sensor data using big data technologies and relying on the Semantic Sensor Network ontology introduced next in Section 3.3.1.

3.3.1 Semantic Sensor Network (SSN) Ontology

The SSN ontology [20] is an ontology designed by the *W3C semantic sensor network incubator group* to describe the capabilities and the properties of sensors, the act of sensing as well as the resulting observations from the sensing process. It was modeled using the broadest definitions of concepts omitting the description of concepts that are related, but not specific to sensors, such as locations and units of measurement. This enabled later expansion through the introduction of related- and sub-concept definitions for each individual application which led to the wide adoption of SSN in both research and industrial applications.

The SSN ontology consists of 41 concepts and 39 object properties which can be conceptually organized to ten modules as shown in Figure 3.1. Namely, it describes:

1. The sensors along with their accuracy and capabilities.
2. Sensor observations and the methods used for sensing.
3. Operating and survival ranges of sensors in addition to the sensor performance within these ranges.
4. The deployment process of sensors in field along with the deployment platform and lifetime.

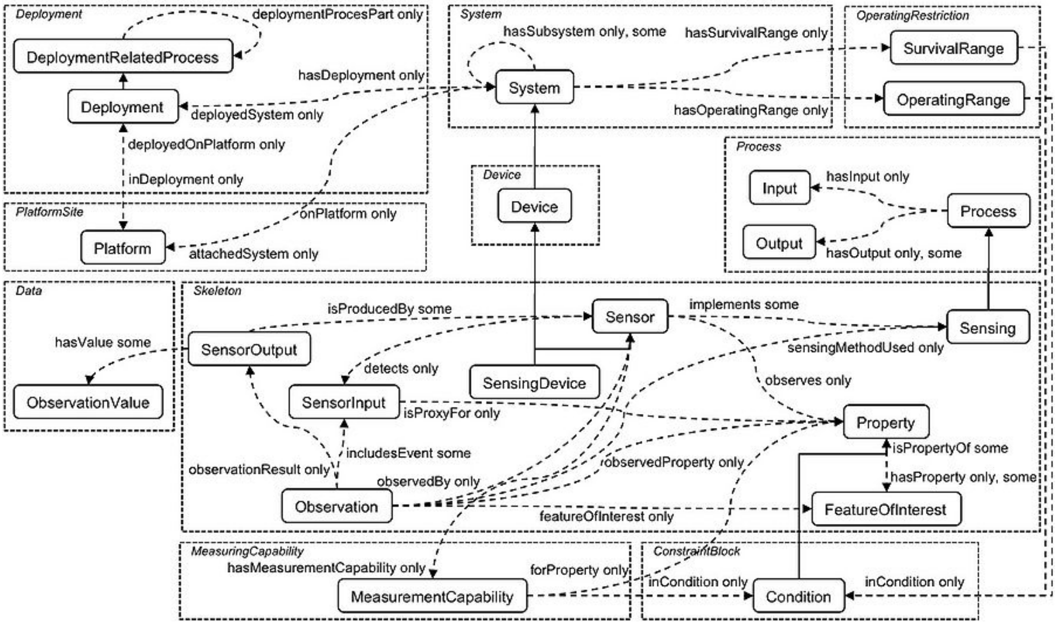

Figure 3.1: The SSN ontology conceptual modules, showing the main ontology classes and relations(as shown in [20]).

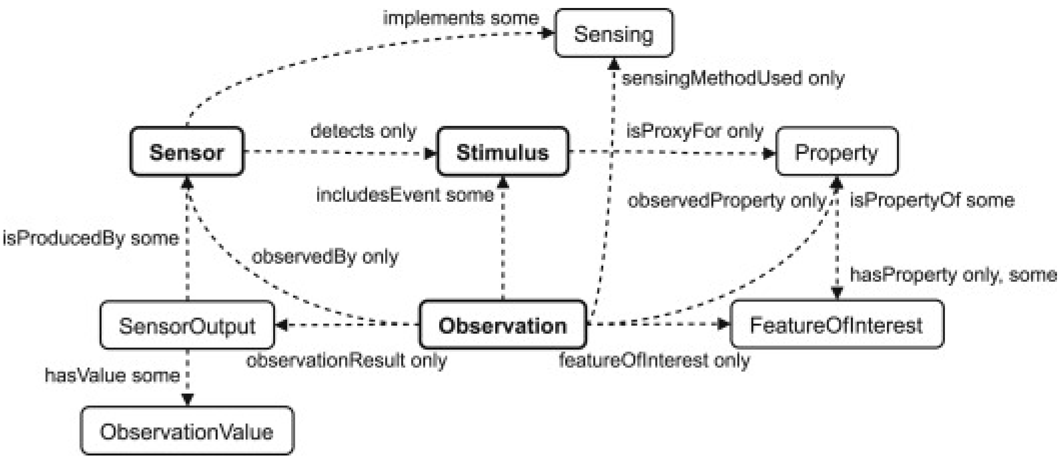

Figure 3.2: The Stimulus-Sensor-Observation Pattern (SSO) with core concepts in bold as shown in [20].

SSN, in its core, is based on the Stimulus-Sensor-Observation (SSO) pattern introduced in [46] and illustrated in Figure 3.2. This pattern aims at linking sensors to what they sense(stimuli) and to the resulting observation of the sensing process. Conceptually, the SSN ontology can be viewed from four main perspectives that represent the stimulus, sensor and observation perspectives following the central SSO pattern in addition to a system perspective that involves system organization and sensor deployments.

The SSN ontology represents the base ontology used in this work to achieve semantification of sensor data as introduced later in Chapter 6.

3.4 Introduction to Big Data

Coupled with the semantic technologies introduced in Section 3.3, this thesis also uses big data technologies in order to achieve optimized processing of semantic sensor data. Due to the fast growth of various data produced that exceeded the computing processing power, came the need for a new scalable solution that meets the high data demands. The term "big data", first introduced in [63], refers to various aspects of handling very large datasets for which the traditional data management systems such as relational database management systems are inadequate or inefficient. Big data technologies thus include means for data collection, data storage, data analysis, data visualization and querying among other challenges.

Generally, big data is often described by the following characteristics known as the 5 Vs of big data:

1. **Volume**: Refers to the quantity of the generated and stored data.
2. **Variety**: Refers to the nature of data. This includes structured data which was traditionally handled with DBMSs in addition to a wide variety of unstructured data types such as messages, social media posts, pictures, sensor data, audio and video data.
3. **Velocity**: Refers to the speed at which the data is generated and the speed at which it needs to be processed according to the usage scenario. This is very important especially in the case of sensor networks which collect sensor data constantly and need to be processed in real-time for certain applications.
4. **Value**: Refers to the ability of materializing big data by achieving a value that could not be achieved otherwise.
5. **Veracity**: Refers to the quality of the data. The quality and accuracy of big data collection is usually less controllable. For example, a malfunctioning sensor within a sensor network can produce erroneous data. However, in most cases, the quantity of the collected data balances out the negative effect of the lesser quality in big data collections.

3.4.1 MapReduce and Hadoop

The major advancements that established the basis for the big data technologies are the MapReduce processing architecture [28] and the Google File System [36] that were introduced by Google in 2004. MapReduce is a distributed processing model that allows parallel execution of queries on huge datasets stored on a large cluster of commodity machines. The MapReduce defines two processing steps:

- **Map**: in a "map" step, a mapping function is applied to all the data items locally in each machine.
- **Reduce**: in a "reduce" step, a certain accumulating function is applied progressively to the resulting items in order to produce the requested outcome of a query.

Apache Hadoop is an open-source framework that was built based on the MapReduce architecture. It includes two main modules, Hadoop MapReduce [10], which is the open source parallel processing system based on the original MapReduce, and the Hadoop Distributed File System (HDFS) [14], which is the underlying distributed file system that provides high data access throughput for applications.

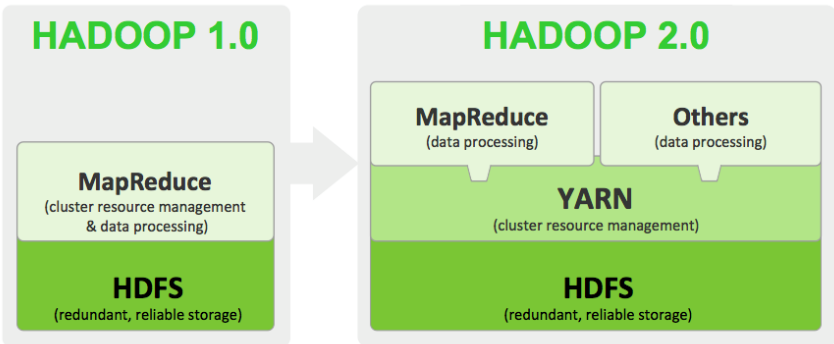

Figure 3.3: Hadoop 1.0 and Hadoop 2.0 architectures

.

Hadoop Yarn [85] is a resource negotiator (Yet Another Resource Negotiator) that was later introduced into the Hadoop 2.0 framework enabling native support of data processing systems other than Hadoop's MapReduce as shown in Figure 3.3. Examples of data processing systems running on top of Hadoop file system include Apache Flink [17] and Apache Spark [93] which both utilize in-memory processing of data to achieve significant performance gains compared to the original MapReduce engine. Apache Spark is utilized in Chapter 6 due to its advantages explained next in Section 3.4.2.

3.4.2 Apache Spark

Apache Spark is a general engine for large-scale data processing that utilizes in-memory computing to achieve up to 100x better performance than the state of the art MapReduce [28] approach.

Spark's efficient performance is a result of using a read-only data structure called Resilient Distributed Datasets (RDDs) [92]. RDDs can be stored in main memory which makes a significant difference especially with iterative algorithms where intermediate results do not need to be replicated and persisted on the distributed file system with each iteration. To maintain fault tolerance, Spark records all transformations and rebuilds any lost RDDs by reapplying the recorded transformations on the original datasets.

Spark supports two types of RDD operations:

- **Spark Transformations**: A transformation creates a new RDD from an existing RDD by applying functions such as:
  - **map**: applies a user-defined function to an RDD and produces a new RDD with the outcomes.
  - **filter**: returns an RDD of elements of the existing RDD that fulfill a certain condition.
  - **union**: returns a combined RDD containing the elements of two existing RDDs.
- **Spark Actions**: An action runs a computation on an RDD and returns a resulting value to the driver program. Example actions include:
  - **reduce**: aggregates all the elements of an RDD by using a user-defined function.
  - **count**: returns the number of elements of an RDD.
  - **collect**: returns all the elements of the RDD.

It is worth mentioning that Spark follows a ”lazy evaluation” approach which means it does not execute transformations until an action is requested which is analogous to the ”late materialization” concept used in relational databases. This allows Spark to optimize the processing of the whole set of operations increasing the overall performance of the system

In Table 3.1, major characteristics of Hadoop MapReduce, Spark and Fink are compared. Note that Apache Flink also includes the major Spark advantages, however Spark was chosen for implementations in this thesis as some Flink components were not available at the time of this work.

Table 3.1: Characteristics of the three different big data processing engines.

	Hadoop MapReduce	**Apache Spark**	**Apache Flink**
In-Memory Caching	Not Supported	Supported	Supported
Stream Processing	Only Batch	Near Real-Time	Real-Time
YARN Support	Yes	Yes	Yes
Languages	Java	Java, Python, Scala	Java, Scala
Machine Learning	using Apache Mahout	using MLlib	using Flink ML

3.5 Summary

In this chapter, a necessary technical background on semantic sensor data and big data technologies is presented. Research related to the semantification of raw sensor data using semantic sensor networks ontologies is discussed. Additionally, relevant methods for processing semantic data by harnessing big data are introduced. Notably, a gab is presented in related research when it comes to the specific case of processing semantic sensor big data. This highlights the need to bridge this technology gap by proposing a novel approach for processing semantic sensor data using big data techniques in Chapter 6. Research related, though application-wise, to the street quality assessment application presented in Section 7.3 is also presented.

This chapter also includes a basic overview of semantic data and ontologies, and a brief introduction to the Semantic Sensor Network ontology (SSN) which are vital for the realization of semantic sensor data. Finally, this chapter presents an overview of big data frameworks including Hadoop MapReduce, Apache Spark and Apache Flink. A qualitative comparison among the frameworks is presented and the key features of Apache Spark utilized in the approach proposed in Section 6.4 are highlighted.

Aspect-Based Sentiment Analysis for Sensing Devices 4

4.1 Introduction: Utilizing Consumer Reviews for Evaluating Market Sensor Devices

As indicated in Chapter 1, the rapid increase in releasing market sensor devices along with the wide range of user domains for which these devices are released, make evaluating and comparing these products a challenging task. Rival products are being released with different feature sets, continuous firmware updates and different design perspectives. All these factors make conventional product testing impractical as it is both difficult and time consuming to mimic all usage scenarios and to run the tests over an extended period of time.

In this chapter, we introduce a novel and dynamic approach for evaluating market sensor devices. The approach relies on the abundance of the online customer reviews for market sensor products. The choice of customer reviews as the main data source for the presented sentiment-based evaluation came after examining other different potential sources such as blog entries, technical articles, and social media posts.

For blog entries and technical articles, one must account for paid content that undermines the value of the evaluation in addition to the scattered nature of such data sources which makes it difficult to obtain sufficient amount of data that could serve the purpose of providing a comprehensive evaluation of the products.

As for social media content, examining thousands of posts and tweets related to consumer sensor devices, we found out that most of the posts do not address a specific feature of the device but rather the device as a whole. As the system aims at evaluating each product at the aspect level, social media was ruled out as a qualifying data source.

In the case of customer reviews, the continuous monitoring by online retailers for advertised content in the reviews add to their authenticity. Moreover, the relativity unrestricted length of the reviews allows customers to express their opinions on every feature of the product which is vital for an aspect-based evaluation of the product.

This chapter proceeds as follows. Section 4.2 introduces the main components of the sentiment summarizer and explains the flow of data between these components. Section 4.3 describes the Natural Language Processing techniques used and their significance for the system components. Sections 4.4 through 4.7 detail the processing of the customer reviews turning each review into a set of product, feature and sentiment tuples that convey the reviewers' experiences with the product. In Section 4.8, we discuss the significance and usability of using the sentiment summarizer to evaluate competing market sensor devices.

4.2 Sentiment Summarizer Overview and Data Flow

The sentiment analysis pipeline presented generates a user-friendly summary containing insights about each of the aspects of the evaluated sensor device or devices. The process starts with the raw customer reviews and passes through multiple different components as indicated in Figure 4.1. Before being passed to the core components of the system, reviews are extracted from their hosting trading websites and passed through an initial preprocessing step where the HTML code is parsed and noise is removed in order to obtain the review text only which is passed to further stages. As the example use case of evaluating leading market fitness trackers through this approach uses Amazon.com as a source for reviews(Section 7.2), an Amazon review extractor is implemented to obtain the required review objects given the name or the ID of a certain product.

After obtaining the review objects, the reviews are passed to the natural language processing (NLP) component, where the reviews are tagged with several NLP features required by the other components. The reviews are then passed to three components of the pipeline, namely the product name extractor, the aspect extractor and the sentiment extractor. The following step is assigning the extracted sentiments to their proper products and product aspects enabling the creation of a user-friendly presentation of the evaluation insights through the summarizer component.

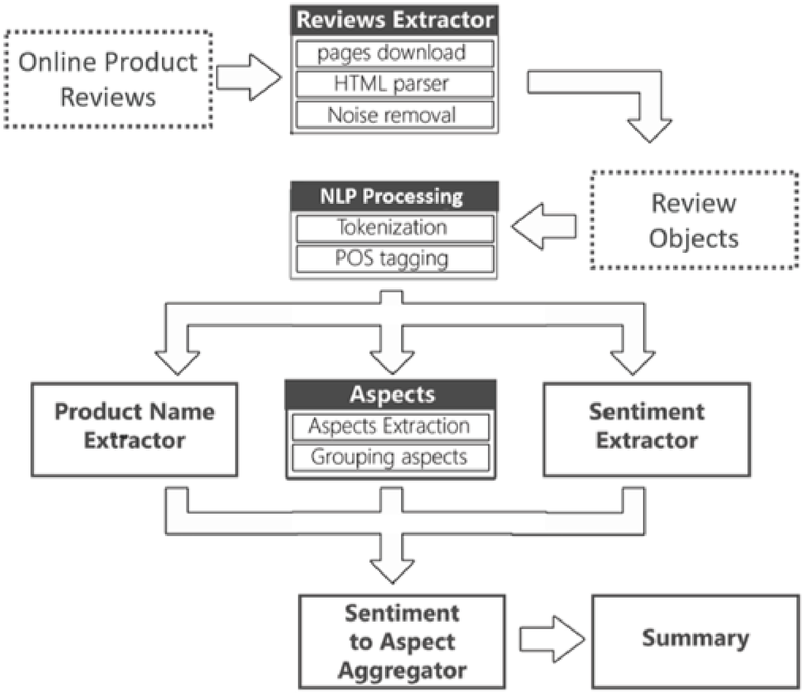

FIGURE 4.1: Sentiment summarizer overview

Below is a brief introduction of the aforementioned components which will be further discussed in details later in this chapter.

- **Product Name Extractor**

  The product name extractor aims at identifying the mentions of a given product or its competing products in the review text. For example, the sentence "I love my *Fitbit One*. I bought it after I lost my *Ultra* 2 weeks ago", the product name extractor infers that first sentence is discussing "Fitbit One", whereas the latter discusses "Fitbit Ultra". This component is essential to assign the sentiment to the proper product as we do not assign all sentiments included in a review directly to the reviewed product considering situations where reviewers use comparisons of the reviewed product to other competing products. Product name extraction is further discussed in Section 4.4.

- **Aspect Extractor**

  The aspect extractor aims at identifying products aspects which denote the features of the discussed by reviewers in the review texts. For commercial sensor devices, a distinction is made between two types of aspects. Static aspects include features shared among all products such as the "price" and the "customer service quality". While dynamic aspects denote features of a sensor devices that may differ among products such as "measuring humidity" and "measuring speed" features in a connected weather sensor and a fitness tracker respectively. The sentence "It accurately **measured** my **pulse**" for example illustrates that the reviewed device has a "pulse measuring" aspect. Aspect extraction is further discussed in Section 4.5.

- **Sentiment Extractor**

  The sentiment extractor aims at inferring the polarity of a sentence or a phrase. The polarity can be positive as in "This sensor is very reliable", negative as in "useless device" and neutral as in "The temperature measured is 20 degrees" which is a non-sentiment-bearing objective sentence. Section 4.6 introduces the sentiment extraction component.

- **Sentiment to Aspect Aggregator**

  This component selects the aspect of the product for which an extracted sentiment is to be assigned. The outcome of this component is a set of (product, aspect, sentiment) triplets that summarize all the reviewer's opinions expressed in the review text.

- **Summarizer**

  The summarizer component generates the final output of the whole pipeline. It presents a web-based summary of all the insights learned from all the reviews of a single sensor device or a group of competing devices. This enables users to easily grasp the pros and cons of each product in order to make an educated decision in acquiring market sensor devices. The summarizer interface is discussed in Section 4.8.

4.3 Natural Language Processing Components

Natural Language Processing (NLP) is a field of research that combines computer science and linguistics methodologies in order to achieve a better understanding, analysis and generation of natural languages (e.g., German, English or Japanese).

Among the typical application domains for NLP are machine translation, information extraction, information retrieval and sentiment analysis, which is where NLP is utilized in this work. Tokenization, named-entity recognition, part-of-speech tagging, parsing, describing the grammatical relationships in sentences and lemmatization are NLP tasks required for the sentiment analysis system presented. Below is a brief introduction to these NLP components.

- **Tokenization**

The process of tokenization can be described by the breaking of a character sequence into pieces(tokens), while stripping it from certain characters such as punctuation. The problem of tokenization is highly language-dependent, however typically in Latin languages, the token roughly takes the form of one word. The tokenizer used in this work is introduced in [83].

- **Part-Of-Speech(POS) tagging**

  In traditional grammar, words(or lexical items in general) which have similar grammatical properties are categorized under a single part of speech(POS). Those words play similar roles within the structure of a sentence. In the English language, commonly listed parts of speech are *noun, verb, adjective, adverb, pronoun, preposition, conjunction, interjection,* and sometimes *numeral, article or determiner*. The part-of-speech tagging approach used in this work is described in [83].

- **Lemmatization**

  Lemmatization is the process of getting the base or dictionary form of a word, which is known as the lemma. Often lemmatization is mistaken with stemming which is a heuristic process that eliminates words endings including affixes. Lemmatization, on the other hand, uses vocabulary and morphological analysis of words in order to return the lemma. For example, stemming the words ”are” and ”saw” would keep them the words unchanged or result in an incorrect trimming of their endings while the lemmatization of these words results in ”be” and ”see” respectively, which are the base forms of these words.

- **Named Entity Recognition (NER)**

  Named Entity Recognition (NER) is a subtask of information extraction that first identifies sequences of words in a text which are the names of things. The identified word sequences are then classified(labeled) into predefined categories such as the names of persons, organizations, locations and states to name a few. The named entity recognizer used here is introduced in [35].

- **Parsing**

  Natural language parsing is the task of recognizing the grammatical structure of sentences. The parser is a vital component of natural language processing. For example, it can recognize the word, or group of words, that form the subject or the object of an identified verb which is very important for understanding the sentence as a whole. The natural language parser used here is introduced in [79].

- **Grammatical relationships**

To obtain a representation of the grammatical relations between words in a sentence, Stanford typed dependency representation [27] is utilized. A typed dependency labels dependencies with grammatical relations, such as subject or indirect object. An example for identifying grammatical relationships between words is marking the relation between ”not” and ”watch” in ”Fitbit is not a watch" as negation.

In this work, the Stanford CoreNLP toolkit [62] is used to provide the natural language processing abilities needed by the system. This toolkit is an extensible pipeline that provides a set of core human language technology tools and is widely used in both the research community and by commercial users as well.

4.4 Product Name Extraction and Significance

A customer review mainly discusses a single product. This implies that most of the features discussed and the sentiment baring phrases included in the review are related to that product. However, reviewers often use comparisons to other competing products as in the following cases: to compare the performance of a feature that is included in both, to highlight a feature in the reviewed product that is missing in other market products, to criticize a feature lacking in the reviewed product where it is available in another, or simply to compare the reviewer’s personal experiences with both products. Below are some examples of comparative sentences in which the sentiment expressed does not describe the reviewed product.

- The Fitbit is much more accurate in steps counting than the Jawbone Up.
- I especially like the Nike Fuel performance measure that is not available in neither the Fitbit One nor the Jawbone Up.
- It would be nice to have a silent alarm such as the brilliant one in the Jawbone.
- I have owned a Fitbit Flex and a Nike Fuelband before, both were completely uncomfortable, unlike my new Jawbone Up.

Given our goal of getting the most accurate assessment of sensing devices necessitates detecting the product names mentioned within a review so that each sentiment is then assigned to the right product. For this purpose, two major approaches for extracting product names are proposed. The first approach relies solely on the review text and uses natural language processing to extract the product names from the reviews. The other approach utilizes the reviews meta-data available from Amazon.com, which is our primary source for the collected product reviews. In both cases, detecting the mentions of product names is done in two steps.

1. In the first step, a list of competing products that can potentially be compared to the reviewed product is assembled.
2. In the next step, each review is checked for the presence of mentions of competing products.

It is worth mentioning that people do not always refer to a product with its official name such as using "The Up" to refer to "Jawbone Up" or "Fitbit" to refer to "Fitbit One" and not "Fitbit Flex" for example.

4.4.1 Using NLP to Extract Product Names

Starting with a corpus of customer reviews for a certain product, the review texts in all products are preprocessed in order to compile a list of potential competing products which is then used to match mentions of product names in each review to the right product from this list.

By manually analyzing naming conventions for products in the sensing devices domain, different naming patterns are learned. For example, some product names consist of the manufacturer's name followed by the model name or a number. Using part-of-speech tagging, word groups following the learned patterns are collected. For example, proper nouns that are followed by an uppercase word such as ”Nike Fuelband” are detected. The mentions of word groups following the different product naming patterns are then sorted by their frequency in the product reviews corpus and only the most frequently mentioned word groups are considered as potential competing product names.

It is worth mentioning that even though some manufacturers produce different sensing devices however people tend to call their most common product by the manufacturer's name alone. This is seen for example with the ”Nest Thermostat” which is often referred to as ”Nest” alone although there are several other products produced by the same manufacturer such as ”Nest Cam” or ”Nest Protect”. For this, any mentions of a manufacturer's name alone is matched to the most common product of this manufacturer.

This approach depends solely on the reviews text in order to determine the mentioned product names. This however can be enhanced by utilizing product meta-data available through market outlets such as Amazon.com as shown next in Section 4.4.2.

4.4.2 Extracting Product Names Using Product Meta-data

In this section, we use the product meta-data included in Amazon.com to determine a list of competing products. Is is noteworthy here to indicate that determining a list of competing products using the help of Amazon meta-data does not mean that the system only works for Amazon reviews. An example of product listings on Amazon is shown in Figure 4.2.

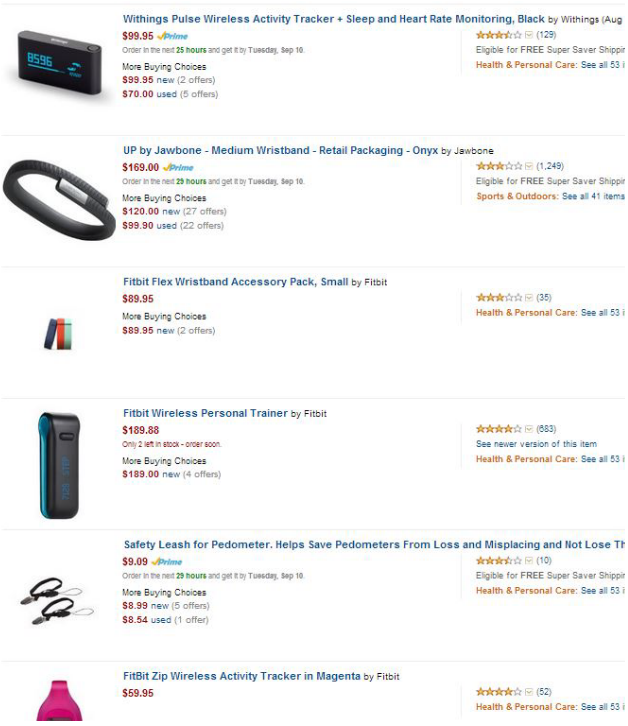

Figure 4.2: Example listings of sensing products on Amazon

For every product listing, the listing includes a title and a manufacturer name. A set of regular expressions are made to match product naming conventions. Two examples of such patterns in the listings of Figure 4.2 are indicated below:

- ([manufacturer name] [model name]) such as ”Fitbit Zip” from ”Fitbit Zip Wireless Activity Tracker in Magenta - by Fitbit”
- ([model name] ”by” [manufacturer name]) such as ”Up by Jawbone” from ”Up by Jawbone - Medium Wristband - by Jawbone”

The process of determining the product names from the listings is illustrated in Figure 4.3.

FIGURE 4.3: Product name extraction from an Amazon product listing

Given the ability to determine the product name through processing a product listing, what remains is determining a set of listings that contains competing products. To obtain this set, we use two different sources of data. The first is the set of similar and recommended products provided for each listing. The other is by utilizing the product hierarchy through Amazon Browse Nodes. Competing products have multiple common parents in the product hierarchy and hence combining the set of products with common parents with the set of recommended products results in a highly precise detection of product names as seen in Section 5.3.

After compiling a list of competing products, each mention of a product name from this list in all the reviews is detected. However, writers often refer to a product by a portion of its name only. For that, a set of additional patterns is generated for each product to match possible references to the product by the reviewer such as plural form and the usage of the manufacturer name or model name separately. It is important here to indicate that this process of detection product names in the review text aims at increasing the accuracy of sentiment assignment, however, without any indication of the product name, it is assumed that the sentiment is assigned to the reviewed product.

4.5 Device Aspects Extraction and Clustering

The main motivation behind evaluating sensing tracking devices is assessing the reliability of these devices, which are essentially sensor data generating devices, thus enabling more people to make educated decisions in acquiring the right product in the evolving consumer sensing devices market. In the sentiment-based evaluation approach followed, providing the user with insights on the aspect level is more useful for the device assessment. For that, aspects of each device are identified following the approaches presented in Section 4.5.1and multiple words or phrases referring to the same aspect are clustered as shown in Section 4.5.2.

4.5.1 Aspect Extraction

Device aspects can be categorized under static and dynamic aspects. Static aspects refer to the device aspects that are common among all devices such as the price and the customer service whereas dynamic aspects are aspects specific to each device and those are the type of aspects we aim at extracting in this section.

As for product name extraction, different aspect extraction approaches are presented from which some rely solely on the review text while others benefit from the available meta-data assigned to the reviewed products through Amazon.com in this case. In what follows we present in Section 4.5.1.1 a mesh up between two such approaches using review text of the product in addition to the product description text provided. An alternative more domain-specific approach using domain-related word filter is also presented in Section 4.5.1.2.

4.5.1.1 Aspect Extraction Through Product Description and Reviews

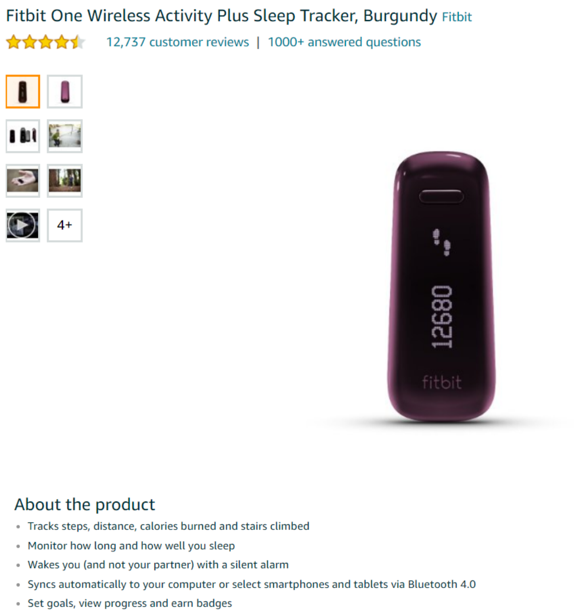

Figure 4.4: A sample listing of product aspects in the product description data

Figure 4.4 shows a sample listing of product aspects in the product description data. In this case, where the aspects are listed in a well-structured manner, extracting nouns and noun groups such as ”steps tracking” and ”sleep monitoring” is considered a sufficient and easy approach to learn the dynamic aspects of a sensing device without the need to analyze review text. This, however, is not the case with all product listings where often the product description data is unstructured, inconclusive and even non-existent sometimes.

For that, another approach which relies on the review text is needed. Since all the products we are trying to evaluate are sensing devices, which basically means that those devices' main dynamic aspects involve measuring certain data, aspect mentions are expected to co-occur frequently with certain verbs such as "monitors" and "tracks". An extensive set of similar verbs describing the action of "sensing" is manually prepared and the nouns which frequently occur within a close proximity to the "sensing" verbs are considered to refer to dynamic aspects of the product. Few examples of the co-occurrences are given below:

- After a long walk, it **registered** only 600 **steps**!
- I love how it **tracks** my **sleep**.
- It accurately **measured** my **pulse**.

To obtain the dynamic aspects following this approach the following steps are followed:

1. The system feeds the aspect extraction component with part-of-speech tagged reviews.
2. Each co-occurrence of a noun or a noun group with a "sensing" verb is marked. For example, "skin temperature" and "calories" co-occur with "sensing" verbs such as "measures" and "registers".
3. After aggregating the co-occurrences for each marked noun or noun group, it is considered a dynamic aspect only if its co-occurrences with a "sensing" verb exceed an experimentally chosen threshold of five co-occurrences.

Although this approach succeeds in extracting many correct aspects of a product, it sometimes mistakenly identifies false nouns or noun groups as aspects. For that, a combined approach is used that utilizes both the product description as well as the product reviews for a refined product aspects extraction. This combined approach identifies an aspect only if it exists in both of the following two sets of candidate aspects:

- **Set A**: Set A contains nouns and noun groups found in product description data of the reviewed product itself and in the descriptions of its top competing products. A sample product description for the Fitbit

Zip sensing device is shown in Figure 4.5. This set basically forms a relaxed set of aspects that are common among the domain to which the reviewed product belongs. Which means that it forms a superset to which all the aspects of the reviewed product belong. By using the descriptions of the top competing products of the domain, the problem of non-comprehensive descriptions for some products is solved as this approach relies on multiple descriptions rather than just a single one.

- **Set B**: Set B product reviews in the same manner indicated above where the nouns and noun groups frequently co-occurring with ”sensing” verbs are identified. The ”sensing” verb and the noun are said to be co-occurring if the distance between the two is within the experimentally chosen range of ten words. In addition, this set includes the nouns in certain syntactic patterns such as ([noun or compound noun] [verb] [adjective], e.g., The steps are accurate.) and ([The device comes with] [noun]).

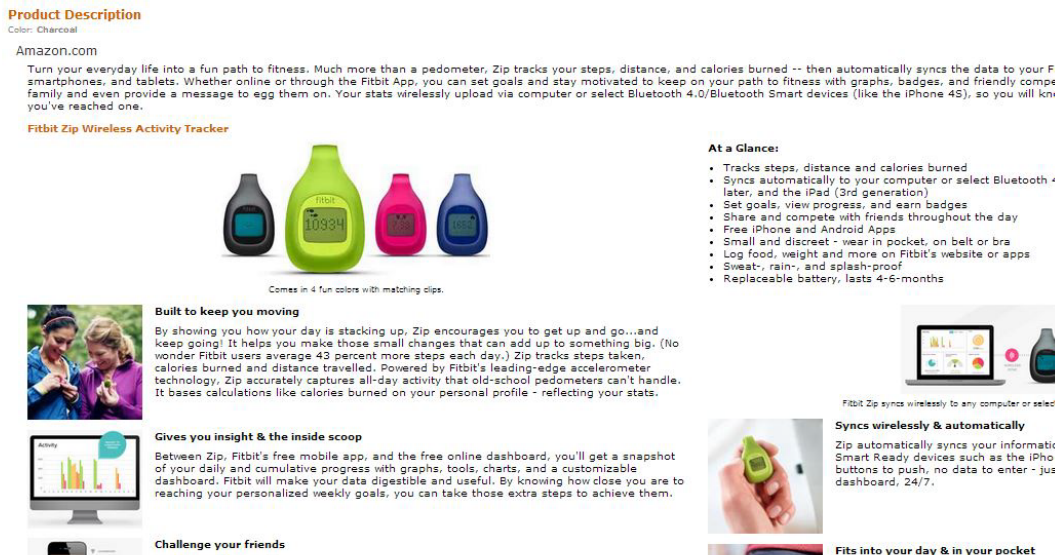

FIGURE 4.5: Product description of Fitbit Zip by Fitbit

As indicated, set A candidates denote a relaxed set of aspects from the whole domain to which the reviewed product belongs where as candidates in set B denote a noisy set of the product aspects that includes in addition to the product aspects, very common nouns that can be in close proximity to a ”sensing” verb such as ”phone” or ”computer”. Combining the two sets, the candidates present in both are assigned as the dynamic aspects of the reviewed product.

4.5.1.2 Domain-Related Word Filter

As previously introduced in Section 2.3, several approaches consider the aspects to be very common nouns. However, to obtain a set of these aspects, the most common nouns such as stop words are filtered out. By manual inspection of the lists of most common nouns in the reviews of three market leading fitness trackers, it is noticed that after the removal of stop words, many fitness-related words such as ”fitness” and ”health”, in addition to competing product names such as ”Fitbit” and ”Jawbone” were among the most common nouns. Details of the outcome are included in Section 5.4.

Since this approach relies on domain-knowledge in order to correctly remove the most common nouns and noun groups that are not considered aspects, a domain-related word filter is built for a specific branch of sensing devices, namely fitness trackers. This word filter was built through two steps:

1. Extracting the most common nouns and noun groups from the corpus of reviews for three market leading fitness trackers. This resulted in a list of 1577 words.
2. The resulting list was manually inspected and the domain-related words were removed in this step reducing the list to 1554 words. It is worth mentioning that although this step involved manual inspection of the list of common nouns, this process is not very time consuming as the list was manually checked in less than two hours in this case.

After obtaining the list of non-domain-related common nouns and noun groups in the reviews, this list is fed into the system where words having same lemmas such as ”steps” and ”step” are combined using the lemmatization approach introduced in Section 4.3. The most common English stop words are also filtered out in this process. Competing product names also commonly appear among the most common nouns, for this, the outcome of the product name extraction component introduced in Section 4.4 is used in order to filter out the product names from the list. Any remaining noun or noun group, appearing in the reviews of a product beyond an experimentally chosen threshold of three mentions is considered to be an aspect of this product.

It is worth mentioning, that even though this approach involves manual inspection of domain-related nouns, however, the ease of this process which has to be done only once for an application domain enables it from being applied for all products in this specific domain.

4.5.2 Clustering of Product Aspects

Grouping words referring to the same aspect of a product is a vital task for the sentiment analysis system. Because even when words are correctly identified as aspects, people tend to use different words or expressions to refer to a single aspect. Dealing with each such word as a different aspect heavily undermines the outcome of the system as the results which are referring to the same aspect are not properly aggregated. The reference to the same aspect could be achieved through the usage of different parts of speech and word choices as shown in the example below for the aspect ”price”:

- Fitbit is relatively **cheap**.
- I got it for a hundred **dollars**.
- Too much **paid** for a little value!
- The Fuelband is very **pricey**.
- It may be **expensive**, but still, it is robust.

As a primary step, frequent noun groups that are sometimes referred to as a concatenation of multiple words and sometimes as separate words are grouped together, for example, ”heart rate”, ”heart-rate” and ”heartrate” are all used forms of the same noun group. Then, the lexical knowledge in the English language provided by WordNet [66] is utilized in order to group all the terms related to a single aspect under one cluster. This enables the addition of all the mentioned forms of an aspect and many other similar terms representing synonyms, co-ordinate terms, derivations, hyponyms, and units. To illustrate, consider the following example where a portion of the cluster describing the ”price” aspect is presented:

- Synonyms: cost, toll, monetary value, worth, reward
- Co-ordinate terms:
  - Cost: expense, payment
  - Value: invaluableness, richness, priceless
- Derivations:
  - Price: pricey, prices, pricing, priced
  - Value: valuable, valueless, valued, values
- Hyponyms: money (hyponym of ”value”)
- Signs: buck, dollar (signs for ”money”)

The correct extraction and the correct clustering of aspects are essential to the sentiment summarizer pipeline. In Section 4.6, the sentiment bearing mentions within customer reviews are extracted. Each extracted sentiment is assigned to the cluster representing the aspect of the product no matter what words were used by the reviewer to refer to this aspect. The process of assigning a sentiment to an aspect is described in Section 4.7.

4.6 Lexicon-Based Sentiment Extraction

As introduced earlier in Section 2.4, sentiment analysis is a very challenging research problem. And despite the importance of all the other components in the proposed sentiment summarizer pipeline, correctly identifying the sentiment remains the most critical task. As the system is targeting a wide class of products, the choice of a lexicon-based unsupervised approach helps giving the system the flexibility and the usability needed to be applied to different categories of sensor products. This choice, however, outperformed state-of-the-art supervised approaches as shown later in Section 5.5, thus, no performance compromise is made.

As the granularity of sentiment analysis in this work is chosen to be on the aspect level, each sentiment-baring phrase within a review is assigned its own sentiment score. The score is basically an aggregation of the partial scores assigned to each token in the phrase, which in turn are assigned to each token based on the sentiment score of this token in the combined lexicon introduced in Section 4.6.1 and altered to compensate for several sentiment shifters as shown later in Section 4.6.2 through Section 4.6.5.

4.6.1 The Sentiment Lexicon

A sentiment lexicon is a lexical resource that assigns a polarity value for every word. For example, it assigns a positive polarity score for words such as ”win”, ”happy” and ”successful”, and a negative polarity score for words such as ”lose”, ”sad” and ”failing”. Several sentiment lexicons are compiled through different approaches and are widely used in both research and applications. Some examples are the MPQA Subjectivity Lexicon [87] and the Harvard General Inquirer [80] in addition to Bing Liu’s Opinion Lexicon [42] and SentiWordNet [34] [4] which are used in this work and introduced below:

1. **Bing Liu’s Opinion Lexicon**

   This lexicon comprises a manually compiled collection of sentiment-bearing words. This collection consists of 2006 positive words and 4783 negative words (a total of 6789 words). Since this collection is manually annotated, it is considered highly accurate. Moreover, it includes common misspellings, morphological variants, slang, and social-media mark-up which is very useful for our system as it has to handle informal text written by reviewers.

2. **SentiWordNet**

   SentiWordNet is built based on the English lexical database WordNet previously introduced in Section 2.3.3. It assigns a probabilistic sentiment score to all the 117,659 synsets of WordNet which include a total of 38,182 non-neutral words. It is worth mentioning that since WordNet is aware of the multiple parts-of-speech each word can assume, SentiWordNet assigns different sentiment scores for the same word in different positions. Since SentiWordNet automatically annotates all the synsets of WordNet, it is considered one of the most comprehensive sentiment lexicons. Nevertheless, due to the lack of human verification, the polarity is not always properly assigned by this lexicon.

In this work, those two lexicons are utilized in a way that the advantages from both of them are materialized. Namely, the precision and the understanding of misspellings, morphological variants, slang, and social-media mark-up from Liu's lexicon are coupled with the size and the part-of-speech and different meanings awareness of SentiWordNet.

Each POS-tagged token is assigned a sentiment score based on the procedure explained in Algorithm 1 and Algorithm 2. Note that Algorithm 2 handles the case where the word is tagged positive in Liu's lexicon and an equivalent approach is followed for words that are tagged negative but it is omitted to avoid repetition. This procedure results in different assignments based on the relative sentiment assignments by each of the two lexicons. Below is a list of all the possible situations. Since SentiWordNet is POS-aware, only the scores assigned to the different meanings of a word for its tagged POS are considered. The scores assigned by SentiWordNet are always normalized to result in a score between -3 and +3.

- **Same Polarities:**
  If a word is tagged positive in Liu's lexicon, and it is also tagged as positive in SentiWordNet, the outcome is a positive sentiment and the score is equal to the score of the most positive meaning of this word in SentiWordNet. Conversely, words considered negative in both lexicons are assigned a negative sentiment with a score equal to the score of the most negative meaning of this word in SentiWordNet.

- **Different Polarities:**
  When for the tagged POS, SentiWordNet outcome contradicts Liu's lexicon outcome, the outcomes for the same word but in different parts of speech in SentiWordNet are inspected and the algorithm proceeds in one of the following two manners:

- If for another POS, the polarity of the word matches the polarity assigned by Liu's lexicon, then it is considered that the word is not being used in its most common POS annotated in Liu's lexicon, and therefore the polarity and the sentiment score from SentiWordNet are assigned to the word. To illustrate, consider the word "like" in "this product is like the others". The most common POS for "like" is the verb, so understandably Liu's lexicon assigns positive sentiment for the word "like". But considering the fact that in this case "like" is a preposition and not a verb, SentiWordNet would assign to it a neutral polarity. By confirming that SentiWordNet has indeed a positive sentiment assigned to the verb "like", this indicates that SentiWordNet does not actually contradict Liu's lexicon, but instead it assigns the correct sentiment for the word in its true POS whereas Liu's lexicon assigns the sentiment of its most commonly used form.

- If for all POS, the polarities assigned by SentiWordNet contradict Liu's lexicon, then it is considered that since Liu's lexicon is more accurate, this contradiction is due to a wrong assignment by SentiWordNet which might occur due to the automatically annotated nature of SentiWordNet. The word is therefore assigned a sentiment of +1 if positive and -1 if negative.

• **Word Only Existing in SentiWordNet:**
In the case where a word is not found in Liu's lexicon, the word is assigned the average sentiment score for all its meanings within its tagged part of speech in SentiWordNet.

• **Word Missing in Both Lexicons:**
The word is considered neutral in this case. SentiWordNet covers a vast number of words and Liu's lexicon lists the most common informal mentions of words. But due to several situations that might happen within an informal text like customer reviews, the usage of foreign languages, internet slang, and misspellings can result in a word not being found in both lexicons.

Algorithm 1 $computeSentimentScore(word, POS)$

```
sentimentScore ← 0

if word ∈ LiuPositive then
  sentimentScore ← computeSentiWordNetScorePositive(word, POS)

else if word ∈ LiuNegative then
  sentimentScore ← computeSentiWordNetScoreNegative(word, POS)

else
  sentimentScore ← average(SentiWordNetSynsets(word, POS))

end if
return sentimentScore
```

Algorithm 2 $computeSentiWordNetScorePositive(word, POS)$

```
sentimentScore ← −3

for synset ∈ SentiWordNetSynsets(word, POS) do
  synsetScore ← getPositiveScore(synset) − getNegativeScore(synset)
  if synsetScore > sentimentScore then
    sentimentScore ← synsetScore
  end if
end for

if sentimentScore ≤ 0 then
  matchingPOS ← true
  for synset ∈ SentiWordNetSynsets(word, POS) do
    synset ∈ SentiWordNetSynsets(word)
    if getPOS(synset) ≠ POS then
      if getPositiveScore(synset) − getNegativeScore(synset) > 1 then
        matchingPOS ← false
      end if
    end if
  end for
  if matchingPOS = true then
    sentimentScore ← 1
  end if
end if

return sentimentScore
```

Table 4.1: Sentiment scores assigns to the tokens of the sentence.

Token	I	do	not	see	any	problem	with	it
Sentiment Score	0	0	0	0	0	-1	0	0

4.6.2 Negation Component

Handling negations is one of the most significant challenges among the sentiment analysis challenges introduced in Section 2.4. This is mainly due to the difficulty in determining the negation scope and whether or not the sentiment assigned to a certain aspect should be shifted due to the presence of the negation word. The most common negation words in the English language are: *no, not, none, no one, nobody, nothing, neither, nowhere, never*. Below are some examples of the usage of negation in sentences:

1. **No** correct values are being registered by the sensor.
2. This device has **none** of the features I like.
3. This design is **neither** pretty nor functional.
4. I do **not** see any problem with it.
5. The fall did **not** succeed in breaking my robust tracker.

For most negation words, using a window approach, in which the polarities of all the opinion words lying within a close proximity to the negation word are shifted, is sufficient. In this work, the windows size of 5 is experimentally chosen. However, handling the negation words ”never” and particularly the word ”not”, is more challenging. ”not” is the most commonly used negation word in the English language, it is used for negating different parts of speech and additionally can be paired with modal verbs such as ”should”, ”could” and ”might”.

The negation component in the system handles negations by assigning sentiment scores to the tokens comprising the sentence. Then identifies the negation scope which includes the tokens that are affected by the negation.

- **Assigning sentiment scores to tokens**

  The sentiment is assigned to each token in the sentence using the approach presented in Section 4.6.1. To illustrate, Table 4.1 shows the sentiment scores for all the tokens in the sentence ”I do not see any problem with it.”.

- **Negation scope detection**

  To determine the negation scope and thus determine the tokens that must have their polarities shifted due to the usage of the negation words, this following approach is used.

  1. Through utilizing the NLP grammatical relationships component introduced in Section 4.3, the word directly dependent on the negation word is determined using the negation typed dependency relationship. For the example sentence *"I do not see any problem with it."*, the negation relation is between the words *"not"* and *"see"*.
  2. Using the parse tree of the sentence, the part of speech of the negated word is determined. Table 4.2 introduces the tags of all the parts-of-speech used in the parse trees of this section. The parse tree of the sentence *"I do not see any problem with it."* is shown in Figure 4.6 where the word *"see"* which is the negated word, is tagged as a verb(VB). Similarly, for the sentence *"It is not an accurate tracker."*, the parse tree is shown in Figure 4.7, and the negated word is *"tracker"* which is tagged as a noun(NN). Also for the sentence *"It looks fresh and not ordinary."*, the parse tree is shown in Figure 4.8 and the negated word *"ordinary"* is tagged as an adjective(JJ).
  3. Depending on the part of speech of the negated word, the closest phrase of same type of the negated word is marked. That is, the closest verb phrase if the negated word is a verb, the closest noun phrase if the negated word is a noun and the closest adjective phrase if the negated word is an adjective. In our examples, the verb phrase *"see any problem with it"*, the noun phrase *"an accurate tracker"* and the adjective phrase *"fresh, not ordinary"* are marked in the sentences of Figure 4.6, Figure 4.7 and Figure 4.8 respectively.
  4. The negation scope is determined to include all the tokens belonging marked phrase if the negation word is not included in that phrase such as the case with negation word *"not"* which does not belong to the marked noun phrase *"see any problem with it"*. However, for marked phrases that include the negation word such as the adjective phrase "fresh, not ordinary", only the sentiment of tokens following the negation word is inverted.

```
(ROOT
    (S
        (NP
            (PRP I)
        )
        (VP
            (VBP do)
            (RB not)
            (VP
                (VB see)
                (NP
                    (DT any)
                    (NN problem)
                )
                (PP
                    (IN with)
                    (NP
                        (PRP it)
                    )
                )
            )
        )
    )
)
```

Figure 4.6: The parse tree of "I do not see any problem with it".

```
(ROOT
    (S
        (NP
            (PRP It)
        )
        (VP
            (VBZ is)
            (RB not)
            (NP
                (DT an)
                (ADJP
                    (JJ accurate)
                )
                (NN tracker)
            )
        )
    )
)
```

Figure 4.7: The parse tree of "It is not an accurate tracker".

4.6.3 Sentiment Intensifiers and Sentiment Diminishers

Sentiment intensifiers and diminishers were introduced previously in Section 2.4. Keeping the same polarity, the sentiment score of any positive word following an intensifier such as "very good" is increased by $+1$. Similarly, the sentiment score of any negative word following a sentiment intensifier is increased by -1 as in the phrase "very bad". In the case of sentiment diminishers such as

```
(ROOT
    (S
        (NP
            (PRP It)
        )
        (VP
            (VBZ looks)
            (ADJP
                (JJ fresh)
                (, ,)
                (RB not)
                (JJ ordinary)
            )
        )
    )
)
```

FIGURE 4.8: The parse tree of "It looks fresh, not ordinary."

"barely" and "hardly", the polarity of the following sentiment-bearing word is shifted. For example, the positive polarity of "functional" in the sentence "it is hardly functional" is shifted to reflect the effect of the sentiment diminisher word "hardly".

4.6.4 Verbs Disabling Sentiment

A list of verbs that disable the sentiment of words following them is compiled. This list includes the following verbs and verb phrases: "wish", "suggest", "though", "wonder", "would like" and "would be". These verbs and verb phrases are set to disable the sentiment of words within 3-word distance from them. For example, the positive sentiment carried by the word "accurate" in *"I wish it was accurate"* is disabled because it was preceded by the verb "wish".

4.6.5 Comparative Sentences

Comparative sentences are sentences that compare an entity to another. They can be either gradable or non-gradable [49]. Gradable comparisons use adjectives or adverbs that can vary in intensity (e.g., greater, less than, better). An example of gradable comparisons is *"Fitbit One looks better than the Jawbone Up"*. Such comparative and superlative (e.g., best) forms help in identifying the comparative sentences. Non-gradable comparisons compare two or more entities implicitly, such as in *"Fitbit One can count floors climbed, but Jawbone up cannot"*.

To identify comparatives, the system uses the product name extractor component to identify any mention of competing product names. This works for both

Table 4.2: Description of the POS parse tags

Tag	Description
ADJP	Adjective phrase
CC	Coordinating conjunction
DT	Determiner
IN	Preposition or subordinating conjunction
JJ	Adjective
NN	Noun,singular or mass
NP	Noun phrase
PP	Prepositional phrase
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
S	Clause (main or dependent)
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
VP	Verb phrase

gradable and non-gradable comparisons since any comparison of a product to another product is expected to contain the name of the competing product as well. Determining the preferred entity is fairly simple in sentences that contain adjectives in comparative or superlative forms. The polarity of the base of the comparative adverb or adjective is checked against the sentiment lexicon to determine the preferred side of the comparison. To illustrate, consider the phrase ”x is better than y”. Since the base form of ”better” which is the word ”good” is positive then the preferred entity is the one at the left side of the comparative relation and vice versa. A similar argument is used for sentences containing ”more”, ”most”, ”less” and ”least”.

4.7 Sentiment to Aspect Matching

After extracting product names and aspect mentions in the review text as shown in Section 4.4 and Section 4.5 respectively, and after computing a proper sentiment score for every sentiment bearing word in the text, handling various challenges as shown in Section 4.6, follows the need to match each sentiment bearing word to the proper aspect it semantically describes. The matching follows the cases shown below:

1. Sentences containing a single aspect: The overall sentiment of the sen-

tence is assigned to this aspect. An example of a such sentence is ”Its *temperature* recordings are very *accurate*.”.

2. Sentences containing multiple aspects: Each aspect is assigned the overall sentiment of the deepest parse tree that contains only that aspect. An example of such sentence is ”Its *temperature* recordings are very *accurate*, but its *pressure* recordings are always *faulty*.”.

3. Sentences containing no aspects:

   - If a product name, or another word referring to it such as ”device”, ”product” or ”sensor” is mentioned. Then the overall sentiment of the sentence is assigned to the product as a whole. An example sentence is ”Jawbone Up is an awesome tracker!”.
   - If these are no references to a product as a whole, then the sentiment of the sentence is assigned to the last-mentioned aspect. An example of such case is the sentences ”The device also measures the *heart-rate*. The values recorded are very *accurate*.”.

After assigning all the sentiment-bearing words to the proper aspects. The overall sentiment for the aspect within the sentence is then computed using the score function in Formula 4.1. This score function, introduced in [29] aims at degrading the weight of the sentiment bearing words assigned to an aspect along with the distance separating the opinion word from the aspect mention. $sentiment\_score(a)$ denotes the overall sentiment assigned to the aspect in a sentence or a phrase, S denotes the set of all opinion words within the sentence, $sentiment\_score(s)$ denotes the sentiment score attached by the sentiment extraction component to the word s.

$$aspect\_score(a) = \sum_{s \in S} \frac{sentiment\_score(s)}{d(a, s)} \tag{4.1}$$

4.8 Sentiment Summarizer Outcome Presentation

The collective experience of multiple users is expressed through their reviews of the products they have owned and used. This collective experience enables our sentiment analysis system to identify the various opinions given on each feature in a market sensor product. This however is not enough. Presenting the results of the evaluation in an easy-to-grasp manner is vital for conveying insights to the average consumer.

Figure 4.9: The controversial aspects perspective of the sentiment summarizer.

In this section, a web-based sentiment summarizer is introduced as the final outcome of the sentiment analysis pipeline presented. This summarizer features a dynamic, interactive design that enables the user to compare several competing sensor devices by navigating around the evaluation results using different perspectives and with different levels of granularity. The different perspectives of the summarizer are introduced below. This section includes text and figures from my publication [44].

4.8.1 The Controversial Aspects Perspective

Figure 4.9 shows an example view for this perspective. This perspective allows the user to know what are the features that were most commented on by the reviewers relying just on the frequency in which an aspect is mentioned in the reviews. This is important to know for example whether a distinguished feature of a product was well regarded by the customers or not.

In this view, the user selects one device and is presented with a chart showing the aspects of this device ordered by their frequency of mentions. This perspective also displays all the aspect mentioning phrases grouped by their polarity. Upon selecting one aspect from the chart, the phrases displayed are filtered to show only the phrases discussing the selected aspect.

FIGURE 4.10: The device perspective of the sentiment summarizer.

4.8.2 The Device Perspective

The device perspective displays the distribution of positive, negative and neutral mentions per aspect of the device. An example view for this perspective is shown in Figure 4.10. The user can limit which polarity classes to view omitting the neutral class for example as is the case in Figure 4.10. The ability to directly inspect the phrases responsible for the polarity distribution for a certain aspect allows the user to learn quickly about the pros and cons of this aspect by only reading the relevant phrases.

4.8.3 The Aspect Perspective

Unlike the former two perspectives, the aspect perspective is centered around the aspects rather than devices. This perspective allows users to compare different products limiting the comparison to a certain aspect. An example of the aspect perspective is shown in Figure 4.11.

In this perspective, the user is presented a list of all the aspects extracted from all the compared devices. After selecting an aspect, the user can compare the evaluations of the different devices with respect to this aspect only. Similar to the device perspective, the user can limit the polarity classes displayed. This perspective allows users to inspect the evaluations of the most relevant features with respect to their preferences.

Figure 4.11: The aspect perspective of the sentiment summarizer.

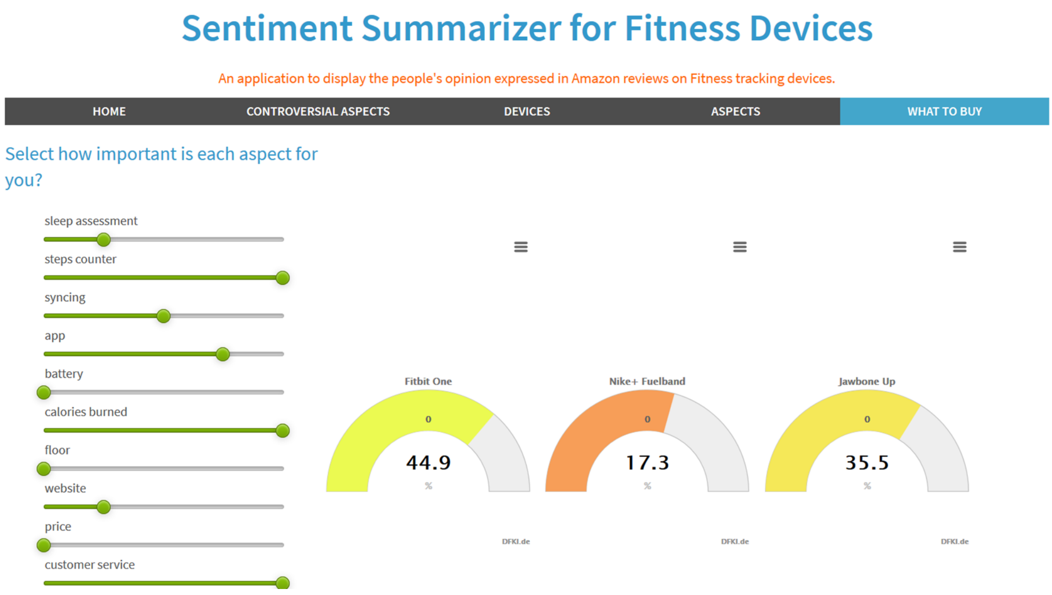

Figure 4.12: The device recommender perspective of the sentiment summarizer.

4.8.4 The Device Recommender Perspective

Given the fine-grained knowledge on the aspects of market fitness devices that can be acquired by using the sentiment analysis system introduced, the final perspective in the summarizer utilizes this knowledge to help each potential customer choose the device that suits their needs best based on their personal preferences. Figure 4.12 shows the user interface for the device recommender perspective in the sentiment summarizer. A user can select using 5-value sliders, how important an aspect is to them, and the recommender then dynamically upgrades score gauges for each device thereby helping each customer making the right purchase decision based on all the acquired knowledge of owners of these devices.

$$Score_d = \frac{100}{k} \sum_{a_{i=1}}^{k} \frac{q_{a_i}}{4} \times \begin{cases} \frac{pos(a_i)-neg(a_i)}{pos(a_i)+neg(a_i)} & if \quad a_i \in A_d \\ -1 \quad otherwise \end{cases} \tag{4.2}$$

The scores are computed using Formula 4.2; where d denotes a device, k denotes the total number of aspects for which the user has assigned a non-zero weight using the slider, a_i denotes a single aspect ($i \in [1, k]$), $q_{a_i} \in [0, 4]$ denotes the weight assigned by the user to aspect a_i, $pos(a_i)$ and $neg(a_i)$ denote the number of positive (res. negative) mentions of the aspect a_i for device d, and A_d denotes the set of all aspects for device d.

Intuitively, when the user gives a non-zero weight for a certain aspect using the slider, this aspect is added to the set of user-requested aspects. For each aspect a_i of these k aspects, the system checks if it is included in the aspects of the considered device. If yes, a value $\frac{pos(a_i)-neg(a_i)}{pos(a_i)+neg(a_i)} \in (-1, 1)$ is returned indicating how recommended (res. unrecommended) the device is when considering this aspect alone. Otherwise, if the aspect requested by user is not available for the device, -1 is returned indicating that the device is not recommended at all when considering this aspect. The result is adjusted for each aspect based on the user-selected weight q_{a_i}. The final score $Score_d$ is then computed as the normalized mean for the returned weighted values for each aspect.

4.9 Summary

In this chapter, a sentiment analysis pipeline for market sensor devices' reviews is introduced in order to evaluate all the features of a consumer sensor device. This approach is driven by the need for a dynamic method to evaluate the various sensor devices which are constantly released into the consumer market. For that, customer reviews, in which people who have experienced a certain product express their opinion about its features, are considered as the main source for the evaluation process.

The different components of this pipeline are introduced throughout this chapter. The process starts with extracting review objects for one or more sensor devices. These reviews are then fed into three core components of the system namely the product name extractor, the aspect extractor and the sentiment extractor.

The product name extractor aims at identifying the product name mentions in the review text so that the sentiment is assigned to the correct product in the cases where the reviewer mentions a product different from that being reviewed as is the case in comparative sentence. Two approaches are introduced, the first relies solely on the review text while the other utilizes available meta-data for the reviewed device.

The aspect extractor helps achieve the goal of identifying the features addressed by the reviewers in the review text. This enables the system to assign sentiment expressed in a phrase to a specific feature of a product rather than a product as a whole. Two approaches are introduced, the first uses both product description and product reviews text in order to identify the aspects while the other uses a domain-related word filter over most common nouns and noun groups to identify the aspects. The system then uses NLP techniques and utilizes lexical knowledge in order to form a single cluster from words referring to a single aspect.

The sentiment extractor identifies sentiment bearing phrases in a review text. A lexicon-based approach which relies on a combination of a highly accurate human-made sentiment lexicon and a comprehensive probabilistic lexicon is used. Several sentiment analysis challenges such as negation handling are addressed in order to overcome sentiment shifting in sentiment-bearing phrases. The sentiment is then assigned to its proper aspect and product.

The main motivation for evaluating market sensor devices is the assessment of their features and the reliability of their measurements so that potential customers can make an educated decision acquiring a reliable sensor product that fits their needs. For that, a user-friendly web-based sentiment summarizer is introduced in which a user can grasp the collective opinion of product reviewers, compare different competing products and get a personal recommendation for a product based on his/her specific needs.

The different components of the sentiment analysis pipeline are next evaluated in Chapter 5 and an example application where three market-leading consumer sensor devices, from the fitness trackers category, are evaluated in Section 7.2.

Evaluation of the Sentiment Summarizer Pipeline 5

5.1 Introduction

The reliability of market sensing devices is very critical to the increase in their adoption by the average consumer and to the real-life scenarios and goals for which they can be utilized. To overcome the difficulties of sensing devices' quality assessment, an end-to-end sentiment summarizer pipeline was introduced in Chapter 4 that uses the crowd knowledge, acquired through customer reviews, to provide a comprehensive assessment for all the aspects of a sensing device. In this chapter, the components comprising this pipeline are evaluated.

For the evaluation, a set of customer reviews of the three market-leading fitness sensing devices introduced in Section 2.5 is used. The evaluation also uses a manually annotated dataset of these reviews as a ground truth dataset. More information on the evaluation datasets is presented in Section 5.2. The metrics used for the evaluation are introduced in Section 5.5.1. The sentiment summarizer pipeline main components, namely, the device name extractor, the aspects extractor and the sentiment extractor are evaluated in Section 5.3, Section 5.4 and Section 5.5 respectively.

5.2 Evaluation Datasets

5.2.1 Consumer Reviews Dataset

This dataset consists of Amazon customer reviews of the three market-leading fitness tracking devices, namely Fitbit One, Jawbone Up and Nike+ Fuelband.

The review objects were extracted by parsing the HTML code of Amazon pages as the official Amazon API does not offer customer reviews. Along with the review text, the title, date, product ID, author ID and the star rating were also extracted as review meta-data.

This dataset consists of 3,241 reviews containing 19,797 sentences and a total of 10,638 aspect mentions. More information on the distribution of the dataset reviews among the three products is shown in table 5.1.

Table 5.1: Amazon customer reviews dataset of top fitness tracking devices.

	Reviews	Sentences	Aspect Mentions	Sentiment-bearing Aspect Mentions
Fitbit One	2,504	15,154	7,682	6,372
Jawbone Up	696	4,397	2,138	1,645
Nike+ Fuelband	41	246	818	612
Total	3,241	19,797	10,638	8,629

As noticed in Table 5.1, Fitbit One and Jawbone Up number of reviews are significantly greater than that of Nike+ Fuelband. This does not necessarily reflect the difference in market share proportions among the three devices since Nike,as a more established sports company, relies more on its own sales channels rather than Amazon sales.

5.2.2 Ground-Truth Dataset

As a ground truth, a set of 105 reviews were manually annotated by a group of 10 computer science students. The reviews annotated were randomly selected according to the following two properties:

1. Even distribution among the three devices (each with 35 distinct reviews)
2. Even distribution among all the five star ratings (21 reviews for each star rating; 7 per device)

The annotations were made through the web-based form shown in Figure 5.1. The participants were asked to annotate each sentence separately. The complete review was also displayed so that the participant could see the sentence within the context of the review it belongs to. In each annotation, the participant selected the product name, product aspect as well as the sentiment in the form of a 7-point Likert scale ranging from -3 (denoting the most negative sentiment) to +3 (denoting the most positive sentiment).

For product name selection, the participants could select a product from an extensive list of relevant product names including Fitbit One, Jawbone Up,

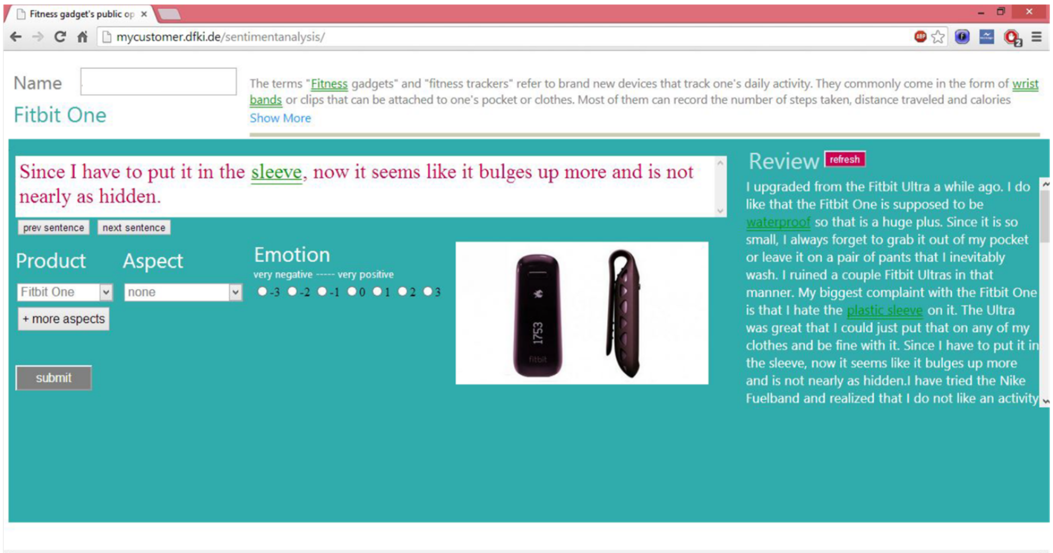

Figure 5.1: A snapshot of the web-based manual sentiment annotator.

Table 5.2: Sentiment distribution of the manually annotated reviews.

Description	Number of Sentences
Positive Sentiment	301
Negative Sentiment	280
Mixed Sentiment	34
Neutral	123
Total	738

Nike Fuelband, Fitbit Flex and Withings Pulse among others. In cases where there is no reference for a specific product, the participants were asked to select "None" as a product name.

As for aspect selection, the following set of aspects were available to choose from: steps counter, distance traveled, sleep assessment, floors climbed, calories burnt, battery, customer service, syncing, price, activity indicator and application functionality. The participants could also select "General" if the sentiment was referring to the device as whole rather than to a specific aspect of the device.

Since our system is aspect-based and not sentence based, multiple annotations were permissible for as single sentence so that sentences with mixed sentiments are properly annotated. An example of a such sentence is "Fitbit's battery life is quite good unlike the Fuelband which barely lasts a single day!".

More Information on the distribution of the manually annotated reviews is shown in Table 5.2.

5.3 Device Name Extraction Evaluation

The process of extracting product names from reviews aims at refining the overall performance of the system as in some cases the names of products, that are different from the reviewed product, are mentioned by the reviewer for the several reasons mentioned in Section 4.4. This process is done by first compiling a list of competing product names for the reviewed product, and then by utilizing this list in extracting the product names mentioned in each review. Using the approach presented in Section 4.4.2, the system produces a list of 26 competing products for the three sensing devices considered in the evaluation.

Figure 5.2: Evaluation of the list of competing product names.

By manually inspecting the list of competing product names, the distribution of the results is shown in Figure 5.2. For five products, the model name was not extracted by the system and hence the accuracy of this list is equal to 81%. It is important to note that not all of these product names are necessarily mentioned in the review dataset because this list is an extensive list of competing product names complied using the product meta-data available using the approach presented in Section 4.4.2. Out of all the 105 reviews manually annotated, only in one sentence the mention of a product name was not correctly identified by the system. This sentence is ”But the new one had the exact same issues”. The participants in the manual annotation correctly identified the word ”one” as referring to ”Fitbit One” which was not identified by the system.

5.4 Device Aspects Extraction Evaluation

Aspect extraction has been approached in several sentiment analysis systems through collecting the most frequent nouns in the reviews, pruning stop words and retaining only the list of words occurring with a frequency greater than that of an experimentally determined threshold [42] [72]. Applying this approach to the fitness tracker reviews dataset after pruning stop words, converting all words to a lower-case and retaining the 40 most frequent nouns resulted in the resulted in the nouns shown in Table 5.3 to be considered as the aspects of the three products reviewed.

The results, although correctly identifying several aspects of the devices, is very noisy and lack the correct grouping of same aspects. For example, the words "customer", *"support"* and *"service"* are parts of the noun groups *"customer service"* and *"customer support"* and hence should be identified as one aspect. In addition, product names such as *"jawbone"* or *"nike"*, which occur frequently in the reviews are identified as aspects too. Notably, for *"Fitbit One"*, the core *"distance traveled"* aspect did not occur among the 40 most frequent nouns in the list due to the noise in this list and this would clearly affect the usability of the sentiment analysis system, as it will be lacking any sentiments assigned to some core aspects.

Applying the aspect extraction approach proposed in Section 4.5.1, where a domain-related word filter is applied to the most frequent nouns and noun groups, results in pruning out frequent nouns that do not represent any aspect such as *"exercise"* and *"tracker"*. Also, this approach utilizes the product name extractor to identify and prune out all product names which occur frequently as clearly shown in Table 5.3.

The aspect clustering approach presented in Section 4.5.2 utilizes the lexical knowledge provided by WordNet in order extend the words belonging to the same aspect groups.

Table 5.4 shows the extracted aspects following this approach. The nouns and noun groups referring to the same aspect are grouped together and the list of extracted aspects occurring with a frequency greater than an experimentally chosen threshold frequency of 4% of the most frequent aspect is retained.

By manually examining the aspect groups extracted from the reviews of the three products, we could confirm that all the core aspects of these products were included in the results. It is also noticed that for the first two products, namely "Fitbit One" and "Jawbone Up", the extraction is highly precise due to the abundance of the customer reviews available for these products. As for "Nike+ Fuelband", it is noticed that there exists more noise especially in the least frequent results. This is due to the lower number of reviews for this product. This indicates that pruning the results beyond a higher frequency

Table 5.3: List of the 40 most frequent nouns in the reviews of the three devices.

Product	Nouns (sorted by frequency)
Fitbit One	fitbit=3921, day=1403, steps=1377, sleep=1173, device=1079, activity=756,time=726, calories=645, weight=630, stairs=573, product=555, clip=528,sync=522, food=474, website=458, app=443, thing=416, data=412,tracker=412, computer=394, days=373, ultra=368, track=353, week=349,times=336, friends=334, iphone=330, pocket=330, goals=316, night=312,customer=297,,tness=294, tracking=289, phone=279, weeks=278,goal=268, pedometer=264, exercise=261, service=256, bit=253
Jawbone Up	band=667, jawbone=541, sleep=515, app=335, product=330, time=320,day=309, days=246, steps=242, device=211,,tbit=195, data=192,sync=187, food=180, activity=158, customer=125, months=123,tracking=123, alarm=121, wrist=118, replacement=118, charge=118,battery=117, weeks=112, week=111, phone=110, bracelet=109, iphone=105,support=105, night=104, problem=97, service=96, thing=94, feature=91,times=84, bluetooth=83, month=80, amazon=80, information=79, life=79
Nike+ Fuelband	nike=392, band=282, fuel=232, fuelband=227, day=214, product=171,calories=103, days=99, time=98, device=95, points=89, steps=89,activity=78, goal=73, app=71, thing=65, wrist=63, watch=61,people=60, week=53, iphone=52, computer=52,,tbit=43, website=43,months=42, issues=42, sync=41, battery=40, problem=40, support=40,work=39, weeks=37, data=36, software=35, pedometer=33, customer=33,movement=33, service=32, mine=32, idea=31

Table 5.4: Aspect groups as extracted by the system

Product	Nouns (sorted by frequency)
Fitbit One	(step, step count), calorie, (stair, floor), (app, application, iphone app),(website, fitbit website, web site), (sleep, sleep tracker, sleep patterns, sleep tracking, sleep mode), friend, pocket, (customer service,customer support), badge, weight loss, activity level, point, fitness pal,distance, battery life, flower, food intake
Jawbone Up	(app, iphone app, up app), (sleep, sleep tracking, sleep mode, sleep tracker, sleep monitor, sleep patterns), step, (alarm, alarm clock), (customer service, jawbone support, jawbone customer, customer support), friend, calorie, up band, battery life, website, (headphone jack, jack), point, food intake, food tracking, replacement band, (power nap, nap), heart rate, expectation, distance, pocket
Nike+ Fuelband	calorie, step, (app, android app, iphone app), (fuel, nike fuel, fuel points, fuelpoint, point), (website, web site, nike website), friend, (customer service, customer support), clasp, screw, heart rate, nike store, rust, battery life, streak, couch, gps, water proof, stair, activity level, ice, nike support, algorithm, distance, lifting, measure, led lights

threshold and solve this problem but, to main consistency, we report here the results using the same threshold used for the other products.

5.5 Sentiment Extraction Evaluation

5.5.1 Sentiment Evaluation Metrics

5.5.1.1 Precision, Recall and Accuracy

To measure the performance of the different components in correctly identifying product names, aspects and sentiment the precision recall and accuracy are computed. These metrics are generally widely used in the fields of pattern recognition, information retrieval and binary classifications. Each of these three metrics and its significance is introduced below. For reference, Table 5.5 describes the confusion table for the results of a binary classification problem.

- **Precision**

  Precision, for a certain class, denotes the ratio of items correctly classified as belonging to this class to all the items predicted as members of this

Table 5.5: Confusion table for binary classification problems.

		Prediction	
		Positive	**Negative**
Actual	**Positive**	True Positives (tp)	False Negatives (fn)
	Negative	False Positives (fp)	True Negatives (tn)

class. This is basically the ratio of true positives to the sum of true positives and false positives as shown in Formula 5.1. For example, if the sentiment extraction component extracts 10 positive sentiment-bearing mentions, 7 of which are actually correctly classified while the remaining 3 are not, then the precision of the sentiment extractor for the positive class is said to be 70%. It is worth mentioning that precision alone is not a sufficient measure of the performance of a classifier as in the case where the system correctly classifies a single item out of a hundred items resulting in a precision of 100% despite missing most of the results. "Recall" presented next accounts for such cases.

$$Precision = \frac{tp}{tp + fp} \tag{5.1}$$

- **Recall**

  Recall, for a certain class, denotes the ratio of items correctly classified to the total number of items that actually belong to this class. This is basically the ratio of true positives to the sum of true positives and false negatives as shown in Formula 5.2. For example, if the sentiment extraction component extracts 7 positive sentiment-bearing mentions out of an actual 10 positive mentions available, then the recall of the positive class is said to be 70%. Solely relying on recall as a performance measure is also not sufficient because in the case when the system classifies all items as positives, there will be no false negatives and thus the recall will be 100% despite having a very low precision. This is why precision and recall are usually combined to measure the performance of binary classification systems.

$$Recall = \frac{tp}{tp + fn} \tag{5.2}$$

- **Accuracy**

  The accuracy is simply defined as the ratio of correctly classified items to the total sum of items. That is ratio of the sum of true positives and true negatives to all the items as shown in Figure 5.3. Accuracy serves as an overall measure of performance of binary classification systems however it does not result in the same level of insights provided by the combined usage of precision and recall metrics.

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{5.3}$$

5.5.1.2 K-Fold Cross-Validation

In Section 5.5.3, the performance of the proposed lexicon-based sentiment extractor component is compared to the performance of the two state-of-the-art supervised classifiers introduced in Section 2.4.3, namely Naive Bayes and support vector machine (SVM) classifiers.

For supervised learning algorithms, an annotated (labeled) dataset is divided into a training set and a test set. The training set is used in the learning step to obtain the classification model while the test set is used to evaluate the performance of the model using a performance measure such as precision and recall. As the labeled data is limited, splitting the data between the training set and the test set compromises either achieving a better model or a more accurate validation as shown below.

- A larger training set leads to a better model since the algorithm has more data available for the learning step.

- A larger testing set leads to better validation accuracy as the performance is measured over a larger number of labeled items.

Since the goal is having both a larger training set and a larger test set, the data is split into k equal partitions and the following steps are repeated k times(folds) where in each fold a new partition is selected as a test set:

1. One partition is selected as a test set and the other $k-1$ partitions are used as a training set.
2. The training algorithm computes a model based on the selected training set consisting of the remaining $k-1$ partitions.
3. The obtained model is validated using the selected test set.

The final performance results is then computed as the average of the k obtained results. Intuitively, the k-fold cross-validation technique uses all the available labeled data as both training and test data which leads to a better evaluation of the model.

5.5.2 Baseline Setup

For evaluating the sentiment extraction component introduced in Section 4.6 two baseline supervised learning approaches are implemented, namely Naive Bayes and Support Vector Machines, both of which are previously introduced in Section 2.4.3. In the evaluation, the following points highlight considerations taken in the setup of baseline classifiers:

1. Aspect-based sentiment classifiers that use supervised learning often assume that a sentence contains only a single aspect. For this reason, sentences only complying to this assumption are included in the evaluation dataset despite the fact that our proposed system can handle several sentiment bearing aspect mentions in a single sentence.
2. Following the work in [69] which suggests that training both SVMs and Naive Bayes classifiers on unigrams yield better results in sentiment classification of customer reviews, both baseline classifiers were trained on unigrams from the dataset labeled by human participants introduced earlier in Section 5.2.2. Stop words were filtered out in a preprocessing step.
3. The baseline approaches where implemented using the data science software platform RapidMiner [56]. The implemented svm classifier is based on mySVM [74] implementation where multiple parameter settings

were experimented and the results reported are the best obtained by using a polynomial kernel function.

4. K-fold cross validation is used as introduced in Section 5.5.1.2 in order to achieve better classification models and verification. The experiments use either 3-fold and 10-fold cross validation.

5. In all experiments, recall, precision and accuracy were calculated to evaluate the performance of each classifier.

5.5.3 Experiments and Results

In this section, two main experiments are considered. In the first experiment, our proposed system and the two baseline approaches classify the sentences in the dataset into three classes, positive, negative and neutral. In the second experiment, the baseline approaches are evaluated in a binary sentiment classification setup omitting all neutral sentences from the dataset to check whether they perform better as usually suggested in the literature [55].

5.5.3.1 Three-Class Evaluation

It is common in sentiment classification research to ignore the neutral class. [55] however, highlights the importance of training classifiers on positive, negative and neutral data items so that the classifier learns to better distinguish the positive and the negative items in real life scenarios where the three classes exist. In this experiment, 10-fold cross validation results for the supervised baseline approaches are reported. The one-vs-all approach is used to enable running SVM on the multi-class setup.

Table 5.6: Three-class precision evaluation of the proposed lexicon-based classifier and the two baseline approaches.

Classifier	**Positive Precision**	**Negative Precision**	**Neutral Precision**	**Accuracy**
Naive Bayes	73.23	46.36	23.71	37.50%
SVM	58.70	44.25	22.89	41.43%
Lexicon-Based	70.15	78.84	33.96	67.28%

Table 5.6 and Table 5.7 list the precision, recall and accuracy results for all the three approaches. Additionally, Figure 5.3 and Figure 5.4 compare the three classifiers' results in both positive and negative classes respectively.

For the positive class, our system achieves a very high recall of 89.86% while maintaining a fairly high precision of 70.15% which is slightly less than the 73.23% precision for Naive Bayes, however Naive Bayes achieves a much less recall as evident from Figure 5.3.

Table 5.7: Three-class recall evaluation of the proposed lexicon-based classifier and the two baseline approaches.

Classifier	**Positive Recall**	**Negative Recall**	**Neutral Recall**	**Accuracy**
Naive Bayes	30.90	25.00	82.11	37.50%
SVM	44.85	35.59	46.34	41.43%
Lexicon-Based	89.86	58.36	30.25	67.28%

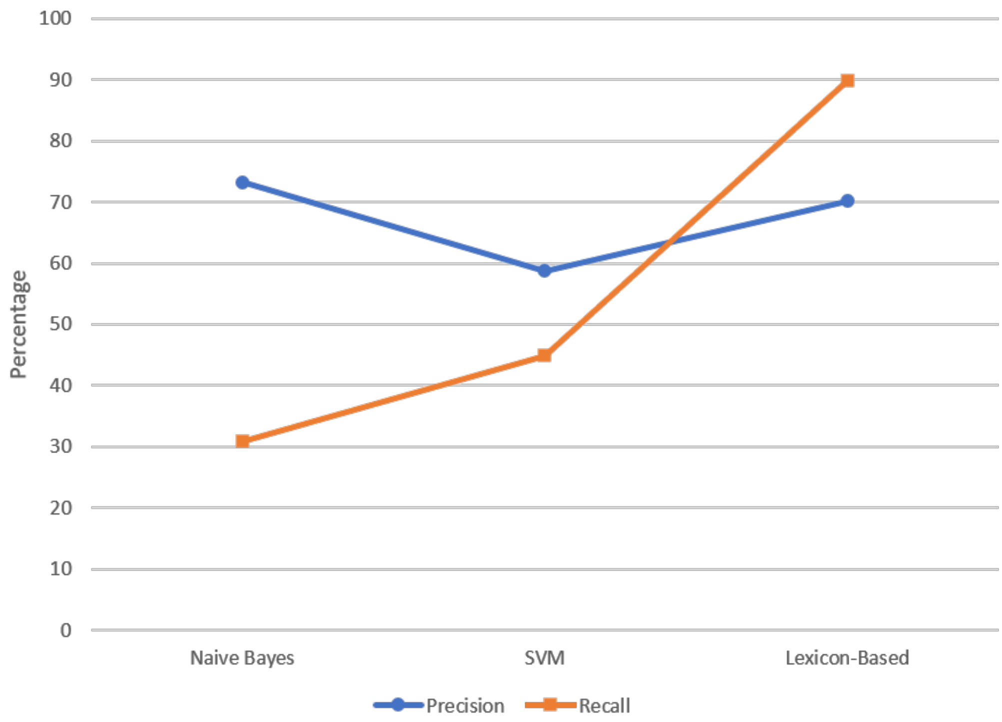

Figure 5.3: Positive class precision and recall in the multi-class classification experiment.

For the negative class, our system outperforms both baseline approaches maintaining a relatively high precision of 78.84% but with a less recall value of 58.36%. Upon further inspection of the results, it appears that several sentences labeled negative in the ground truth were labeled as neutral by our system. This indicates that the system has a better distinction between positive and negative classes than between negative and neutral classes.

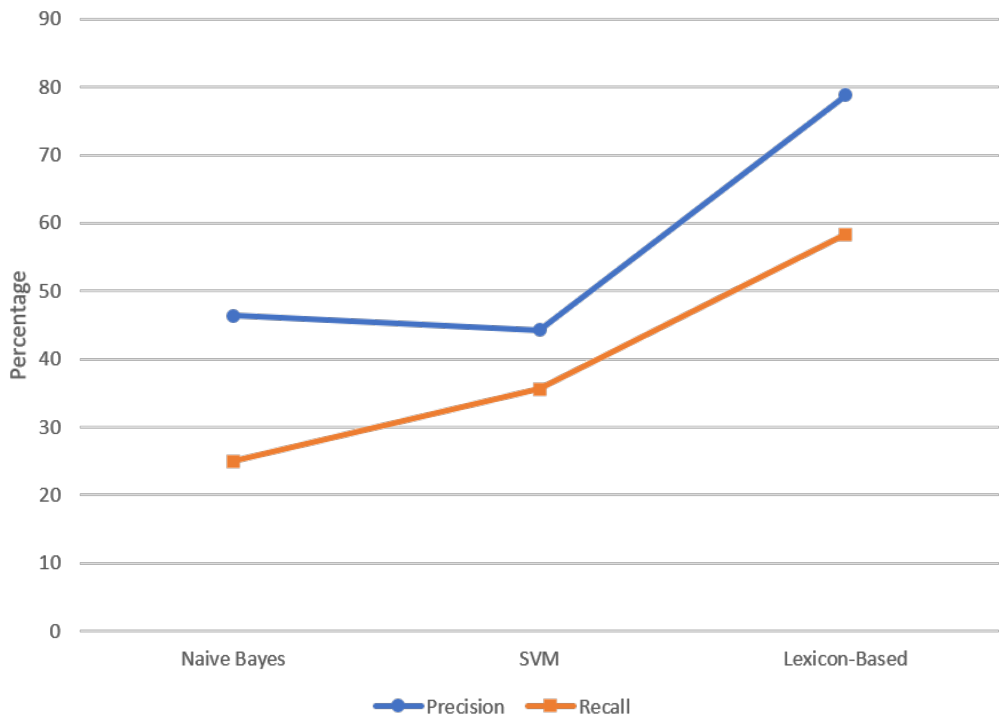

FIGURE 5.4: Negative class precision and recall in the multi-class classification experiment.

Table 5.8: Evaluation of the two baseline approaches trained on two classes.

Classifier	Folds	Positive Recall	Negative Recall	Positive Precision	Negative Precision	Accuracy
NB	3	46.18	80.36	71.65	58.14	62.65%
SVM	3	41.53	76.16	65.10	54.87	58.25%
NB	10	46.18	84.64	76.37	59.40	64.72%
SVM	10	60.80	70.82	69.06	62.78	65.64%

5.5.3.2 Two-Class Evaluation

In order to assess the performance of the two baseline approaches when trained on sentences with clear polarity, neutral sentences were removed from the labeled dataset and the classifiers were trained on strictly positive and strictly negative sentences. The outcome of this experiment shows a significant improvement over the three-class experiment that includes neutral sentences as shown in Table 5.8. The accuracy of SVM reached 65.64% while that of Naive Bayes reached 64.72% which are still less than the accuracy by our lexicon based classifier in the more realistic three-class setup.

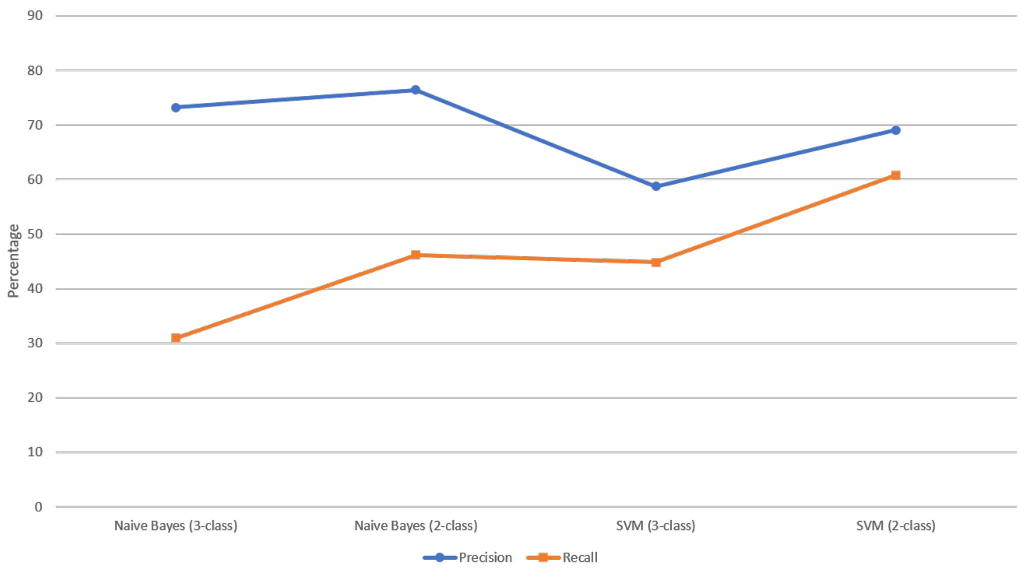

Figure 5.5: Comparing the positive class precision and recall in the two-class and three-class classifications.

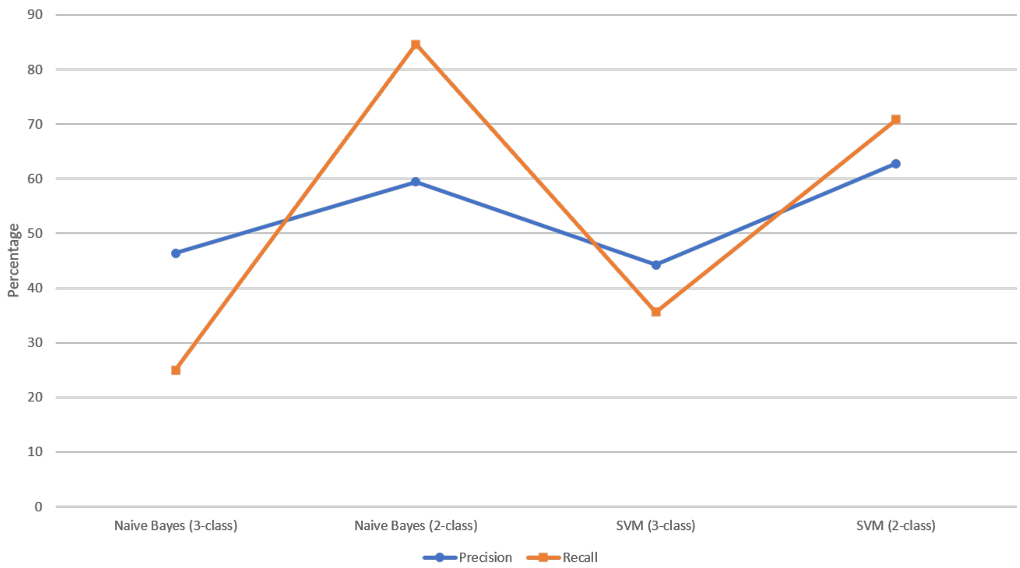

Figure 5.6: Comparing the negative class precision and recall in the two-class and three-class classifications.

Figure 5.5 and Figure 5.6 compare the precision and recall of Naive Bayes and SVM classifiers 10-fold cross validation considering three-class classification and two-class classification for the positive class and the negative class respectively.

The results of this experiment hold with the results of [69] that also trained the Naive Bayes and SVM classifiers on data items with clear polarity. The results also confirm the importance of including the neutral class in the datasets because it resembles a more realistic setting as suggested in [55].

It is worth mentioning here that the introduced lexicon-based sentiment classifier is unsupervised and hence has the advantage of versatility as it does not require a training step on labeled data as is the case with the two other classifiers.

5.6 Summary

In this chapter, the core components of the sentiment analysis pipeline introduced in Chapter 4 are evaluated. For this purpose, a corpus of customer reviews for three market leading fitness trackers is used. The corpus includes 3,241 Reviews containing 10,638 aspect mentions. In order to enable the evaluation of sentiment extraction, a group of 105 reviews evenly distributed among the three devices were manually annotated by student participants for product names, aspect mentions and sentiments (over a Likert scale) to establish a ground-truth dataset. This dataset was used to compare the presented lexicon-based sentiment extractor to two baseline supervised sentiment classifiers, namely Naive Bayes and Support Vector Machines.

The product name extractor achieved an accuracy of 81% identifying product names out of all the names of competing products of the three considered devices. However, examining the ground-truth dataset showed that only one product name mention was not correctly identified in practice.

The aspect extractor component was then compared to related approaches where it was able to recall all the core aspects of the products with minimal noise. The clustering of equivalent aspects also led to the sentiment being assigned always to the proper aspect cluster instead of sentiment mentions being distributed over several equivalent aspects.

For evaluating the sentiment extractor, the baseline classifiers were trained on unigrams as suggested by related work and only sentences containing one aspect mention were considered as this is a limitation of these approaches. The 3-class evaluation of the proposed sentiment extractor resulted in significantly better performance compared to the baseline approaches. Overall, the accuracy of the sentiment extractor reached 70% approximately where that of Naive Bayes and SVM classifiers recorded 37.5% and 41.43% respectively.

By asserting the reliability of the proposed system, performing an aspect based evaluation for market sensor devices can allow an average consumer to leverage the opinions of people who experienced those devices in order to acquire a reliable sensor device that meets his/her preferences.

Realization and Management of Semantic Sensor Data 6

By utilizing both semantic techniques and big data technologies, this chapter discusses a main contribution of the thesis by proposing a scalable approach for semantification and management of vast amounts of sensor data. Introducing semantics to raw sensor data ensures better usability of the data while relying on big data technologies for processing and storage ensures the reliability and the scalability of the approach.

6.1 Sensor Data: A Strategic Information Asset

Modern technical systems and production processes often comprise a large number of sensors providing data about current operating conditions as well as the system's environment. Examples can be found in vehicles (private and industrial), factories ("Industry 4.0"), agriculture (weather and oil sensors) and in energy production (solar and wind power plants). In the past, sensor data has mainly been used for controlling the current operations of a system. Nowadays however, high-frequency data about operating conditions and usage contexts can in addition be seen as a *strategic information asset* for the operator or manufacturer.

This is made possible through the advancements in big data technologies which allow processing the vast amounts of collected sensor data and by the use of the ontological modelling of sensor networks which adds semantic compatibility for raw sensor data. The approach proposed in this chapter harnesses the separation between TBox statements (statements describing the semantic sensor network model) and ABox statements (representing the actual sensor data) to create an encoding scheme that distinguishes how those statements are handled by the big data processing engine, Apache Spark in this case.

This scheme preserves the semantic nature of the data on a conceptual level, thus, preserving semantic access to data records. However, the actual storage of sensor data in Spark RDDs (Resilient Data Records) resembles column-oriented database storage in which sensor data items of the same type are grouped together. Such grouping increases the proximity of sensor data coming from the same sensor type instead of simply storing RDF triples representing the data randomly in Spark RDDs.

In this chapter, the advantages of using semantic sensor data are presented in Section 6.2. Section 6.3 introduces the steps needed to transform raw sensor data into semantic data. In Section 6.4, the aforementioned processing technique of semantic sensor data is detailed. Finally, Section 6.5 introduces a fast prototyping approach for achieving semantic sensor data systems by harnessing the various sensors incorporated in smartphones. This approach was used in the use case of Section 7.3.2. This chapter is partly based on my publication [45] and includes some of its passages.

6.2 Advantages of Semantic Sensor Data

Beyond the challenges with respect to the generation and technical handling of massive sensor data streams, ontology-based transformation of raw sensor data into semantic sensor data helps achieving the following advantages:

- Easy data integration through incorporating different ontologies that describe new application contexts in order to extend the data processing and reasoning capabilities over the collected data.
- Machine-interpretability allowing autonomous or semi-autonomous processing and reasoning about sensor data despite the increasing complexity in sensor networks deployed in modern systems.
- Creation of different abstraction levels where non-technical users operate without the need to be familiar with the technical details of data format and data integration.
- Reduction of data analysis costs due to the need of highly-trained technical staff to handle raw sensor data collections. This is especially critical for small and medium-sized enterprises.
- Explicitly capturing the context of data generation which allows the easy utilization of the sensor data in further applications that they were not initially intended when the data was collected.

6.3 Ontology-Based Modelling of Sensor Data

Two steps are needed to transform raw sensor data into semantic data. The first step in this transformation is the creation of an ontology that represents the conceptual model that describes the sensor network involved in generating the data. While the second step is transforming the actual raw sensor data into semantic sensor data using the created ontology.

For the first step, the Semantic Sensor Network Ontology (SSN) introduced in Section 3.3.1 is used and a base ontology is created by extending the original SSN ontology to describe the deployed sensor network. The Semantic Sensor Network Ontology was developed by the W3C Semantic Sensor Network Incubator Group[1] (SSN-XG) as an ontology to describe sensors and sensor networks for the use in sensor web and sensor network applications. The SSN ontology is created around the Stimulus-Sensor-Observation design pattern [46] where the act of sensing is conceptually separated into three parts: a stimulus, a sensor and an observation. In addition to its *Skeleton* module that describes the sensing activity, several conceptual modules are built in the SSN ontology to cover key sensor concepts such as *Deployment*, *Device*, *System*, *OperatingRestriction* and *Data*. The Semantic Sensor Network ontology was quickly adopted in research and applications and became the standard ontology for semantic sensor networks.

A portion of an ontology describing a sensor network that is created based on the SSN ontology is shown in Figure 6.1. For the readability of the figure, only a single sensor, namely the GPS, is described in the model.

The second step is using a mapping function that creates all the RDF triples needed for each sensor data record. This typically includes a time-stamp of the recorded sensing action, the type of the recorded sensing action and the value or values recorded in this sensing action. An example of this mapping is illustrated in Figure 6.2 where a raw acceleration sensor data record is mapped to five RDF statements.

6.4 Big Data Processing of Semantic Sensor Data

Current and emerging use cases involve huge amounts of raw sensor data which imposes the use of a scalable cluster computing framework that ensures data parallelism and fault-tolerance. Apache Spark [93] is a general engine for large-scale data processing that utilizes in-memory computing to achieve up to 100x better performance than the state of the art MapReduce [28] approach. Spark's efficient performance is a result of using a read-only data structure

[1] https://www.w3.org/2005/Incubator/ssn/

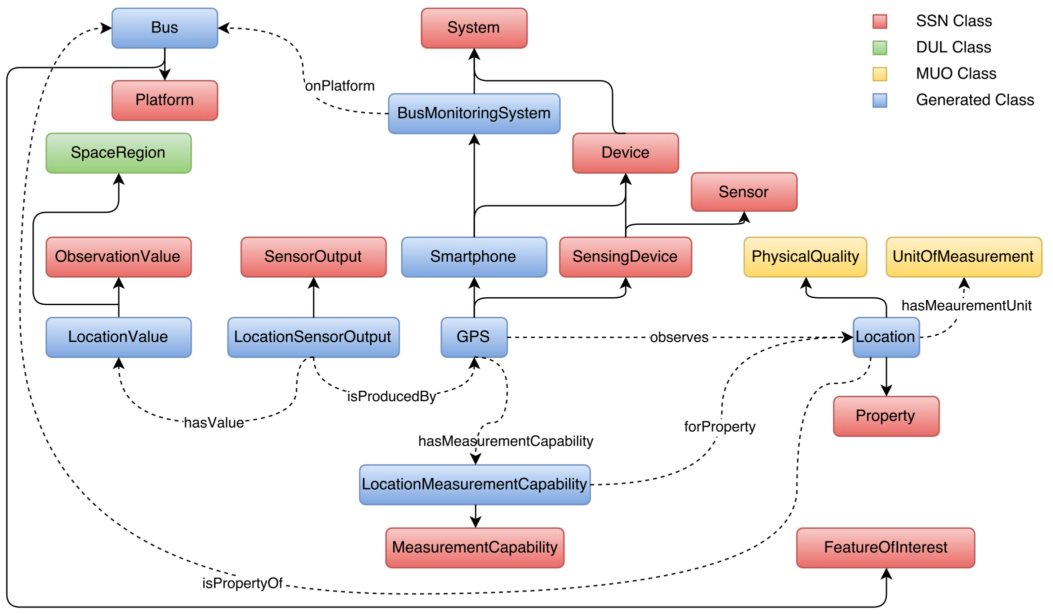

Figure 6.1: Example of a generated SSN-Based ontology describing a smartphone with GPS sensor connected to a bus.

called Resilient Distributed Datasets (RDDs) [92]. The choice of Apache Spark to implement our approach and its comparison to other alternative big data tools was discussed previously in Section 3.4.2. The approach however can be slightly modified to work for other big data processing engines that allow main memory caching.

In any ontology knowledge base, a natural separation exists between *TBox* and *ABox* statements where the former denotes the set of classes and properties that describe the conceptualization of a system and the latter denotes a set of facts about individuals belonging to these classes. This separation is used to distinguish between the base ontology statements and the generated RDF statements representing the collected sensor data. The methods of processing TBox and ABox statements are introduced next in Section 6.4.1 and Section6.4.2 respectively and the means of querying the resulting distributed sensor network knowledge base is introduced in Section 6.4.3.

6.4.1 Processing of TBox Statements

Given the relatively small size of a base ontology, there is no need for a distributed processing of its statements. Instead, the base OWL ontology is processed locally to produce an encoding of the TBox statements which is passed as a Spark broadcast variable to be available during run-time on all

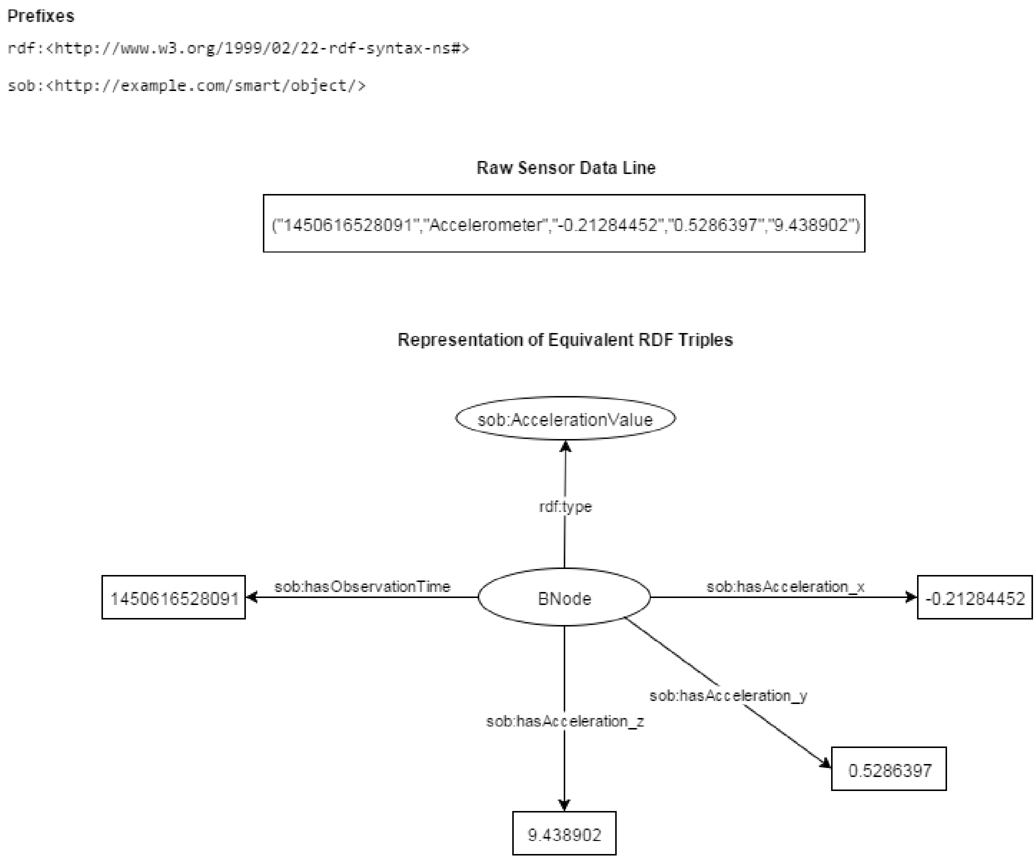

FIGURE 6.2: Example of mapping raw sensor data record to its equivalent RDF triples.

the working nodes of the Spark cluster. Two tables for the classes and the properties of the ontology are created:

- **Ontology Classes**: For each class, a mapping is created from its URI to a unique numerical ID and a set of the IDs of its sub-classes using `rdfs:subClassOf` property.
- **Ontology Properties**: For each property, a mapping is created from its URI to a unique numerical ID and the numerical IDs for its domain(`rdfs:domain`) and range(`rdfs:range`) classes as well as a set of the IDs of its sub-properties using `rdfs:subPropertyOf` property.

The use of numerical IDs to replace string based ontology URIs enhances the utilization of the available memory and improves the performance of identifiers' comparisons. Note that using the transitivity of `rdfs:subClassOf` and `rdfs:subPropertyOf` properties, the set of sub-classes and sub-properties are computed by recursively finding the sub-entities of the direct sub-classes (resp. sub-properties) of a class or a property. [24] uses a similar encoding however does not account for multiple inheritance of classes and properties which is common in most ontologies such as the SSN ontology used here.

6.4.2 Processing of ABox Statements

For the ABox statements, which comprise the vast majority of the statements in a knowledge base, Spark is used in all operations involving these statements starting with the creation of the statements using raw sensor data and in all the following steps including querying the knowledge base and analyzing its data. As indicated in Section 6.3, a set of RDF statements are assigned to each record of sensor data. These statements however, are not stored in a single RDD containing all the triples created. Instead, an RDD is created for each property containing a key-value pair representing the subject and the object of the conceptual RDF triple. Figure 6.3 shows an example outcome for this process where numerical IDs are replaced with textual IDs for clarity. The analogy to column-oriented DBMS enhances the performance where not all triples need to be loaded into main memory as it is the case with many sensor data analysis scenarios such as the use case presented in Section 7.3.2.

Note that restoring the triples format is a straight forward process where for each element in a property RDD, a triple can be formed from the element's key and value as the subject and the object of the triple respectively, and the property represented by the RDD as the property of the triple.

6.4.3 Querying Semantic Sensor Data

In order to query the semantic sensor data using Spark, the query is transformed into an equivalent set of Spark operations. [89] proves that any conjunctive query could be transformed into a set of the following Spark operations:

- **Map**: applies a specified function to all the elements of an RDD.
- **Filter**: returns a new RDD consisting of the subset of the data in an existing RDD that satisfies a certain predicate.
- **Join**: joins two RDDs based on the equality of their respective elements' keys.

This essentially permits the execution of SPARQL queries on the dataset through transforming the SPARQL query into its equivalent set of Spark operations. An example of such transformation is presented in Section 7.3.1. Additionally, analysts could further utilize Spark transformations and actions

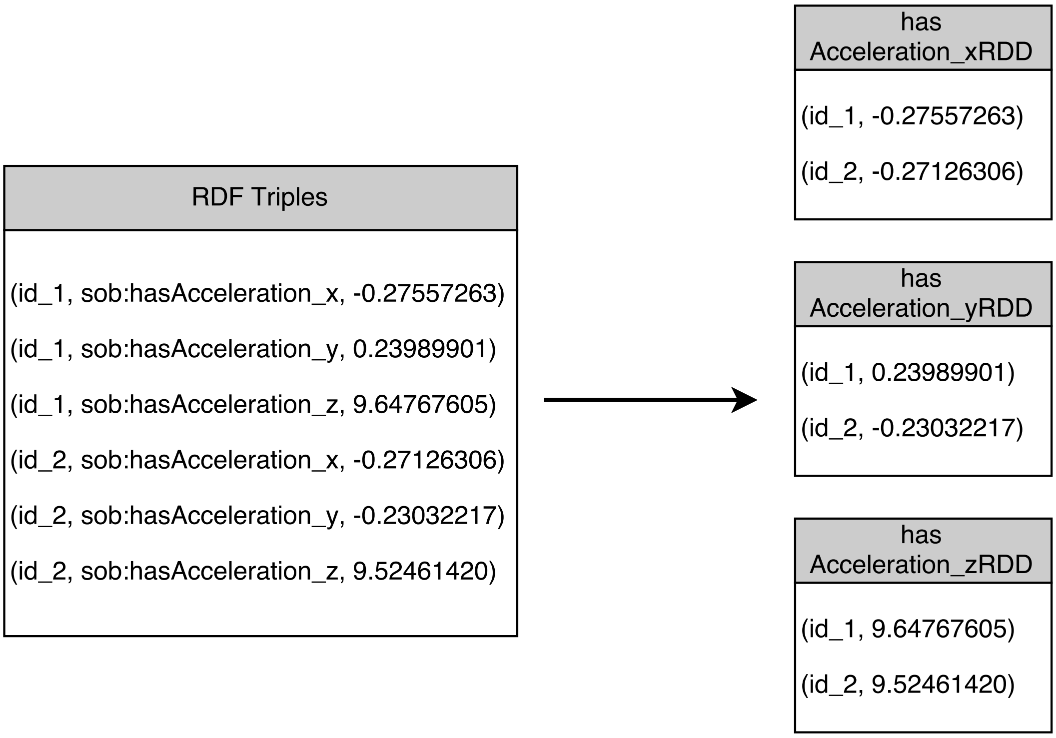

FIGURE 6.3: An example of storing ABox RDF statements in properties' RDDs.

like *sortByKey*, *ReduceByKey*, *count* and *reduce* to get more insights from the dataset.

6.5 Smartphone-Based Prototyping of Semantic Sensor Analysis Systems

To illustrate the potential of harnessing big data and semantic technologies for processing sensor data, we propose a simple prototyping approach that uses smartphones for transforming objects or devices into semantic sensor data sources and providing the means to semantically analyze the captured data. This approach is defined by the following stages:

1. **Generating sensor data**: Smartphones nowadays are equipped with a notable set of sensors including accelerometer, gyroscope, GPS, light sensor, etc. Given its connectivity, storage, and processing features, physically attaching a smartphone to an object of interest is enough to generate sensor data about this object.

2. **Semantic modeling of the object with deployed sensors**: Given a set of sensors being tracked, an ontology is generated based on SSN to model the use of the smartphone and its tracked sensors to monitor the object of interest. The TBox statements forming this ontology are encoded using the scheme presented above (Section 6.4.1).

3. **Transforming raw sensor data into semantic data**: Utilizing the ABox encoding scheme presented in Section 6.4.2, sensor data is transformed to semantic data using the generated SSN-based ontology. The use of big data technologies, namely Apache Spark, aims at facilitating the process of scaling up the system when smartphones are replaced by permanent sensors and the need to process continuous sensor streams arises.

4. **Semantic sensor data analysis**: After obtaining semantic sensor data, the possibilities for materializing semantics, also when combined with other contextual data, can be achieved. This involves the use of Spark queries for robust and scalable performance.

It is worth mentioning that smartphones are not the best option when it comes to deployment of sensors in real life sensor networks scenarios. This is due to their high energy consumption and need for a permanent power connection, operating system overhead and unreliability, relatively large size, and inability to operate in several environments and conditions such as underwater or in high temperature surroundings. However, the effortless deployment of the proposed system can often yield sufficient insights and results for certain simple applications and serves as a prototype that could scale up to meet high demand application scenarios with only few changes required.

The SensorTracker smartphone application used to collect sensor data is next introduced in Section 6.5.1.

By combining the user's choice of tracked sensors in this smartphone application, the user's manual description of the object to which the smartphone is deployed, and the Semantic Sensor Ontology (SSN), an SSN-based ontology is automatically created to describe the sensors attached to the object of interest. Using the simple assumption of attaching the sensing device (smartphone) to any object of interest, and knowing the available sensors on the smartphone, the system generates a base ontology to describe the sensors deployment setup. The generated ontology considers the object of interest as sub-class of both the `ssn:Platform` and `ssn:FeatureOfInterest` classes. Since SSN does not include a model for describing sensor values and their units of measurement, the SSN-based ontology is complimented by the use of the Dolce Ultralite Ontology(DUL) [64] and the Measurement Units Ontology(MUO) [1].

The SensorTracker application detailed next records the sensor data for all the smartphone sensors chosen by the user in CSV format. After parsing the raw data files and discarding any erroneous records, a group of RDF triples are created depending on the type of the sensor in each data record, encoded using the ABox encoding scheme introduced in Section 6.4.2, and added to the knowledge base.

6.5.1 SensorTracker App

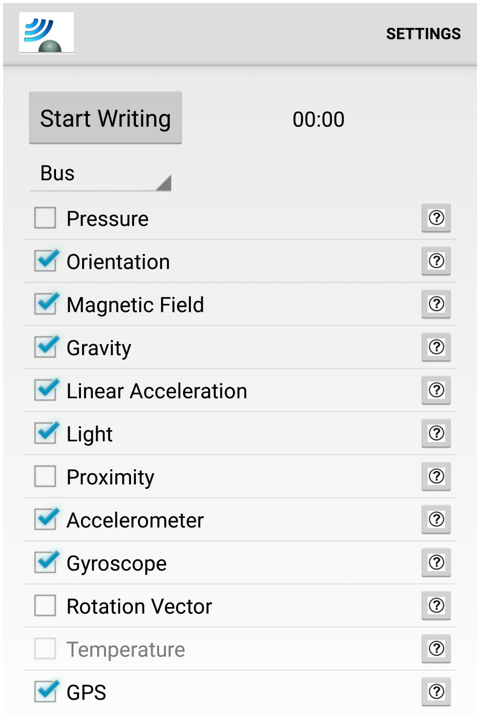

FIGURE 6.4: Snapshot of the SensorTracker Android app interface.

The starting point for realization and analysis of semantic sensor data is the generation o sensor data. Using smartphones for acquiring data has the advantages of abundance of sensors included, connectivity, storage space and processing power. SensorTracker is an Android app developed for the purpose of acquiring sensor data. After attaching the smartphone to the object of interest, sensors to be tracked are easily selected using SensorTracker interface as shown in Figure 6.4. Even though smartphones are not the ideal sources for sensor data streams as indicated before, the following properties were considered to overcome the limitations and utilize the advantages of using a smartphone in building SensorTracker application:

- **Robustness**: SensorTracker runs in the background and was tested for extended running times of several days recording a total of over 14 million records equivalent to 1.2 GB of raw sensor data as detailed in the experiment setup of Section 7.3.1.

- **Storage and Connectivity**: Sensor data is saved in the form of CSV files and can be optionally transferred or inserted to a remote database using internet connectivity when available.

- **Local Processing**: Given the processing power of modern day smartphones, SensorTracker does local aggregation of sensor data within user-specified time frames to reduce redundant data if required. Also, SensorTracker uses sensor fusion techniques to provide fused sensor data and not only data from the phone's physical sensors. For example, using accelerometer, gyroscope and magnetic field sensor data to produce fused orientation sensor data.

6.6 Summary

In this chapter, the challenge of handling the sheer amount of sensor data being generated at high paces today and expecting further growth in the future is addressed. Addressing this challenge is done on two levels.

On the semantic level, using ontological modelling of the sensor data generation context enabled the transformation of raw sensor data to semantic sensor data. This transformation that relies on the widely used SSN ontology, leads to achieving several advantages including simple data integration, machine-interpretability, different levels of abstraction over the data in addition to saving the whole data generation context which allows the utilization of the gathered data in future applications.

On the processing level, an encoding scheme for semantic sensor data is presented. This scheme leverages the in-memory processing capabilities of modern big data storage and processing technologies in order to ensure scalability and reliability for semantic sensor data management. Apache Spark was deployed for the practical implementation of this scheme.

Finally, a prototyping approach for semantic sensor analysis systems is introduced by utilizing the sensors equipped in modern smartphones. This approach uses an Android application, SensorTracker, to collect sensor data. An SSN-based ontology is automatically generated to model all the activated sensors allowing the import of sensor data collected by the application as semantic sensor using the encoding scheme described before. This prototype was used in the street quality assessment application presented in Section 7.3.

Applications and Use-Cases 7

7.1 Introduction

In the previous chapters, two of the main challenges characterizing the growing use of sensors in modern technical systems and consumer products are addressed. On the one hand, the mass introduction of consumer-oriented sensing devices to the market necessitates the development of a dynamic approach for evaluating these devices in order to facilitate the acquisition of reliable sensor products by the average consumer. This goal was achieved through the development of the aspect-based sentiment analysis pipeline presented in Chapter 4 that evaluates all the features of a consumer sensing device using customer reviews of the device.

On the other hand, handling the sheer amount of sensor data generated by consumer sensor devices as well as commercial sensor networks is also addressed. Chapter 6 introduces an approach that couples semantic technologies with big data technologies in order to provide a solution that not only provides an efficient and scalable method for storing and processing of sensor data, but also facilitates data integration, data analysis and the future utilization of collected sensor data through semantification of raw sensor data. It further introduces a prototyping approach for achieving semantic sensor data systems harnessing the multiple sensors incorporated in smartphones.

This chapter presents two real-life use cases of the formerly presented approaches. In Section 7.2, an evaluation of the three market-leading fitness trackers previously introduced in Section 2.5 is presented following the approach discussed in Chapter 4. In Section 7.3, a street-quality assessment application using the prototyping approach presented in Chapter 6 is introduced.

Additionally, and to demonstrate the potential of harnessing publicly available collections of sensor data, Section 7.4.2 introduces a recommender system application that analyses publicly available running route data to provide personalized suggestions for new routes considering the runner's performance, visual and nature of route preferences. The applications presented in this chapter are part of my publications [44], [43], and [45]. This chapter includes some of their passages and figures.

7.2 User-Sentiment-Based Evaluation for Market Fitness Trackers

Wearable fitness and health trackers have been in an accelerated growth since their recent introduction to consumer markets. Given the growth potential of this market sector, much more devices, with unbounded set of features, are being introduced to the consumer at a fast pace. The increasing public interest in fitness trackers as consumer products, in addition to their potential uses in health care necessitate the presence of proper evaluations for all the aspects of these devices. The huge number of devices available on market and the speed in which new devices with new features are released, makes it very difficult to come up with a proper evaluation for market products that won't quickly turn obsolete especially when taking longer device usage-time as a factor in evaluation.

Very little work has been done on evaluating market sensor devices. In [25], the energy expenditure of footwear-based physical activity monitors was studied. The study used devices from the following manufacturers: Actical, Actigraph, IDEEA, DirectLife and Fitbit. In a more similar work, [38] performed a study to compare two popular fitness trackers, namely Fitbit One and Nike+ Fuelband in terms of accuracy, type of data provided, available APIs, and user experience. The outcome of this practical study is consistent with the evaluation results detailed in this section.

In this section, a comprehensive evaluation of the top three market devices from manufacturers that together dominated over 97% of the fitness trackers market at the time of the evaluation according to NPD Group[1] is presented. The devices, previously introduced in Section 2.5 are Fitbit One, Jawbone Up and Nike+ Fuelband. It is worth mentioning that the choice of these three products is strictly for their leading position in the consumer market. The evaluation is the achieved by utilizing the framework introduced in Chapter 4 that evaluates market sensor devices based on the public opinion expressed through customer reviews. Table 7.1, Table 7.2 and Table 7.3 show the aspects detected by the system and the distribution of the sentiments among these aspects for Fitbit One, Jawbone Up and Nike+ Fuelband respectively.

[1]https://www.npd.com/latest-reports/consumer-technology-reports/

7.2.1 Controversial Aspects

Regardless of the sentiment, the frequency of aspect mentions in reviews is very significant, especially for evaluating distinguished aspects that set a product apart from its market competitors. For example, the NikeFuel activity measure introduced in the Nike+ Fuelband is the most discussed aspect for this device with 22.7% of the Fuelband's aspect mentions discussing this unique aspect. In contrast, the flower activity indicator in the Fitbit One, which resembles NikeFuel for the Fuelband, takes a share of 0.7% only from the Fitbit's aspect mentions and it is the least discussed aspect for the device. This indicates that Nike's distinguished activity measure has successfully captured the interest of consumers whereas Fitbit falls short in this regard. The Power Nap aspect of Jawbone Up, which is a feature that calculates the optimal nap duration for the user and wakes them up through gentle vibrations, is barely discussed with 0.7% of aspect mentions despite being an innovative and positively received feature. This signifies that the usability of this feature can be questioned. However, Sleep assessment, in general, has proven to be an important feature to users grabbing over 20% of Jawbone's aspect mentions and about 15% of Fitbit's aspect mentions. This indicates that Nike is missing a highly demanded feature by not including it in their device. Also, Fitbit's ability to compute floors climbed holds over 8% of its aspect mentions almost equal to the share of "Calories Burned" feature and thus it also represents a feature demanded by consumers.

Table 7.1: Fitbit One results

Aspect	**Positive**	**Negative**	**Neutral**	**Total**
Steps Counter	605	183	492	1280
Sleep Assessment	570	199	373	1142
Syncing	408	146	263	817
Calories Burned	315	113	245	673
Floors Climbed	276	99	267	642
App Functionality	332	114	182	628
Website	332	85	176	593
Battery	206	100	201	507
Price	186	102	124	412
Pocket/Clipping	109	67	112	288
Customer Service	148	50	60	258
Distance Traveled	90	23	88	201
Badges	71	12	32	115
MyFitnessPal	38	9	21	68
Activity Indicator (Flower)	36	8	14	58
Total	3722	1310	2650	7682

Table 7.2: Jawbone Up results

Aspect	**Positive**	**Negative**	**Neutral**	**Total**
Sleep Assessment	225	82	128	435
Battery	113	113	123	349
App Functionality	159	57	95	311
Steps Counter	90	30	97	217
Syncing	78	65	66	209
Price	63	55	30	148
Customer Service	51	31	45	127
Alarm	51	15	24	90
Calories Burned	38	13	27	78
Food Logging	28	18	15	61
Website	19	5	16	40
Distance Traveled	17	2	16	35
Powernap	6	2	8	16
Pocket	7	4	3	14
Activity Logging	5	1	2	8
Total	950	493	695	2138

Table 7.3: Nike+ Fuelband results

Aspect	**Positive**	**Negative**	**Neutral**	**Total**
NikeFuel	70	49	67	186
Price	41	29	38	108
Battery	35	21	41	97
Calories Burned	33	21	38	92
Steps Counter	29	22	31	82
App Functionality	38	15	22	75
Syncing	29	20	18	67
Website	19	10	23	52
Customer Service	10	13	9	32
Distance Traveled	8	6	13	27
Total	312	206	300	818

7.2.2 Performance Aspects Evaluation

In this section, the public satisfaction of the common performance-related aspects in all three devices is assessed. Namely: *Steps Counter, Distance Traveled and Calories Burned*. It is worth mentioning that similar insight and figures can be generated by the sentiment summarizer for all the other aspects. However, we display focus here on the common performance aspects and later discuss useful insights for each device in Section 7.2.3.

7.2.2.1 Steps Counter Evaluation

All three devices seem to be satisfactory to consumers with respect to steps counting as shown in Figure 7.1 despite a significant advantage for the Fitbit One. This actually supports the physical testing of accuracy of several fitness trackers including Fitbit One and Nike+ Fuelband by [38] in which the One outperformed all other devices.

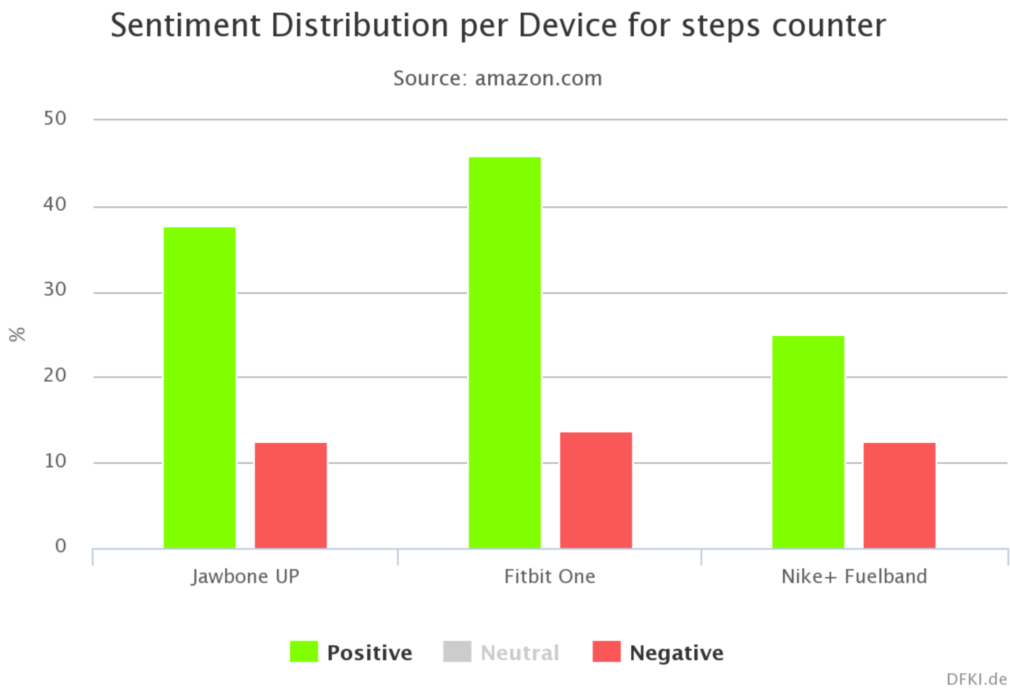

FIGURE 7.1: Results of the public sentiment evaluation for aspect ”Steps Counter” as displayed by the sentiment summarizer

7.2.2.2 Distance Travelled Evaluation

The good performance in counting the steps by the devices leads to a good estimation of the distance travelled by users as shown in Figure 7.2. Since all three devices do not include location sensors, the distance value is calculated by all these devices through estimations based on the individual's personal data provided to the device during the set up process.

7.2.2.3 Calories Burned Evaluation

In a similar manner, the devices estimate calories burned based on the performance of the user and his/her entered physical data (e.g. height, age, weight, gender, etc.). Both Fitbit One and Jawbone Up seem to be equally

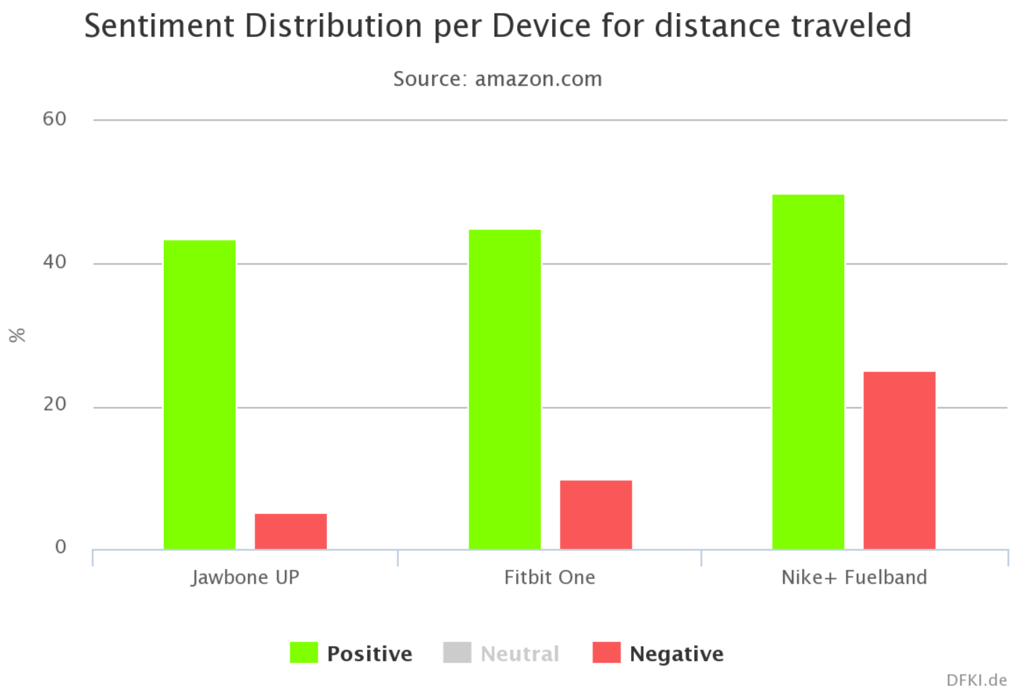

Figure 7.2: Results of the public sentiment evaluation for aspect ”Distance Travelled” as displayed by the sentiment summarizer

satisfactory for users estimating the calories burned whereas the Fuelband has a significantly less positive-to-negative mention ratio compared to the two other devices as shown in Figure 7.3.

7.2.3 Further Discussion and Insights

In addition to the statistics provided, the sentiment summarizer presented in Section 4.8 allows for easy tracking of positive, negative and neutral mentions of each aspect of every device throughout the reviews. For example (Figure 7.4), shows a part of the list of all the negative mentions of ’Customer Service’ for Jawbone Up. This gives a new dimension in the evaluation by highlighting pros and cons based on the experience of customers. Many useful conclusions can be drawn from people’s opinion. Some of them are given below.

7.2.3.1 Fitbit One

Many proponents of Fitbit commented on how it helped them greatly pay attention to their sleep quality and how it encouraged better sleeping habits. The ratio between the number of users in favor of the sleep assessment made by the tracker to those against it is almost 3:1. Some customers however, found the

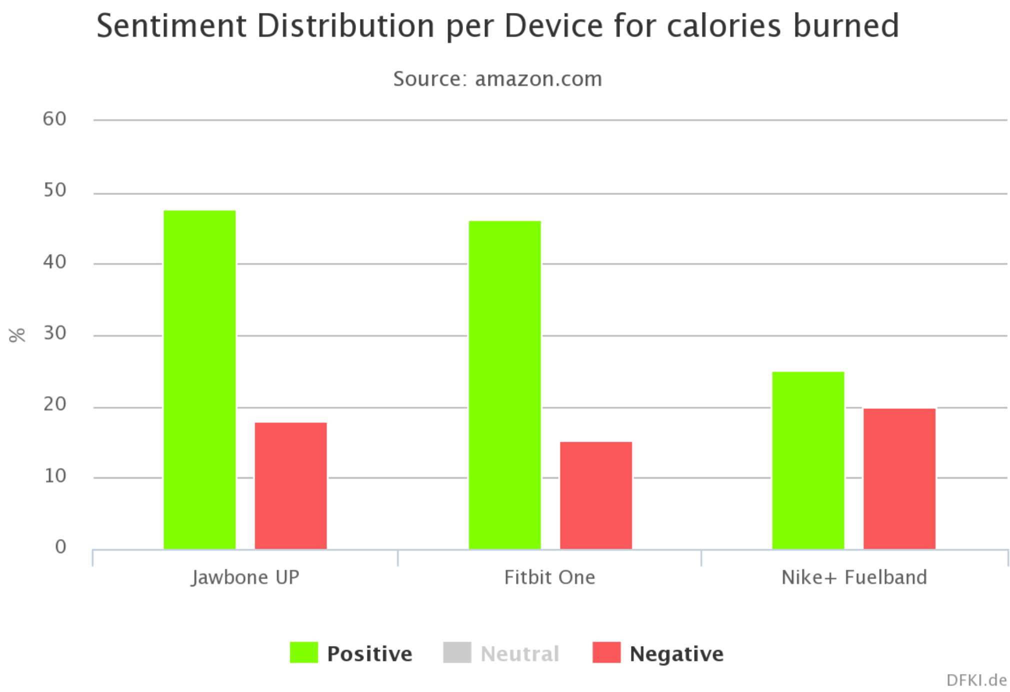

FIGURE 7.3: Results of the public sentiment evaluation for aspect "Calories Burned" as displayed by the sentiment summarizer

sleeping band to be uncomfortable or the sleeping data to be inaccurate. One of them said: *"Tossing in the middle of the night or pushing covers off your body will register as waking up, and watching an hour or two of television without moving your hand before falling asleep will count as sleep time. Unfortunately, this makes the sleep tracker a good idea that is not very accurate."*

Some reviewers commented that it holds a good charge (e.g., charging it only on weekends) and some were pleased that it emails them a reminder when the charge is low. Quoting a user *"The charging cord and Bluetooth dongle are better executed than the stand used in the previous version, and it's MUCH easier to travel with."*. However, some said that their Fitbits did not hold a charge for more than 8-12 hours and others experienced the total death of the device.

Many users find setting up the Fitbit One and syncing the data very easy and smooth. They also love syncing the Fitbit with many third-party apps especially to MyFitnessPal app. In fact, the system identified MyFitnessPal as one of the aspects of Fitbit because people comment a lot on syncing their Fitbits to it. For the Fitbit One app itself, some people claim that it is buggy for most Androids. Others complained that it requires a dongle to sync to PC or MAC saying that the dongle is very small and can be easily lost. Some

I am glad I got to try this product for free through my job rather than wasting over a hundred dollars on an item with poor build and poor customer service.

If there is any problem with the product, you call customer service and they replace or resolve the problem, even sometimes over the phone.

Terrible customer service.

I ran through all the steps I'd learned from customer support earlier, but the problem continued until the bracelet actually crashed my iPhone.

Very disappointed with the Jawbone customer service!

Customer service doesn't seem very responsive!

However, there is no manual and customer support is unresponsive.

I have never, not once, been told by any customer support agent ANYWHERE that they couldn't exchange an item for me or help me with a problem just because I didn't purchase it directly from the company.

Figure 7.4: Snapshot of a part of the list of negative *Customer Service* aspect mentions for Jawbone Up as shown in the sentiment summarizer.

could not pair it to Android. A frequent traveler said that the time shown after syncing is not accurate.

Although people are pleased that it is small and light, many people lost it because, according to them, it can easily fall out of the holder and the holder does not attach firmly to clothes. Therefore, they highly recommend having it inside the pocket rather than clipped onto the pocket or belt. Despite some who are motivated by the growth of the ”activity-progress flower”, many others believe that Fitbit should change this visualization.

7.2.3.2 Jawbone Up

The most popular feature of Jawbone Up is sleep assessment. Most of the customers who reviewed this device expressed how they like its accompanying features, namely the silent vibrating alarm, Power Nap which wakes a user up at the optimal time, and the Idle Alert which reminds users to move when they have been inactive for too long. Nonetheless, many people complained of having to change it manually to sleep mode before sleeping and changing it back to awake mode later, which one can forget. Some customers also questioned the accuracy of the sleep information the band provides.

A major problem many users faced is the complete death of the band after a period of use. Nonetheless, many users said that the customer service replaced their bands. On the other hand, some described the customer service as poor and one of the complaints said that they could not exchange the item or help with its problems because it was not bought directly from the Jawbone company.

Many people recommend calibrating the band to record the distance accurately. Some sentences analyzed by the summarizer stated that wearing the Jawbone on ankle is said to increase the accuracy of the band, yet, some commented that it is uncomfortable.

Some customers said that they did not experience any problems with Syncing. However, the Jawbone UP does not sync wirelessly which made some complain about having to take the band cap off and inserting the band in the headphone's jack of the phone every time. Last but not least, the Jawbone Up app was generally positively rated by most consumers.

7.2.3.3 Nike+ Fuelband

NikeFuel score is perhaps the most remarkable feature of the Fuelband being, by far, the most reviewed aspect of the band as shown in Table 7.3. By reading some of the sentences discussing this aspect, the Fuel score seems very motivating for many. However, some are skeptical of how it is calculated. A user said *"I've noticed that the Fuel really adds up when running and walking but not as much with other workouts that are way more intense than a jog on the treadmill."*. Another said *"Of course, if you shake your wrist while on the elliptical or jogging, you can increase your points by 50%."*. Some people said that it is very motivational, however, they did not recommend it for people who want to accurately track their steps.

With respect to syncing, some people found syncing over the phone or USB to be smooth and easy. On the other hand, it failed to sync with some and was described as slow. Some users complained that the screws in the band rust. One suggested using a fingernail polish to stop the rust. Many users liked the accompanying app which was described as motivational, easy-to-use and well-designed. In fact, the number of positive reviews on the Nike+ Fuelband app is more than twice the negative reviews on it. Yet, some were disappointed that it did not come with an Android app.

7.3 Street Quality Assessment

Bad street quality leads to damages in vehicles, traffic and might cause serious threats to both drivers and pedestrians. Bad street quality can be a result of multiple anomalies such as potholes, manholes, bumps and any other damages to the street that disturb the driving experience. Detecting these street anomalies can help governments and municipalities fix them quickly maintaining street conditions that are safe and comfortable for drivers, bike riders as well as pedestrians. An equally important use for detecting street anomalies is the ability to warn drivers approaching a street anomaly especially at night if this feature is provided through GPS integration or through a smartphone app.

In this section, a street anomaly detection system is proposed to highlight the smartphone-based prototyping approach for semantic analysis systems presented in Chapter 6. The main motivation for using the proposed approach is being able to effortlessly obtain a working prototype of sensors attached to an object of interest, modeling the obtained sensor deployment according to the SSN ontology, and enabling the design of big data algorithms and results analysis to run on the collected sensor data in addition to other possible integrated data from different sources. This usage scenario conveniently scales up when applied to different sensor data sources and much larger amounts of collected data. In what follows, Section 7.3.1 introduces the experiment setup where a public transport bus sensor data is collected and Section 7.3.2 discusses the application details and outcome.

7.3.1 Setup and Collected Data

In order to assess the prototyping approach of Section 6.5, we used a simple setup where an Android smartphone (Samsung Galaxy Note 4) was attached to a bus of the Rhine-Neckar public transport network (VRN[2]) that operates on multiple lines in the city of Kaiserslautern in Germany. The smartphone was stored inside a closed cabinet in the bus and connected to a stable power supply through a USB cable. To achieve consistent sensor recordings that reflect the movements of the bus rather than the smartphone itself, the phone was fixed to a keep a vertical position along the experiment period of 8 days. Over this time, the app recorded a total of $14,248,629$ records in which the bus crossed a distance of over 1600 km.

Figure 7.5 shows the trajectory followed by the bus during the 8-day experiment in addition to the bus stops located in Kaiserslautern.

To illustrate the ability to query the data using the proposed system as introduced in Section 6.4.3, the following SPARQL query to retrieve all bus

[2] http://www.vrn.de/

FIGURE 7.5: Bus trajectory followed during the experiment(purple) with locations of bus stops(green).

trajectory points is presented with its equivalence in Spark using Spark's Python API:

```
SELECT ?lat ?lon
    WHERE{
?a rdf:type sob:LocationValue.
?a sob:hasLatitude ?lat.
?a sob:hasLongitude ?lon.
}
```

which is equivalent to the following operations pipeline in Spark:

```
(TypeRDD
 .filter(lambda (nodeID,typeID):
        equalsType(typeID,"sob:LocationValue") )
 .join(hasLatitudeRDD)
 .join(hasLongitudeRDD)
 .map(lambda (nodeID,(typeID,lat,lon)): (lat,lon))
).collect()
```

Note that, following the encoding scheme presented in Section 6.4, the method `equalsType()` uses the TBox encoding availability in memory to verify that the node type is either equal to the class `sob:LocationValue` or to one of its sub-classes.

7.3.2 Street Anomaly Detection

As an application for the analysis of the collected public transport bus data using the proposed system, the acceleration and the location data are utilized to assess the quality of the streets of the city through detecting street anomalies. By providing an experimental definition of an anomaly as a location where a spike of at least $1.8m/s^2$ occurs within a 2 seconds time frame along the z-acceleration axis, we were able to locate positions of street anomalies on the streets travelled by the bus. Figure 7.6 shows the bus acceleration data along all axes with a clear spike in z-acceleration value indicating the presence of a street anomaly.

In this use case, the focus is on demonstrating the ease of building applications and semantic analysis on top of the obtained sensor data knowledge base. The evaluation of the precision of the introduced street anomaly detector is hence omitted. Still, as a refinement of the results, clustering of the retrieved street anomaly locations is applied using DBSCAN [33] with Haversine distance [78] as the spatial distance metric. The results of this clustering is shown in Figure 7.7.

In a related application, [32] utilizes acceleration data from taxi cars to distinguish the different types of street anomalies. Using smartphones, the detection of anomalies is computed locally on the phone where only locations of detected anomalies are stored and later clustered in order to obtain a final result. As this might be sufficient for this exact use case, the original sensor data collected and processed locally in smartphones is lost and could not be utilized in any further applications different from the original application context of detecting potholes. Following our proposed approach, all the raw sensor data is be stored and processed using big data techniques. And by storing the data in semantic form, many other applications could be built on top of the data by integrating different information sources. Potential examples are analyzing the demands for taxis at times of concerts or sports events, analyzing the activity per taxi

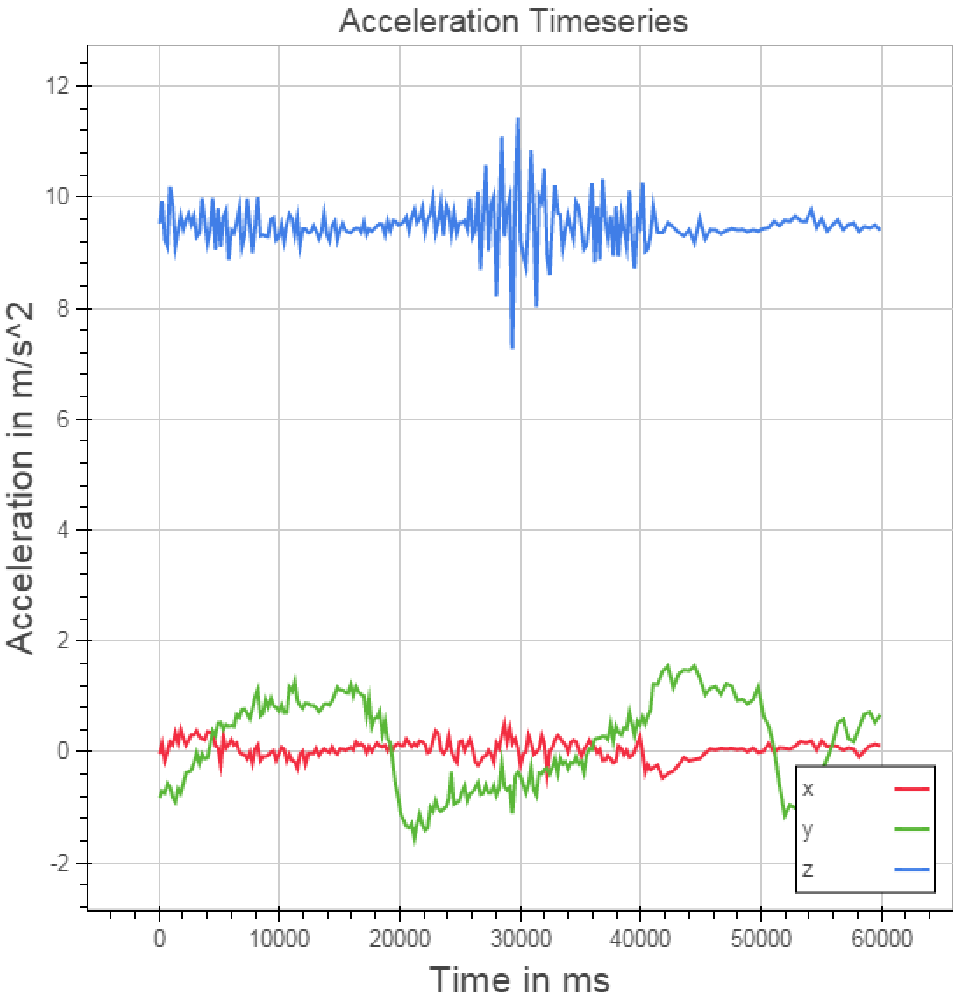

FIGURE 7.6: One minute of acceleration data with a spike in acceleration across z-axis denoting presence of a street anomaly.

station, and analyzing taxi delays if integrated with the company reservations data.

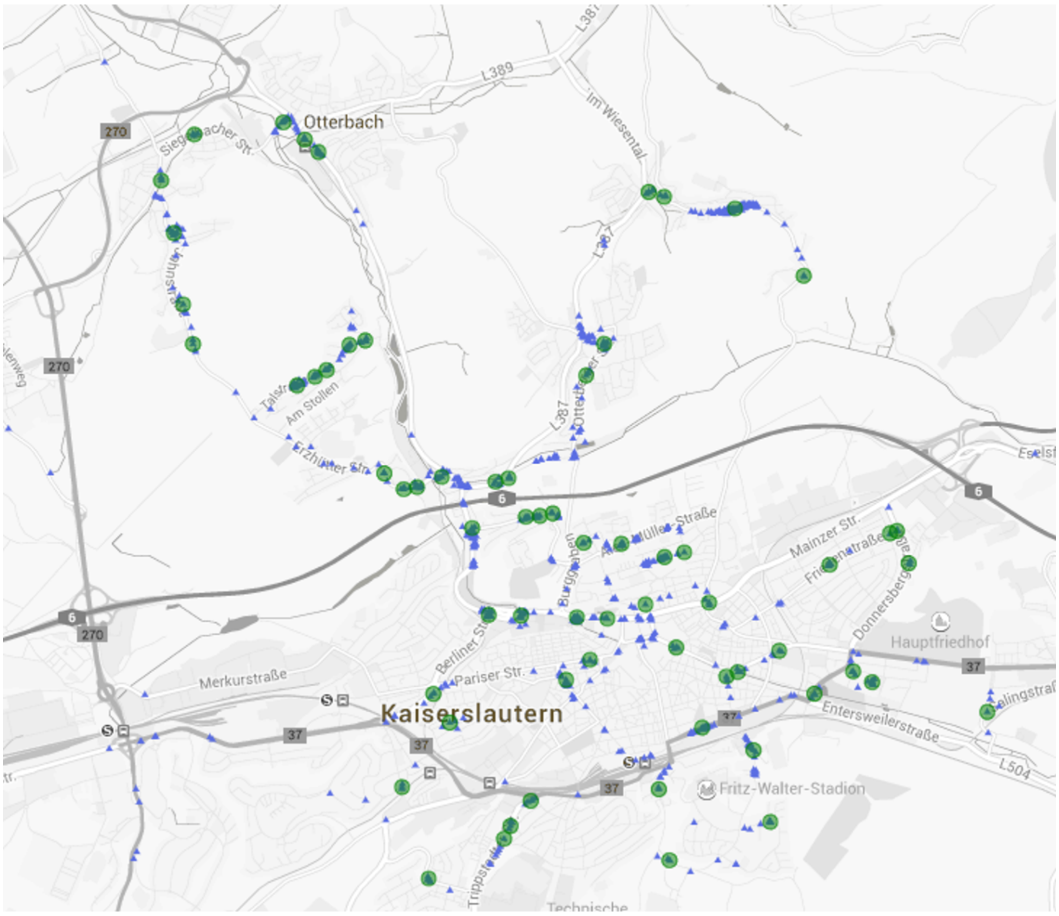

Figure 7.7: Locations of clusters of street anomalies (green) with single anomalies displayed in blue color.

7.4 Preference Based Filtering and Recommendations for Running Routes

Personal fitness have been gaining an increasing attention from both hardware manufacturers and software developers in the recent years. Utilizing the built-in smartphone sensors and GPS, many apps have been built to monitor the activity of users and provide useful insights and recommendations based on their performance. Interestingly, sensor data gathered by users through many different apps and devices are aggregated in online fitness platforms such as MapMyFitness[3], which integrates with more than 400 fitness tracking devices, sensors and wearables and contains data of over 160 millions of running, cycling and walking routes around the world.

[3]http://about.mapmyfitness.com/

The route data is mainly used to retrieve nearby routes based on the user's current location. [41] highlights the significance of MapMyFitness data to place physical activity into Neighborhood Context. Proximity, however, is not the only feature that a person considers in their choice of suitable running routes. For that, in this section we use this collection of running data to demonstrate the potential of harvesting publicly available data collection to provide an added value for runners in this use case.

In this section, several other aspects of running routes are considered to enable recommending the users routes that best fit their preferences. In this context, the considered features of running routes fall into three categories, namely:

- ***Performance features:*** such as distance and variation in elevation.
- ***Visual features:*** describing the route's surrounding environment such as proximity to water or to parks.
- ***Nature of route:*** such as whether a route is a track or not, an on-road or an off-road route and whether it ends at its starting point.

Although there has not been known work utilizing the features listed above for running routes recommendations, few studies share a similar objective though using different approaches.[73] uses Flickr[4] meta-data to determine pleasant locations and suggests more beautiful walking routes to destinations accordingly. [53] applies artificial neural networks as a data mining methodology for a context-aware running route recommender system. A walking route recommender system considering route safety, amenity and walkability is introduced in [76]. Finally, studies such as [19] and [70] introduced preference-based route navigation for car drivers.

In what follows, a classification of running routes based on different features of the route is proposed in Section 7.4.1. Based on this classification, Section 7.4.2 presents a route recommender system that is designed to fit the user's needs based on her preferences and performance history. This spares the user the need to set his/her preferences when looking up a running route by learning his/her preferences over time. The recommender system is evaluated in Section 7.4.3 using data from active runners assuming the use case where users are recommended running routes that match their preferences in new locations in which they had no previous activity.

7.4.1 Filtering Routes by Features

A running route is basically a set of ordered location points denoting longitude, latitude and elevation. Through utilizing these data points, several features

[4]https://www.flickr.com/

of the route are inferred enabling the classification and filtering of routes to match the personal preferences of any individual. This section describes the significance of the considered features and the approaches used in their computation. It is worth mentioning that in the following computations the original data points are sampled using Ramer-Douglas-Peucker algorithm [30] to obtain a sufficiently similar route using a much smaller subset of the route data points. This step enhances the performance especially computations that require external API calls.

7.4.1.1 Performance Features

- **Distance**

  The distance of a route is perhaps the most critical feature for people when deciding if a route is suitable for them. Usually the distance is provided among other meta-data in fitness information systems like *MapMyFitness.com*. However, if not provided, distance between two points in a route are accurately calculated using the Haversine formula [78] which computes great-circle distances between two points on a sphere using their longitudes and latitudes.

- **Variation in Elevation**

  The loss and gain in elevation along running routes are vital for quantifying their strenuousness with respect to steepness. The variation in elevation is represented by two distinct values, which are the *total descent* and *total ascent*. The *total ascent* value denotes the sum of upward vertical distance covered by the runner in order to complete the route. The total descent value is the value of downward vertical distance that the route entails.

7.4.1.2 Nature of Route

According to International Association of Athletics Federation[5] (IAAF), running events are classified upon the nature of their location into track, road and cross-country running. Accordingly, the following list introduces an approach to verify whether a route is a running track or not and an approach that distinguishes between road and cross-country running routes.

- **Running Track**

  Running tracks are characterized by their standardized shape and length. In order to assess if a route is a track or not, a supervised learning approach is applied. Considering the convex shape of a running track, all the points defining the track must intuitively trace, in close proximity,

[5] http://www.iaaf.org/disciplines

the smallest convex set containing these points, i.e. their convex hull. Figure 7.8 and Figure 7.9 show the convex hull defined by a running track and an arbitrary route respectively. The Quickhull algorithm [6] is used to compute the convex hull for each running route. Two features are then computed to enable the classification process, namely:

- ***Average distance to convex hull:*** which is the average of all the distances between the points constituting a route and its convex hull. This feature distinguishes convex and non-convex routes.
- ***Convex hull area/perimeter ratio:*** which helps distinguish an arbitrary convex-shaped route from a proper running track.

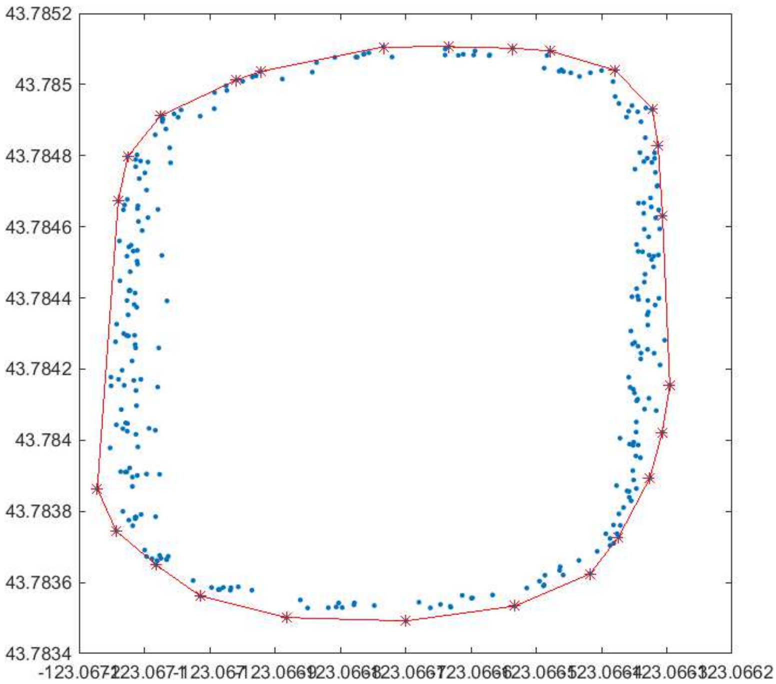

FIGURE 7.8: The convex hull defined by the points of a running track.

Using a set of 100 labeled routes divided across 20% training data and 80% testing data, the Naive Bayes classifier is used to achieve a 100% accuracy in the classification of routes as tracks or non-tracks.

- **On-Road and Off-Road Routes**

  In order to classify whether a route is on-road or off-road, a mapping API is used to check the proximity of each of the route points to the nearest road. To compensate for GPS inaccuracy, a threshold is experimentally

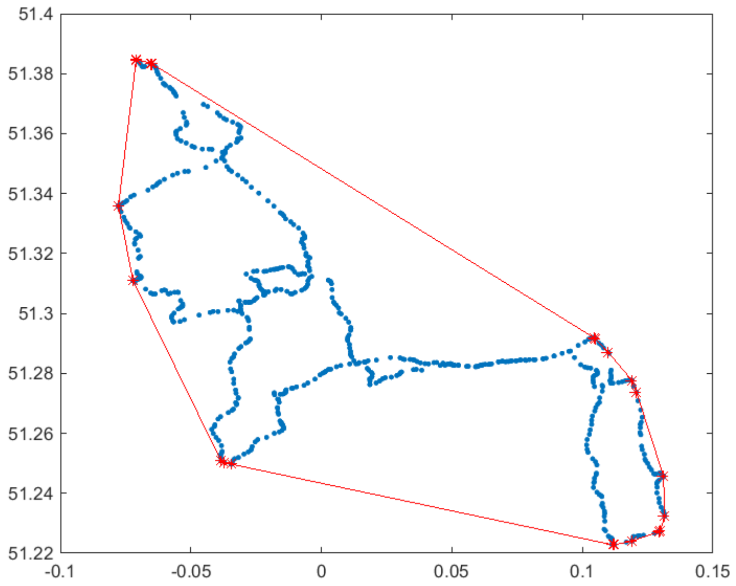

Figure 7.9: The convex hull defined by the points of an arbitrary route.

chosen above which a point is considered off-road. Given that running routes can be composed of both on-road and off-road segments, a route is thus described by the percentages of its on-road and off-road segments.

- **Same Starting and Ending point**

  People often try to end their runs in the same place of their start to guarantee an equal total ascents and total descents in their runs. This also is useful in situations where the runner starts from a parking spot or near a bus stop for example. This is simply verified by assessing the proximity of the starting and ending points of a route.

7.4.1.3 Visual Features

The choice of a perfect running route is also influenced by the route's surrounding environment. In this Section, the proximity to parks and water sources (e.g. lake, sea, etc.) is considered. The same approach, however, could be extended to include other places of interest.

- **Using Google Places API**

  Google Places API is used to check the proximity of route's points to a park or water source. In addition to the performance overhead using API calls, several parks and water sources were not recognized by this method.

- **Using Color Coding for Each Point**

  Mapping APIs use different colors to annotate different types of terrain in a map. Following the retrieval of an image showing the pixels surrounding each of the route's points as shown in Figure 7.10, scanning for the color code of water sources and parks enables the verifying the route's proximity to them. This approach is highly precise, however it is computationally expensive as it requires an API call followed by a color scan for each point in the route.

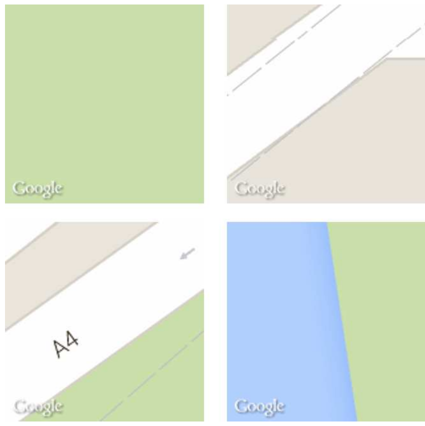

FIGURE 7.10: Examples of pixels surrounding route points using Google Maps API.

- **Using Color Coding for the Whole Route**

  By applying a similar approach to *"Using Color Coding for Each Point"*, however retrieving one pixel map surrounding the whole route as shown in Figure 7.11 and mapping points of the route to an array of pixels in the retrieved image, highly accurate results are achieved even with reducing the number of API calls to one per route.

  Evaluated using 150 labeled routes, this method resulted in a precision of 98.3% and a recall 96.77% for park proximity and a precision of 100% and a recall of 91.42% for water source proximity. Table 7.4 and Table 7.5 show the evaluation results for park proximity and water source proximity respectively.

7.4.2 Running Route Recommender System

The goal for building a running route recommender system is to be able to provide a user with running routes recommendations in any location, especially in new locations where the user has a little knowledge of the area and potential routes that might match her preferences. The running route recommender system is chosen to be content-based and uses the features introduced in

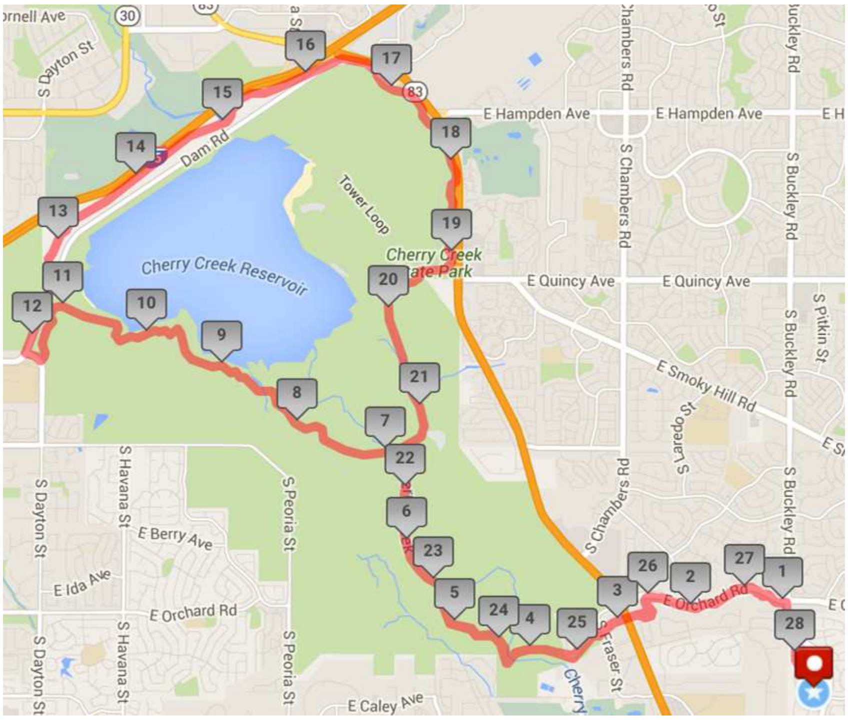

Figure 7.11: A pixel map surrounding a route as retrieved from Google Maps API.

Table 7.4: Results of inferring proximity to a park.

Inferred \ Truth	Park	No Park	Total
Park	60	1	61
No Park	2	87	89
Total	62	88	150

Section 7.4.1 to describe a route. Collaborative Filtering, which relies on the notion that users who have had similar preferences in the past are likely to have similar preferences in the future, could hardly be applied in this context. Arguably, collaborative filtering could be used if the recommendations are limited to locations where a user has a running history, this however does not apply for the intended use case where the recommender system should provide relevant routes in any location of the user's choice.

7.4.2.1 Recommender System Overview

Figure 7.12 shows the main components of the proposed recommender system. The system takes as input two lists representing the user routes and the location routes. Since each route is represented by mixed numerical and

Table 7.5: Results of inferring proximity to a water source.

Inferred \ Truth	Water	No Water	Total
Water	32	0	32
No Water	3	115	118
Total	35	115	150

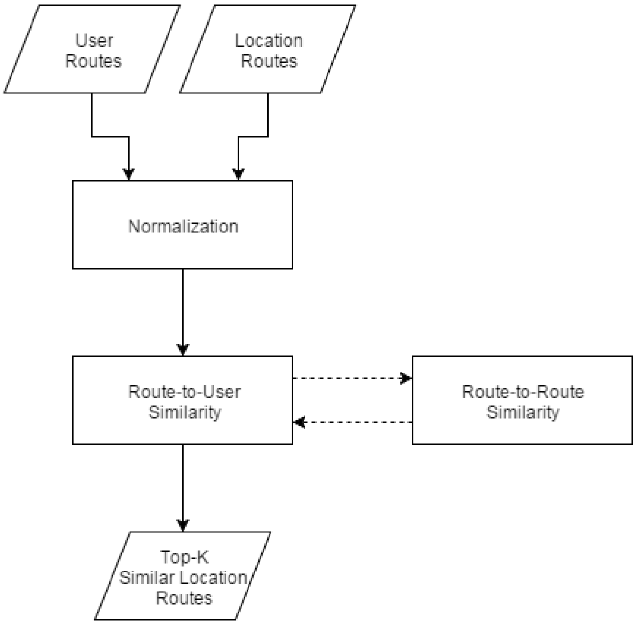

FIGURE 7.12: Overview of the running route recommender system.

categorical features, a statistical approach for normalization of mixed metrics is applied as introduced in [81] where the contribution of each feature to the similarity measure is divided by the contribution mean for this feature. Several approaches are introduced to compute similarity of a route to a user, all of which rely on a consistent route-to-route similarity computation. After a proper aggregation of the route-to-user similarities, the top-k location routes similar to a user are returned as recommendations.

7.4.2.2 Similarity Measurement

A normalized route r is defined as an $n - dimensional$ vector representing the n numeric and categorical features of a route namely: distance, elevation, percentage of on-road segment, same start and end point, close to a water source, close to a park, and represents a track.

$$r = (dis, ele, road, closed, water, park, track) \tag{7.1}$$

Note that elevation is a combined feature of total ascents and total descents in a route as shown in Formula 7.2. The amount of these contributions, indicated by α, is determined experimentally in Section 7.4.3.3. Intuitively, ascents contribute more to the elevation feature as they have a huge impact on the difficulty of a running route.

$$ele = \alpha.asc + (1 - \alpha).des \tag{7.2}$$

The similarity of two routes is defined as the Euclidean distance separating the two routes and is defined in Formula 7.3. T

$$ED(r, r') = \sqrt{\sum_{i=1}^{n} (r_i - r'_i)^2} \tag{7.3}$$

This similarity is applied to compute the similarity of a route to a user using the following multiple approaches:

- **Average Route-to-User Similarity** This approach assigns the average similarity of a location route and all user routes as the similarity score of this location route to the user. Let U and L denote the sets of all user routes and location routes respectively and $card(U)$ denote the number of user routes. The average similarity of a route to a user S_{avg} is presented in Formula 7.4.

$$S_{avg}(l, U) = \sum_{u_i \in U} \frac{ED(l, u_i)}{card(U)}; \forall l \in L \tag{7.4}$$

  This approach computes all the pairwise similarities of the location routes and the user routes in order to obtain the top-k recommended location routes for a user.

- **Highest Similarity Pair**

  Instead of averaging the similarity of a location route to all user routes as in the average route-to-user similarity, this approach also computes all the pairwise similarities of location routes and user routes, however, for all location routes, it assigns the highest similarity score of a location route to any of the user routes as the route to user similarity. This means that it is enough for a location route to be highly similar to only one user route to be included in the user recommended routes.

$$S_{max}(l, U) = \max_{u_i \in U}(ED(l, u_i)); \forall l \in L \tag{7.5}$$

- **User Representative Route**

  This approach assigns one route u_{rep} to represent all the user routes as a first step (Formula 7.6). It then computes the similarity of location routes to this user representative route as shown in Formula 7.7.

$$u_{rep} = \frac{\sum_{u_i \in U} U_i}{card(U)} \tag{7.6}$$

$$S_{rep}(l, U) = ED(l, u_{rep}); \forall l \in L \tag{7.7}$$

  This approach has a computational advantage over the previous two approaches because it does not require the computation of all pairwise location route to user route similarities.

Following any of the approaches proposed, the recommender system selects the top-k location routes and returns them as an output.

7.4.3 Evaluation of Route Recommender System

For the evaluation of the proposed system, a testing dataset is built based on user and location routes from MapMyFitness and annotated preferences from participating active runners.

7.4.3.1 Dataset and Metric

A group of 14 active users of MapMyFitness with various locations and an average of 177 routes per user are considered for the evaluation. Figure 7.13 shows the number of logged routes ran by each of these users.

To resemble a real-life situation where users from different locations move to a new location with little or no information about its running routes, a set of 100 routes were selected from the city of London as location routes. The 14 users were required to annotate their ratings on Likert scales for a total of 30 routes each through a webpage which presents them a map for every running route along with additional data about the route as shown in Figure 7.14. The user ratings form the ground truth to which the system-produced recommendations are compared. Normalized Discounted Cumulative Gain (nDCG) is then used to measure the performance of the recommendation system based on the graded relevance of the recommended routes [47].

7.4.3.2 Overall Recommendation Evaluation

After tuning the system to the experimentally determined optimal contribution ratio of ascents to descents and using the best approach to compute route-to-user similarity and as shown in Section 7.4.3.3 and Section 7.4.3.4 respectively,

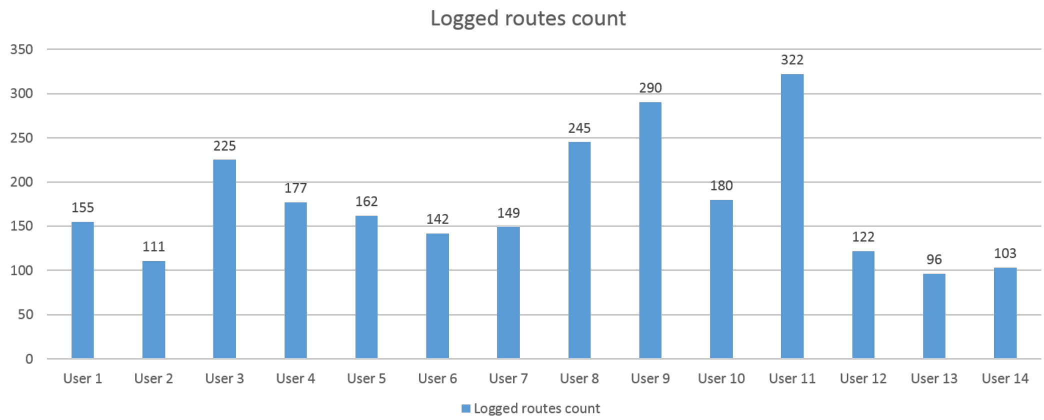

Figure 7.13: Number of logged routes per user.

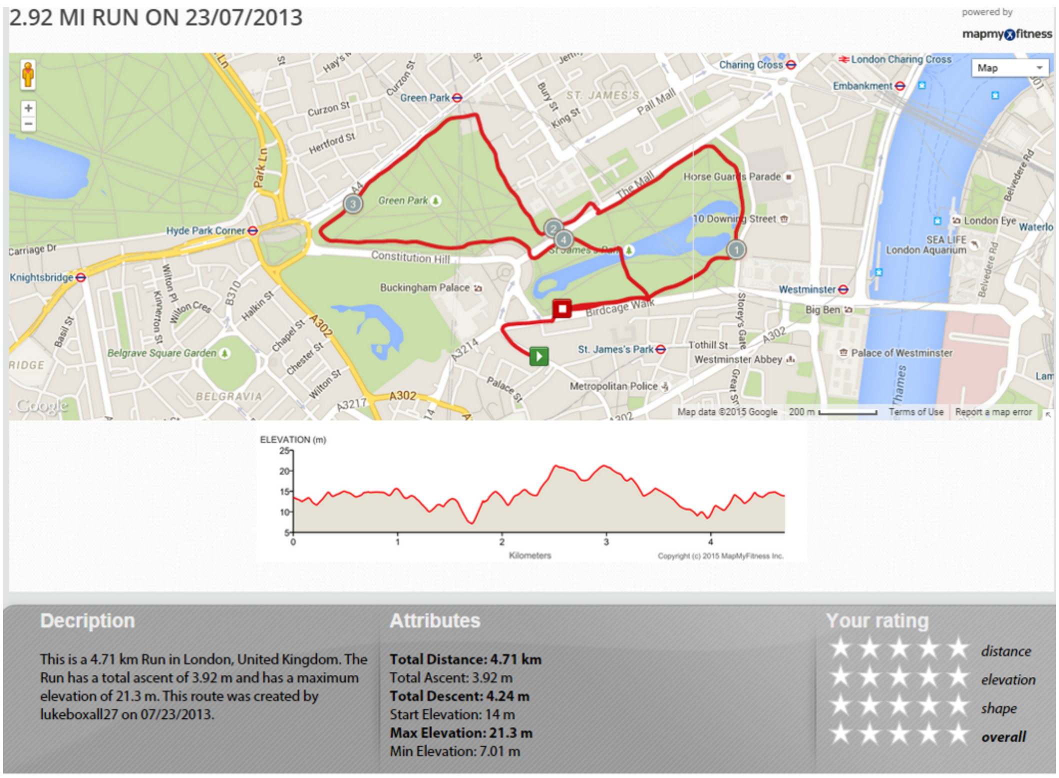

Figure 7.14: A snapshot of a route rating's webpage.

the nDCG scores of the top 5 recommended routes for each user are presented in Figure 7.15. The average nDCG score attained in this final configuration is 84.13%. This indicates the quality of the recommendations provided by the system in terms of both the routes selected and the order in which they are recommended.

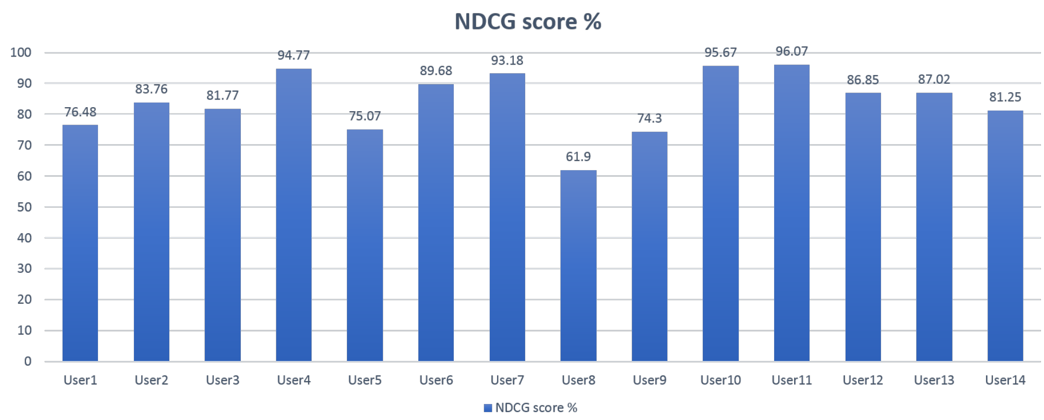

FIGURE 7.15: nDCG-5 scores for system recommendations per user.

The performance of the system varies along with the variation of the number of returned recommendations by the system. To study the effect of this variation, nDCG scores of one and up to 30 recommendations per user are computed. The resulting average nDCG scores per number of recommendations are presented in Figure 7.16. For a total of 30 rated routes per user, a recommendation of up to five routes seems reasonable as it is not probable to have much more highly similar routes to the user's routes among the 30 rated location routes.

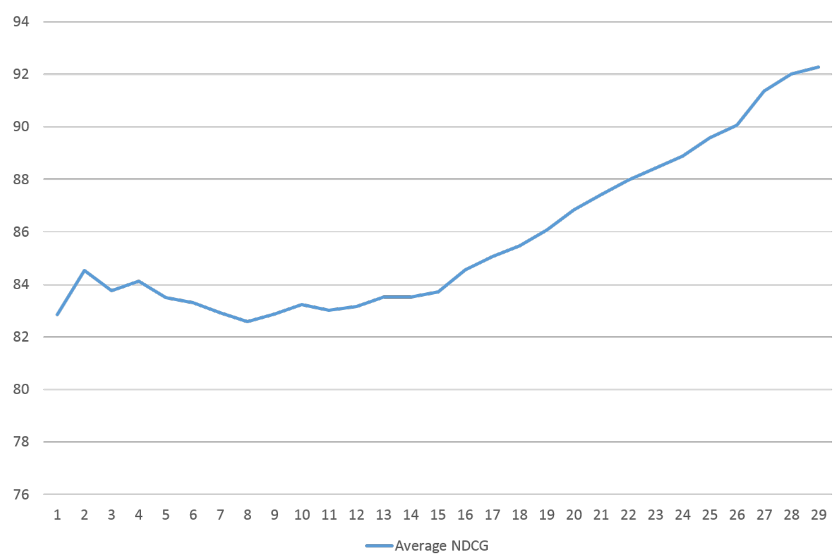

FIGURE 7.16: Average nDCG scores for different number of returned recommendations.

7.4.3.3 Optimal Ascents-to-Descents Ratio

As previously indicated in Section 7.4.2.2, the contribution of ascents (resp. descents) to the computation of *elevation* feature is determined by α (resp. $(1-\alpha)$) in Formula 7.2. Figure 7.17 exhibits the effect of varying the elevation weight in steps of 0.05 between ascents and descents. The optimal value for α is $0.75 - 0.8$ as shown in the figure where the nDCG hits a maximum of 84.13% indicating that ascents are three to four times as important as descents for determining runners preferences on average.

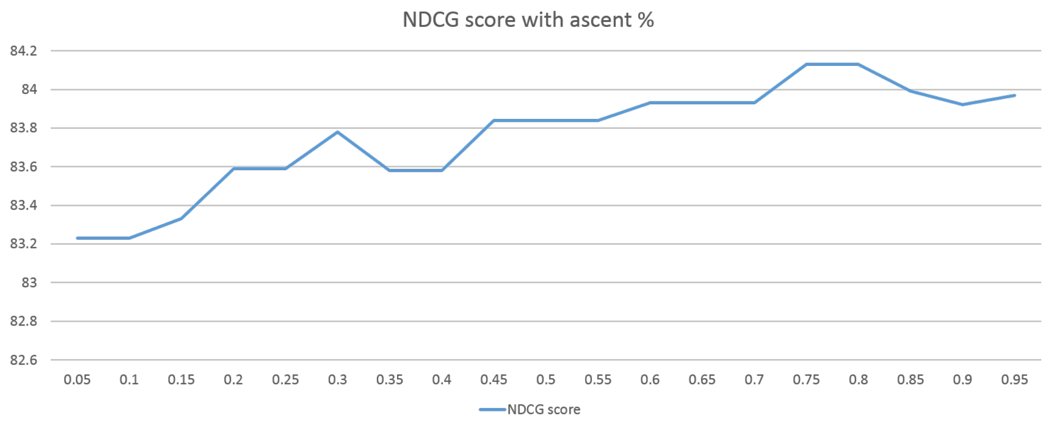

Figure 7.17: Effect of varying the weight of ascents-to-descents (α) on the overall quality of recommendations.

7.4.3.4 Route-to-User Similarity Evaluation

Three approaches for aggregating the similarity of each location route with respect to the user as a whole have been introduced in Section 7.4.2.2. The performance of these different approaches is shown in Figure 7.18. The *Representative Route* approach produces the highest average nDCG score among the 14 participants, with a score of 84.13%. The *Average Route-to-User Similarity* approach came in second place with a score of 81.93%. Finally, the *Highest Similarity Pair* method produced the lowest result of 75.14%. Notably, and as indicated previously, the *Representative Route* approach has the best performance among the three considered approaches as it does not require computing all the pair-wise similarities among all the user routes and location routes.

7.5 Summary

In this chapter, three real-life use cases relevant to consumer sensing devices and harnessing sensor data are presented.

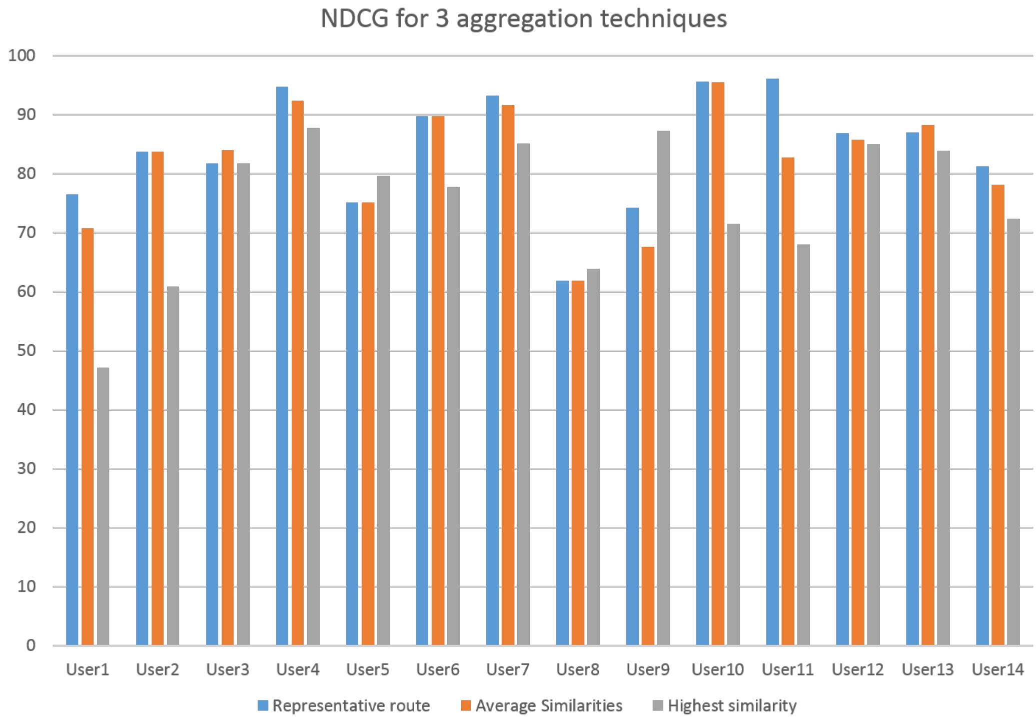

FIGURE 7.18: Comparison of the effect of the different route-to-user similarity methods.

In the first use case, the various aspects of the three market-leading fitness trackers, namely Fitbit One, Jawbone Up and Nike+ Fuelband, are evaluated. The evaluation was achieved by following the aspect-based sentiment analysis approach detailed in Chapter 4. For the main aspects of these fitness trackers, Fitbit One had the best assessment for the "steps counting" aspect. For the "distance traveled" aspect, all three devices had a similar assessment with a slight advantage for the Nike+ Fuelband. As for the "calories burned" aspect, Jawbone Up had the best assessment among the devices with a slight advantage on Fitbit One and a significant advantage over Nike+ Fuelband. Many other insights learned from the evaluation of these devices were also included.

In the second use case, a street quality assessment application that aims at detecting street imperfections is presented. This application demonstrates the ability for prototyping a semantic sensor data analysis system using smartphone sensors for data collection and the big data encoding scheme presented in Chapter 6 for the semantification of the sensor data based on an ontology generated automatically to describe the sensor deployment. A smartphone was attached to a public bus for data collection and high-level access to semantic data was used to detect street imperfections in the city of Kaiserslautern.

In the third application, openly available running route data was used to provide filtering of routes based on performance features, nature of the route and visual features. A recommender system was then built using these features to provide personalized recommendations for runners based on their preferences. This application aims at enhancing the experience of runners especially in locations they are not familiar with.

Conclusions and Future Work 8

In the final chapter of this thesis, Section 8.1 concludes the work and highlights the main lessons learned. It also discusses the limitations and shortcomings of the thesis. Section 8.2 provides an outlook of future work that can build on the various contributions of this thesis.

8.1 Conclusions

This thesis addresses two major challenges that resulted from the increased adoption of sensors in both consumer markets and commercial technical systems.

The first major challenge addressed in this thesis is the need for providing a robust, dynamic and reliable means of evaluating consumer-oriented sensor devices. Achieving this goal helps overcome the limitations of physical product testing which is not reliable given the fast pace of product releases, growing feature sets of the products and the continuous updates that can impact their performance. This thesis introduces an end-to-end pipeline that applies sentiment analysis techniques to opinion rich customer reviews of market sensing devices and provides potential consumers with an easy to grasp summary of other customers' experiences with the products at the aspect level. Thus, facilitating educated and personalized purchases of market sensing devices.

The proposed pipeline features a combined lexicon of a manually crafted sentiment dictionary and SentiWordNet, a part-of-speech-aware probabilistic lexicon. The combined lexicon assigns sentiment scores based on a set of rules for all the cases where the manually crafted and probabilistic lexicons report opposite polarities for a single word. The sentiment analysis pipeline also features an NLP-based approach for detecting the negation scope that replaces the negation window approach used in related work. This is particularly significant because negations are the most occurring sentiment shifters in text. Sentiment extraction using the proposed approach outperforms state-of-the-art supervised SVM and Naive Bayes classifiers trained on unigrams. It is worth mentioning, that the evaluation was done on sentencing containing only one aspect as the supervised approaches assign the sentiment score for the sentence as whole, whereas the proposed lexicon-based approach handles the cases where multiple aspects are mentioned within a single sentence.

The second major challenge addressed in this thesis is the need for handling vast amounts of sensor data by scaling up to meet storage and processing capabilities, along with enhancing the usability of the collected sensor data. This thesis introduces a novel approach that couples semantic techniques with big data technologies to meet the aforementioned requirements. The Semantic Sensor Network (SSN) ontology is used for the semantic modeling of sensor data and Apache Spark big data engine is used for the processing.

The proposed approach harnesses the separation between TBox statements (ontology statements), and ABox statements (actual sensor data) in an ontological knowledge-base to create an encoding scheme that distinguishes how Spark handles the two statement types. This scheme preserves the semantic nature of the data on a conceptual level, thus, preserving semantic access to data records. However, the actual storage of sensor data in Spark RDDs (Resilient Data Records) resembles column-oriented database storage in which sensor data items of the same type are grouped together opposite to simply storing RDF triples representing the data randomly in Spark RDDs. To my knowledge, this approach is the first to couple ontological modeling of sensor data with big data techniques. It is noteworthy that the proposed encoding scheme can be applied to other in-memory big data engines such as Apache Flink, however, the choice of Spark was due to it being more evolved at the time of this study.

This thesis also features contributions on a practical and application levels.

One practical contribution of this thesis is providing a prototyping approach that enables creating a semantic sensor network for monitoring an object of interest by simply physically attaching the smartphone to the object. This approach automatically generates an SSN-based ontology model for the setup which enables the semantification of the recorded sensor data. It is worth mentioning that the ontological model can be preserved even if the smartphone is replaced with permanent sensors past the working prototype. This approach is limited to the standard sensors equipped in smartphones.

As an application contribution, this thesis presents a comprehensive aspect-level evaluation of market-leading fitness trackers namely Fitbit One, Nike+ Fuelband and Jawbone Up. This represents a use-case of the aforementioned sentiment analysis pipeline proposed for evaluating market sensor devices. Amazon reviews for the three devices are used for the evaluation. The results for this evaluation comply with the physical testing of the performance aspects of these devices reported in two independent studies [38] [77]. Results of non-performance aspects evaluation are not verified due to the unavailability of physical testing results for these aspects.

In another application contribution, street quality in the German city of Kaiserslautern was assessed as a use-case for the smartphone-based semantic sensor network prototyping approach mentioned above. In this application, over 14 million sensor data records are recorded by a smartphone attached to a public transport bus for a duration of 8 days. Since this approach ensures the semantification of the generated sensor data, high-level semantic access to the data using rules for variation of acceleration resulted in an outcome of location of potential potholes. Spatial clustering is then applied to report the frequently detected locations only. The findings of this application were not evaluated due to the impracticality of the evaluation. A smartphone app was developed so that a user tags locations of potholes in the city. However, since the decision whether a specific street-imperfection should be tagged is subjective, the collected data was inconsistent and hence could not be used as ground truth for the evaluation.

In a final application contribution, a preference-based filtering approach is used to create a content-based running-route recommender using publicly available workout sensor data. From an application perspective, the running route recommender proposed is novel as it considers three categories of features, namely, performance, visual, and nature-of-route features. The considered visual features represent the proximity of the route to parks and water sources. This feature is computed through analyzing the color-coding of the route surroundings retrieved using a mapping API. Although this approach has a high accuracy of over 95%, its computation represents a bottleneck because it requires external API calls. Among the features representing the nature of routes is tagging the route as a standard running track. This is important because many users prefer running in a closed, altitude-neutral routes. The track-detection approach uses a Bayesian classifier trained on a set of specific features that reflect the route's proximity to its convex hull, utilizing the convex shape of a running track and its standardized dimensions. The classification was tested on a manually annotated set of 100 routes and achieved perfect accuracy. The route recommender built on top of this filtering approach is evaluated using data from highly-active runners. The recommendations generated by the proposed recommender achieved a normalized discounted cumulative gain (nDCG) of 84% which signifies that the recommender not only provided similar routes to the user's manually provided preference, but also provided a similar order among the returned routes.

8.2 Future Work

In this section, I present potential research directions that follow the work accomplished in this thesis.

- **Sentiment analysis pipeline**

  The many challenges of sentiment analysis include handling additional challenges such as sarcasm which is frequently used in informal subjective writing. Moreover, handling special characters such as emojis or emoticons can enhance the sentiment extraction when available although such characters are not frequently used in product reviews. Additionally, presenting more insights and perspectives in the summarizer outcome can be useful to consumers. This can include insights based on regional preferences and different periods of the product lifetime.

- **Management of semantic sensor data**

  The introduced sensor data management approach does not impose any practical limitation on querying semantic sensor data stored according to the presented scheme. However, SPARQL has become the query language of choice by many users when it comes to semantic data. This fact makes building a query transformation system that transforms the widely familiar SPARQL queries to optimized query plans executed using the proposed architecture an interesting future direction.

  Application wise, extending the generated bus network ontology by integrating public transportation lines and plans in addition to other semantic data sources such as weather and events data can significantly extend the applications of such a knowledge base.

- **Running route recommender**

  The performance-related route data available for the presented application relied mainly on location, speed and altitude data. Considering the fact that modern fitness trackers include additional sensor data such as heart-rate and skin temperature, expanding the scope of the considered features to account for such data can lead to better recommendations to users.

Bibliography

[1] Measurement units ontology (muo). `http://idi.fundacionctic.org/muo/`. Accessed: 2016-02-29.

[2] Gabriel Aguilera, José Luis Gálan, José Carlos Campos, and Pedro Rodríguez. An accelerated-time simulation for traffic flow in a smart city. 2013.

[3] Sasikanth Avancha, Chintan Patel, and Anupam Joshi. Ontology-driven adaptive sensor networks. In *MobiQuitous*, volume 4, pages 194–202, 2004.

[4] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec*, volume 10, pages 2200–2204, 2010.

[5] Seyed-Ali Bahrainian and Andreas Dengel. Sentiment analysis and summarization of twitter data. In *Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on*, pages 227–234. IEEE, 2013.

[6] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22(4):469–483, December 1996.

[7] Payam Barnaghi, Stefan Meissner, Mirko Presser, and Klaus Moessner. Sense and sensâĂŹ ability: Semantic data modelling for sensor networks. In *Conference Proceedings of ICT Mobile Summit 2009*, 2009.

[8] BCCResearch. Wearable computing: Technologies, applications and global markets. Technical report, BCC Research, 2014.

[9] Pushpak Bhattacharyya. Indowordnet. In *In Proc. of LREC-10*. Citeseer, 2010.

[10] A. Bialecki, M. Cafarella, D. Cutting, and O. OâĂŹMalley. Hadoop : a framework for running applications on large clusters built of commodity hardware. *http://lucene.apache.org/hadoop*, 2005.

[11] William Black, Sabri Elkateb, Horacio Rodriguez, Musa Alkhalifa, Piek Vossen, Adam Pease, and Christiane Fellbaum. Introducing the arabic wordnet project. In *Proceedings of the third international WordNet conference*, pages 295–300, 2006.

[12] Sasha Blair-Goldensohn, Kerry Hannan, Ryan McDonald, Tyler Neylon, George A Reis, and Jeff Reynar. Building a sentiment summarizer for local service reviews. In *WWW workshop on NLP in the information explosion era*, volume 14, pages 339–348, 2008.

[13] Erik Boiy and Marie-Francine Moens. A machine learning approach to sentiment analysis in multilingual web texts. *Information retrieval*, 12(5):526–558, 2009.

[14] Dhruba Borthakur. The hadoop distributed file system: Architecture and design. *Hadoop Project Website*, 11(2007):21, 2007.

[15] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.

[16] M Busse, W Schwerdtner, R Siebert, A Doernberg, A Kuntosch, B König, and W Bokelmann. Analysis of animal monitoring technologies in germany from an innovation system perspective. *Agricultural Systems*, 138:55–65, 2015.

[17] Paris Carbone, Asterios Katsifodimos, Stephan Ewen, Volker Markl, Seif Haridi, and Kostas Tzoumas. Apache flink: Stream and batch processing in a single engine. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 36(4), 2015.

[18] Giuseppe Carenini, Jackie Chi Kit Cheung, and Adam Pauls. Multi-document summarization of evaluative text. *Computational Intelligence*, 29(4):545–576, 2013.

[19] Yanyan Chen, M.G.H. Bell, and K. Bogenberger. Reliable pretrip multipath planning and dynamic adaptation for a centralized road navigation system. *Intelligent Transportation Systems, IEEE Transactions on*, 8(1):14–20, March 2007.

[20] Michael Compton, Payam Barnaghi, Luis Bermudez, RaúL GarcíA-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, et al. The ssn ontology of the w3c semantic sensor network incubator group. *Web semantics: science, services and agents on the World Wide Web*, 17:25–32, 2012.

[21] Fog Computing. the internet of things: Extend the cloud to where the things are, 2016.

[22] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[23] Olivier Curé, Hubert Naacke, Mohamed-Amine Baazizi, and Bernd Amann. On the evaluation of rdf distribution algorithms implemented over apache spark. *arXiv preprint*, 2015.

[24] Olivier Curé, Hubert Naacke, Tendry Randriamalala, and Bernd Amann. Litemat: a scalable, cost-efficient inference encoding scheme for large rdf graphs. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 1823–1830. IEEE, 2015.

[25] Kathryn L Dannecker, Nadezhda A Sazonova, Edward L Melanson, Edward S Sazonov, and Raymond C Browning. A comparison of energy expenditure estimation of several physical activity monitors. *Medicine and science in sports and exercise*, 45(11):2105, 2013.

[26] 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)*, volume 35, page 43. Bangkok, Thailand, 2001.

[27] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454. Genoa Italy, 2006.

[28] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[29] Xiaowen Ding, Bing Liu, and Philip S Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*, pages 231–240. ACM, 2008.

[30] D Douglas. Algorithms for the reduction of the number of points required to represent a line or its a caricature. *The Canadian Cartographer*, 10(2):112–122, 1973.

[31] Mohamad Eid, Ramiro Liscano, and Abdulmotaleb El Saddik. A universal ontology for sensor networks data. In *Computational Intelligence for Measurement Systems and Applications, 2007. CIMSA 2007. IEEE International Conference on*, pages 59–62. IEEE, 2007.

[32] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. The pothole patrol: using a mobile sensor network for road surface monitoring. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 29–39. ACM, 2008.

[33] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[34] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A high-coverage lexical resource for opinion mining. *Institute of Information Science and Technologies (ISTI) of the Italian National Research Council (CNR)*, 2006.

[35] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[36] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. In *ACM SIGOPS operating systems review*, volume 37, pages 29–43. ACM, 2003.

[37] Thomas R Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5-6):907–928, 1995.

[38] Fangfang Guo, Yu Li, Mohan S. Kankanhalli, and Michael S. Brown. An evaluation of wearable activity monitoring devices. In *Proceedings of the 1st ACM International Workshop on Personal Data Meets Distributed Multimedia*, PDM '13, pages 31–34, New York, NY, USA, 2013. ACM.

[39] David J Hand and Keming Yu. Idiot's bayesâĂŤnot so stupid after all? *International statistical review*, 69(3):385–398, 2001.

[40] Katherine A Heller and Zoubin Ghahramani. A simple bayesian framework for content-based image retrieval. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2110–2117. IEEE, 2006.

[41] Jana Ariel Hirsch, Peter James, Jamaica R M Robinson, Kyler M Eastman, Kevin Daniel Conley, Kelly R Evenson, and Francine Laden. Using mapmyfitness to place physical activity into neighborhood context. *Frontiers in Public Health*, 2(19), 2014.

[42] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[43] Hassan Issa, Amir Guirguis, Shary Beshara, Stefan Agne, and Andreas Dengel. Preference based filtering and recommendations for running routes. In *Proceedings of the 12th International Conference on Web Information Systems and Technologies*, pages 139–146, 2016.

[44] Hassan Issa, Alaa Shafaee, Stefan Agne, Stephan Baumann, and Andreas Dengel. User-sentiment based evaluation for market fitness trackers - evaluation of fitbit one, jawbone up and nike+ fuelband based on amazon.com customer reviews. In *Proceedings of the International Conference on Information and Communication Technologies for Ageing Well and e-Health - Volume 1: ICT4AgeingWell,*, pages 171–179, 2015.

[45] Hassan Issa, Ludger van Elst, and Andreas Dengel. Using smartphones for prototyping semantic sensor analysis systems. In *Proceedings of the International Workshop on Semantic Big Data*, SBD '16, pages 7:1–7:6, New York, NY, USA, 2016. ACM.

[46] Krzysztof Janowicz and Michael Compton. The stimulus-sensor-observation ontology design pattern and its integration into the semantic sensor network ontology. In *Proceedings of the 3rd International Conference on Semantic Sensor Networks-Volume 668*, pages 64–78. CEUR-WS. org, 2010.

[47] Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *SIGIR*, pages 41–48, 2000.

[48] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160. Association for Computational Linguistics, 2011.

[49] Nitin Jindal and Bing Liu. Mining comparative sentences and relations. In *AAAI*, volume 22, pages 1331–1336, 2006.

[50] Jeong-Hee Kim, Hoon Kwon, Do-Hyeun Kim, Ho-Young Kwak, and Sang-Joon Lee. Building a service-oriented ontology for wireless sensor networks. In *Computer and Information Science, 2008. ICIS 08. Seventh IEEE/ACIS International Conference on*, pages 649–654. IEEE, 2008.

[51] Soo-Min Kim and Eduard Hovy. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1367. Association for Computational Linguistics, 2004.

[52] Rob Kitchin. The real-time city? big data and smart urbanism. *GeoJournal*, 79(1):1–14, 2014.

[53] S. Knoch, A. Chapko, A. Emrich, D. Werth, and P. Loos. A context-aware running route recommender learning from user histories using artificial neural networks. In *Database and Expert Systems Applications (DEXA), 2012 23rd International Workshop on*, pages 106–110, Sept 2012.

[54] Christian Koch and Ioannis Brilakis. Pothole detection in asphalt pavement images. *Advanced Engineering Informatics*, 25(3):507–515, 2011.

[55] Moshe Koppel and Jonathan Schler. The importance of neutral examples for learning sentiment. *Computational Intelligence*, 22(2):100–109, 2006.

[56] Sebastian Land and Simon Fischer. Rapidminer 5-in academic use. *Available at: docs. rapidi. com/files/academic/RapidMiner RapidMinerInAcademicUse en. pdf.(Access: 12 August 2013)*, 2012.

[57] Jay Lee, Hung-An Kao, and Shanhu Yang. Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia Cirp*, 16:3–8, 2014.

[58] Lillian Lee. Measures of distributional similarity. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 25–32. Association for Computational Linguistics, 1999.

[59] Xiao-Li Li, Lei Zhang, Bing Liu, and See-Kiong Ng. Distributional similarity vs. pu learning for entity set expansion. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 359–364. Association for Computational Linguistics, 2010.

[60] Bing Liu. Opinion lexicon: A list of english positive and negative opinion words or sentiment words, 2004.

[61] Bing Liu, Wee Sun Lee, Philip S Yu, and Xiaoli Li. Partially supervised classification of text documents. In *ICML*, volume 2, pages 387–394, 2002.

[62] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[63] John R Mashey. Big data and the next wave of infrastress. Computer Science Division Seminar, University of California, Berkeley, oct 1997.

[64] C Masolo, S Borgo, A Gangemi, N Guarino, A Oltramari, and L Schneider. Dolce: a descriptive ontology for linguistic and cognitive engineering. *WonderWeb Project, Deliverable D*, 17, 2003.

[65] Artis Mednis, Girts Strazdins, Reinholds Zviedris, Georgijs Kanonirs, and Leo Selavo. Real time pothole detection using android smartphones with accelerometers. In *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, pages 1–6. IEEE, 2011.

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

[67] Tamoghna Ojha, Sudip Misra, and Narendra Singh Raghuwanshi. Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges. *Computers and Electronics in Agriculture*, 118:66–84, 2015.

[68] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

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

[70] G. Pang, K. Takahashi, T. Yokota, and H. Takenaga. Adaptive route selection for dynamic route guidance system based on fuzzy-neural approaches. In *Vehicle Navigation and Information Systems Conference, 1995. Proceedings. In conjunction with the Pacific Rim TransTech Conference. 6th International VNIS. 'A Ride into the Future'*, pages 75–82, Jul 1995.

[71] Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 938–947. Association for Computational Linguistics, 2009.

[72] Ana-Maria Popescu and Orena Etzioni. Extracting product features and opinions from reviews. In *Natural language processing and text mining*, pages 9–28. Springer, 2007.

[73] Daniele Quercia, Rossano Schifanella, and Luca Maria Aiello. The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. *CoRR*, abs/1407.1031, 2014.

[74] Stefan Rüping. mysvm-a support vector machine.

[75] David J Russomanno, Cartik Kothari, and Omoju Thomas. Sensor ontologies: from shallow to deep models. In *System Theory, 2005. SSST'05.*

Proceedings of the Thirty-Seventh Southeastern Symposium on, pages 107–112. IEEE, 2005.

[76] W. Sasaki and Y. Takama. Walking route recommender system considering saw criteria. In *Technologies and Applications of Artificial Intelligence (TAAI), 2013 Conference on*, pages 246–251, Dec 2013.

[77] Alaa Shafaee. Evaluating Fitness Trackers through Physical Testing and Assessment of the Public Opinion. The German University in Cairo, 2013. Bachelor Thesis.

[78] Sinnott. Virtues of the haversine. *skytel*, 68:158, dec 1984.

[79] Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. Parsing with compositional vector grammars. In *ACL (1)*, pages 455–465, 2013.

[80] Philip J Stone, Dexter C Dunphy, and Marshall S Smith. The general inquirer: A computer approach to content analysis. 1966.

[81] Maria M. Suarez-Alvarez, Duc-Truong Pham, Mikhail Y. Prostov, and Yuriy I. Prostov. Statistical approach to normalization of feature vectors and clustering of mixed datasets. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 468(2145):2630–2651, 2012.

[82] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *Journal of the Association for Information Science and Technology*, 63(1):163–173, 2012.

[83] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

[84] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

[85] Vinod Kumar Vavilapalli, Arun C Murthy, Chris Douglas, Sharad Agarwal, Mahadev Konar, Robert Evans, Thomas Graves, Jason Lowe, Hitesh Shah, Siddharth Seth, et al. Apache hadoop yarn: Yet another resource negotiator. In *Proceedings of the 4th annual Symposium on Cloud Computing*, page 5. ACM, 2013.

[86] Piek Vossen. *A multilingual database with lexical semantic networks*. Springer, 1998.

[87] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2):165–210, 2005.

[88] Kenneth J Witt, Jason Stanley, David Smithbauer, Dan Mandl, Vuong Ly, Al Underbrink, and Mike Metheny. Enabling sensor webs by utilizing swamo for autonomous operations. In *8th NASA Earth Science Technology Conference*, pages 263–270, 2008.

[89] Reynold S Xin, Josh Rosen, Matei Zaharia, Michael J Franklin, Scott Shenker, and Ion Stoica. Shark: Sql and rich analytics at scale. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of data*, pages 13–24. ACM, 2013.

[90] Hong Yu and Vasileios Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 129–136. Association for Computational Linguistics, 2003.

[91] X Yu and E Salari. Pavement pothole detection and severity measurement using laser imaging. In *Electro/Information Technology (EIT), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.

[92] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pages 2–2. USENIX Association, 2012.

[93] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. *HotCloud*, 10:10–10, 2010.

[94] Xiaoming Zhang, Yunping Zhao, and Wanming Liu. A method for mapping sensor data to ssn ontology. *International Journal of u-and e-Service, Science and Technology*, 8(9):303–316, 2015.

List of Own Publications

[1] Hassan Issa, Amir Guirguis, Shary Beshara, Stefan Agne, and Andreas Dengel. Preference based filtering and recommendations for running routes. In *Proceedings of the 12th International Conference on Web Information Systems and Technologies*, pages 139–146, 2016.

[2] Hassan Issa, Alaa Shafaee, Stefan Agne, Stephan Baumann, and Andreas Dengel. User-sentiment based evaluation for market fitness trackers - evaluation of fitbit one, jawbone up and nike+ fuelband based on amazon.com customer reviews. In *Proceedings of the International Conference on Information and Communication Technologies for Ageing Well and e-Health - Volume 1: ICT4AgeingWell,*, pages 171–179, 2015.

[3] Hassan Issa, Ludger van Elst, and Andreas Dengel. Using smartphones for prototyping semantic sensor analysis systems. In *Proceedings of the International Workshop on Semantic Big Data*, SBD '16, pages 7:1–7:6, New York, NY, USA, 2016. ACM.

[4] Alaa Shafaee, Hassan Issa, Stefan Agne, Stephan Baumann, and Andreas Dengel. Aspect-based sentiment analysis of amazon reviews for fitness tracking devices. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 50–61. Springer, 2014.

Hassan Issa

hassan.issa@dfki.de
http://www.hassanissa.com

EDUCATION

PhD Computer Science 2013-2018
University of Kaiserslautern

"Assessment, Semantification and Applications of Sensor Data"
Thesis advisor: Prof. Dr. Andreas Dengel (DFKI / TU Kaiserslautern)

MSc Computer Science 2009-2012
Saarland University

"Semantic-Aware Document Similarity Search"
Thesis advisor: Prof. Dr. Ralf Schenckel (MPI-INF / Saarland University)

BSc Computer Science 2006-2009
Lebanese University

Bachelor Project: E-Voting System in Lebanon

WORK EXPERIENCE

Researcher 2018-
German Research Center for Artificial Intelligence (DFKI)

Research Assistant 2013-2018
German Research Center for Artificial Intelligence (DFKI)

Research Assistant 2011
Max Planck Institute for Informatics (MPI-INF)

DISTINCTIONS

- 2009-2011 Graduate School Scholarship - Saarland University
- 2009 Honor list - ranked 2^{nd} in Computer Science for year 2009 - Lebanese University

LANGUAGES

- Arabic (Native)
- English (Fluent)
- German (Very Good, DSH-2)

GENERAL INTERESTS

Sports, photography, technology, programming, reading and online learning