NTNU – Trondheim
Norwegian University of
Science and Technology

Enlightening the Bus Oracle

Increasing the Utility of a Bus Information System Based on Natural Language Processing

Erling W Eeg-Henriksen

Master of Science in Computer Science
Submission date: June 2015
Supervisor: Rune Sætre, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Abstract

BusTUC is a Natural Language Processing (NLP)-based system for bus information in Trondheim, Norway. Its service is used in numerous applications, and new applications are still being made. However, not all of these see much use, and users of the system are often only included in the evaluation of the result. Because of this, some BusTUC projects end up having little effect on Trondheim's bus passengers. This thesis details an approach to improving BusTUC that ensures that improvements both reach the users and are perceived as useful by them.

A review of the relevant literature was conducted, both establishing the necessary background for BusTUC and exploring recent developments in BusTUC and other NLP systems. Then, a survey was created to investigate which additional functionality there is demand for in BusTUC. With an analysis of earlier surveys' biases, the new survey had the soundness of its results increased. Focusing on improvements to the core text-to-text system to ensure that all applications of BusTUC would benefit from them, the thesis identifies several good candidates.

The results of this research was analyzed and discussed, leading to the implementation of three of the improvements identified as highly useful. Analyzing the resulting expansion of the system's functionality, the thesis concludes that the approach has merit and does indeed help ensure that the development is beneficial to bus passengers.

Sammendrag

BussTUC er et system for buss-informasjon i Trondheim som baserer seg på prosessering av naturlig språk. Tjenesten den utgjør brukes i mange applikasjoner, og flere applikasjoner lages fortsatt. Det er imidlertid ikke alle av disse som brukes i stor grad, og brukere av systemet blir ofte kun inkludert for evaluering av sluttresultatet. På grunn av dette ender noen BussTUC-prosjekter med liten effekt for Trondheims buss-reisende. Denne masteroppgaven beskriver en fremgangsmåte for å forbedre BussTUC som tilser at forbedringene når frem til brukerne og blir ansett som praktiske.

En gjennomgang av den relevante vitenskapelige litteraturen ble utført. Dette etablerte den nødvendige bakgrunnen for BussTUC og utforsket nylig utvikling i BussTUC og andre språk-prosesserende systemer. Deretter ble en spørreundersøkelse opprettet for å undersøke hvilken tilleggs-funksjonalitet for BussTUC det finnes etterspørsel for. Gjennom analyse av svakheter i tidligere spørreundersøkelser, ble validiteten til spørreundersøkelsens resultater forsterket. Med fokus på forbedringer av det grunnleggende tekst-til-tekst-systemet, for å sørge for at forbedringene ville gagne alle BussTUC-brukende applikasjoner, ble flere gode kandidater identifisert.

Resultatene av forskningen ble analysert og diskutert, og ledet til implementeringen av tre av forbedringene som var blitt identifisert som svært praktiske. Etter analyse av den resulterende utvidelsen av systemets funksjonalitet konkluderer masteroppgaven med at fremgangsmåten er fordelaktig, ettersom det ble demonstrert at den sørger for at utviklingen kommer brukerne til gode.

Preface

This thesis is the result of work done for *TDT4900 – Computer Science, Master's Thesis* at the Department of Computer and Information Science (IDI), Faculty of Information Technology, Mathematics and Electrical Engineering (IME) of the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway. A pre-study for the thesis was performed during the autumn semester of 2014, while the rest took place during the spring semester of 2015. Rune Sætre was the supervisor of the project.

Acknowledgements
I would like to thank my supervisor, Rune Sætre, and Björn Gambäck for their feedback and support. Thanks also go to Sofia Nascimento Bakke and Ole Kristian Nakken, who worked on another BusTUC-related project alongside me, for our discussions and their contributions to BusTUC as a whole. Finally, I wish to thank my father for his proofreading assistance, and the rest of my family and friends for their encouragement.

Erling Wishman Eeg-Henriksen
Trondheim, June 11, 2015

Contents

List of Figures

List of Tables

Glossary

ASR Automatic Speech Recognition. 18

AtB Trondheim's main bus company.. 1, 15, 17, 20, 42, 44, 46, 47, 54, 59

BusTUC is a bus travel information system for Trondheim, that answers natural language questions in natural language.. i, iv, 1–4, 7, 10–20, 22, 23, 25–27, 33, 34, 36–44, 46, 48–51, 53–60, 69, 70

CBR Case-based Reasoning. 18

CFG Context-Free Grammar. 10

CWA Closed World Assumption. 12

DCG Definite Clause Grammar. 9, 10

GPS Global Positioning System. 17, 25, 26, 39, 42, 43, 70

IDI Department of Computer and Information Science. iv, 16, 17

IME Faculty of Information Technology, Mathematics and Electrical Engineering. iv

NLP Natural Language Processing. i, 2–4, 7, 9, 21, 25, 41, 44, 57

NTNU the Norwegian University of Science and Technology. iv, 4, 13, 18

SMS Short Message Service. 17, 45, 46

TABuss Tore Amble Buss. 17, 18, 25

TQL TUC Query Language. 12, 13, 16, 46, 48, 69

TTS Text-To-Speech. 18

TUC The Understanding Computer. 13, 46, 48, 59, 69

XG Extraposition Grammar. 10

Chapter 1

Introduction

This report describes work done on a natural language processing system for bus-related queries. The goal is to investigate how to best extend the system's current functionality in order to increase its usefulness, and to successfully implement and study improvements based on this. The system to be improved is BusTUC, a system allowing users to send natural language queries for information about bus routes in Trondheim, Norway.

1.1 Motivation

For many people, public transportation has become an essential part of daily life. The accessibility of information about the available public transportation is therefore very important to the public and companies offering such services. Considering the often constant presence of mobile devices, and the availability of Internet connections, letting such information exist solely in the form of brochures and posters on public transportation stations is insufficient. Because of this, most companies offering such transportation also offer information through both websites and cell phone applications (apps). In addition to these, privately developed applications with similar functionality often spring up as well.

This is also the case in Trondheim, Norway. The bus company AtB runs the buses and offers information in several forms. One of the services offered by AtB, *Bussorakelet* (*the Bus oracle*) is powered by BusTUC's ability to understand questions and give answers in natural language. Numerous other applications also exist, such as the Android applications *Bartebuss*[1], *Busstider*[2] *TABuss*[3], *MultiBRIS*[4], *AtBuss*[5] , several of which also include BusTUC as part of their functionality.

New applications like these are still appearing as the result of student projects related to BusTUC, but some of these end up being used very little, as they compete against established, popular applications like *Bartebuss*. It is of obvious importance that features and improvements actually reach the bus travelers. It might therefore be beneficial to focus more on the improvement of the core BusTUC service itself, as improvements to it would immediately be available to users, no matter which BusTUC-utilizing application they prefer.

The development is also often driven by what the developer finds useful, while users are only heard when assessing the end result. Involving bus travelers more in the shaping of BusTUC could ensure that the improvements made are ones that are actually desired and will be used.

1.2 Autumn project

A project leading up to this thesis was completed in the autumn of 2015, and will be referred to primarily as the "pre-study". This project reviewed the background literature for the BusTUC system and investigated the popularity of various potential improvements. The review of background literature forms much of sections 2.1 and 2.2.1. However, there were several notable weaknesses in the survey used to assess the popularity, possibly leading to somewhat biased data. The steps taken to deal with this are shown in chapter 3.

1.3 Goals and research questions

The task given was to expand the functionality of the Natural Language Processing (NLP)-based bus information system BusTUC, to include support for additional information deemed useful by bus travelers. It was specified that the pre-study introduced in section 1.2 should be used as a starting point, and that at least some of the potential improvements it presented should be implemented. These are included in the assessment of improvements described in chapter 3. Dialog support was specified as an additional option.

[1]Bartebuss, a bus information application for Trondheim. Available for both Android and iOS devices. On Google Play:
https://play.google.com/store/apps/details?id=com.runemartin.bartebuss

[2]https://play.google.com/store/apps/details?id=net.a2bsoft.buss

[3]https://play.google.com/store/apps/details?id=test.BusTUC

[4]https://play.google.com/store/apps/details?id=com.retro.MultiBRIS (prototype)

[5]https://play.google.com/store/apps/details?id=net.fosstveit.atbuss

Based on this task, two goals were specified:

G1 *Discover how an NLP-based information system like BusTUC can be made more useful*

The BusTUC system has numerous areas of improvement. In order to improve the service BusTUC provides its users, the improvements desired by users and potential users must be assessed. The structure of BusTUC must also be understood, as well as the concepts, tools and methods upon which it has been built. The existing literature on the subject should also be studied, to gain an overview of alternative methods used for similar systems.

G2 *Extend the BusTUC system to support additional useful information*

Based on what functionality users deem beneficial and the way in which BusTUC and associated technology works, BusTUC's functionality should be extended to make the system more useful.

What the research aims to answer has also been condensed into two points, as shown with the research questions RQ1 and RQ2 below.

RQ1 *What additional information is it useful for NLP-based transportation query systems like BusTUC to support?*

Explore what information it would be beneficial to add support for. Examples could be supporting queries that specify time by a variable (such as opening hours) or possibly even queries about local train routes. As the usefulness of such information is decided by the users, it would be natural to ask for their opinions.

RQ2 *Would BusTUC's usefulness to its users best be increased through a survey-first approach to expanding its functionality?*

This question has two sides to it: Firstly (RQ2-1), analysis of expansions of the BusTUC system is best done in the context of the established literature on both BusTUC and other, relevant systems. Secondly (RQ2-2), the survey-first approach for BusTUC must be used and analyzed in order for this question to be answered.

1.4 Research methods

The research is briefly outlined in this section. It can be divided into the following phases:

1. Literature review

   A literature review was conducted, resulting in an overview of the established literature on BusTUC, its underlying technologies, structure and applications, as well as other work done on NLP transportation information systems.

   The following electronic libraries were used to find the relevant literature: DIVA[6], Google Scholar[7], IEEE Xplore[8], and Scopus.[9] Additionally, some of the BusTUC papers were obtained from the Norwegian University of Science and Technology (NTNU) upon request.

   Search key words Used when researching using the libraries above.

   - Public transportation
   - Information system
   - Route planner
   - Natural language
   - Natural language processing
   - Computational semantics

2. Assessment of possible improvements

   Potential improvements to BusTUC's functionality were assessed to determine which could most effectively increase the system's service to its users. A survey was conducted and analyzed for this.

3. Implementation and analysis

   This includes the implementation of improvements based on the survey's results, and an analysis and discussion of their impact on the system.

   Finally, this leads to a conclusion of the research. Suggestions for future work are also provided.

During all stages, Microsoft OneNote and the BusTUC wiki page were used for planning and keeping track of tasks.

[6]http://www.diva-portal.org/ (as of June 11, 2015)
[7]https://scholar.google.com (as of June 11, 2015)
[8]http://ieeexplore.ieee.org (as of June 11, 2015)
[9]http://www.scopus.com (as of June 11, 2015)

1.5 Thesis structure

Chapter 2 consists of the literature review.

Chapter 3 describes the creation of the survey, based on an analysis of the pre-study's biases.

Chapter 4 contains the results of the survey, along with analysis and discussion.

Chapter 5 details the implementation.

Chapter 6 presents the results of the implementation.

Chapter 7 provides an analysis and discussion of the results, their significance and the research as a whole.

Chapter 8 concludes the research with answers to the research questions and lists suggestions for future work.

Chapter 2

Methods and related work

This chapter delves into the relevant literature on BusTUC, as well as other related systems, technologies and the underlying Natural Language Processing (NLP) methods and concepts. Section 2.1 contains the latter, while section 2.2.1 focuses on BusTUC and section 2.3 presents other work on similar systems.

2.1 NLP concepts and methods

This section explores some areas of NLP that are relevant and important to understand when studying BusTUC and similar systems.

2.1.1 Computational Semantics

"Computational semantics" is a field of research dealing with the construction and use of formal models describing the meaning of natural language phrases. An overview of some relevant concepts and procedures in this field of study is given as an introduction to the procedure of generating semantic representations usable by machines to understand statements (within their area of "expertise").

Blackburn and Bos [2003] give an introduction to representation of natural language, as well as inference on such representations. They divide the field in two by asking how one can:

1. automate the creation of semantic representations of natural language

2. automate inference with logical expressions.

Using first-order logic formulas in our models of natural language helps to ensure that the resulting models are both understandable for humans and relatively easily translated into code useful for NLP systems. In the creation of this type of model, one will typically define a *vocabulary*, which clearly states what relationships and constants are included in a model.

$$\{ \; (\mathrm{BUS66}, 0), \\ (\mathrm{SAMFUNDET}, 0), \\ (\mathrm{BUS}, 1), \\ (\mathrm{STATION}, 1), \\ (\mathrm{STOPSAT}, 2) \; \}$$

In the example vocabulary above, `STOPSAT` is a relationship taking two arguments, for example `STOPSAT(BUS66,SAMFUNDET)` ("Bus 66" stops at "Samfundet"); while `BUS66` and `SAMFUNDET` are constants, shown by the fact that they take no (`0`) arguments. `BUS` and `STATION`, being of arity `1`, can here be said to be properties, allowing us to describe single entities as being a bus or a station – for example `BUS(BUS66)` ("Bus 66" is a bus)[1].

The above is a basic introduction to how models are defined. Below, a model is described using the vocabulary above, to show that "Bus 66" is a bus and stops at "Samfundet", which is a station. For this model, let the domain D of what the model describes be the entities d_1, d_2. These correspond to the constants in the vocabulary. F is a function which specifies the semantic values of each element of D; in other words, what d_1 and d_2 actually *mean* (referred to as an *interpretation function* in [Blackburn and Bos, 2003, ch. 1]).

$$F(\texttt{BUS66}) = d_1 \\ F(\texttt{SAMFUNDET}) = d_2 \\ F(\texttt{BUS}) = d_1 \\ F(\texttt{STATION}) = d_2 \\ F(\texttt{STOPSAT}) = d_1, d_2$$

With vocabularies and models like the examples above, a "world" is described in a way that allows for a relatively easy transition into first-order logic. Adding rules and further complexity to the model of the world, one will lay the groundwork for first-order inference.

By following these procedures, one can design software capable of reasoning within its domain (for example, bus travel in Trondheim) when given statements or questions in first-order form. Sentences in natural language can be given this form – much like the transformation from meaning to model, the first steps of which were exemplified above.

[1] Note that one could alternatively let `BUS` and `STATION` be constants, and use a relation `(IS_A,2)` to describe "Bus 66" being a bus and "Samfundet" being a station.

2.1.2 Grammars

A *grammar* is a collection of rules for a given language, be it a natural language or not.[2] With a grammar it is possible to check whether given sentences are valid within the language to which the grammar belongs. Knowing this, the importance of grammars for NLP systems becomes apparent.

Definite clause grammars

A Definite Clause Grammar (DCG), as described in [Pereira and Warren, 1980], is a formalism in which grammars are expressed with first-order predicate logic, a method of language representation outlined in section 2.1.1 and given a concrete example in Listing 2.1. DCGs are a special case of Colmerauer's *Metamorphosis grammars* [Colmerauer, 1978]. DCGs are descriptions of language, but being in first-order logic they can be written as Prolog code, and thus in themselves become usable parts in language computation. For example, the final part of Listing 2.1 shows the logic for "When does the bus leave Dragvoll?" as a request for an `A` which is a `time` corresponding to a `bus B` leaving the `station` "Dragvoll".

Listing 2.1: Output tree and logic for "When does the bus leave Dragvoll?"

```
                     whenq
      ----------------|----------------
     /          |                      \
    lit         pp                     ynq
     |        ---|--           -----------|----------
     |       /      \         /                      \
   when    < >    np_head    lit                     stm
            |      --|-       |            ---------|---------
            |     /     \     |           /                   \
           prep  < >    < >  does     np_head              do_phrase
                  |      |             ----|---            -----|-----
                  |      |            /        \          /           \
                 the    noun    determiner0   noun    vp_kernel       < >
                                   -|-         |       ---|---         |
                                  /   \        |      /       \        |
                                lit   < >     bus   verb     name   adverbial1
                                 |     |             |         |
                                 |     |             |         |
                                the   art          leave   dragvoll

[which(A):::(dragvoll isa station,B isa bus,A isa time,
             dob/leave/B/dragvoll/C,srel/in/time/A/C,
             event/real/C)]
```

[2] For example, a programming language

Context-free grammars

A grammar works with an *alphabet* containing *terminal* and *non-terminal* symbols. These are then used to represent words and phrases of the language, respectively. In Context-Free Grammar (CFG)s, each rule is defined as a non-terminal equal to a certain sequence of symbols (both terminal and non-terminal), thus defining a valid form of non-terminals. Listing 2.1 gives an example of a sentence analyzed in the context of a CFG, as such analysis produces a *parse tree* which visualizes the internal hierarchy of the sentence's parts.

Rules of CFGs can be expressed in logic, creating *definite clauses* (often called *Horn clauses*). DCG is the result of a generalization of the process of creating definite clauses from CFGs. A collection of definite clauses of a CFG can form a program in Prolog that efficiently parses sentences, creating a parse tree from the top and down. With DCG, non-terminals may be compound terms, for example a noun phrase `np( X, S)` where `S` is a sentence. Furthermore, in the definition of a rule, one may also include procedure calls that create additional conditions for the rule to be valid. Implementing this in Prolog, a grammar can be made that when used to analyze a sentence, classifies and creates a hierarchy for the sentence's parts. The result is easily visualized. Such analysis allows a program to decide whether a sentence makes sense, and if it does, use the result of the analysis to accomplish its goal (for example providing information corresponding to an input question).

Extraposition grammars

The BusTUC system's grammar, *Consensical Grammar*, is a version of the Extraposition Grammar (XG) [Pereira, 1981], which again is a generalization of the Definite Clause Grammar (DCG) described in section 2.1.2. With the background provided by that section, this one gives an introduction to XGs.

In natural language, one often encounters sentences containing an unknown, which in the sentence is replaced by, for example, *that*. An example is the sentence:

The bus that Alice took was red.

For this sentence, one could say that there exists a corresponding sentence *Alice took [?]*, where one does not know what was taken in the previous sentence. In the first sentence, Alice's vehicle of choice is extraposed (hence the name *extraposition grammar*) to the left, in the form of *that*, which represents the unknown vehicle. Linguists mark this with t for the trace, what replaces[3] the unknown noun phrase (in this case, the bus); and with i, denoting the relationship between our unknown

[3] A procedure named "*It* replacement" [Bach and Horn, 1976].

vehicle *t* and *that*, the representative of this missing part. This results in the following representation:

$$\text{The bus that}_i \text{ [Alice took } t_i\text{] was red.}$$

Note how the hypothetical sentence involving *t* is included, and *i* marks *t*'s relation to *that*. There are numerous other sides to this approach, such as the constraint that for a given noun phrase there cannot be a relative pronoun that is both outside the noun phrase, and referring to a relative clause inside the noun phrase. However, these are outside the scope of this report. To give an example of the importance of this type of grammar in BusTUC, the following, similar input is given to the system:

"The first bus that stops at Samfundet"

By observing the tree BusTUC constructs for the query, the treatment of a relative clause in extraposition grammar as outlined above can be located: see Listing 2.2's "`rel_clause`", with "`that`" representing the bus that stops at "*Studentersamfundet*" ("`bus`", "`stop`", "`at`", "`studentersamfundet`"). From the left, the second "`bus`" leaf node has been added after analysis like what was introduced above.

Listing 2.2: Output tree for "The first bus that stops at Samfundet"

```
        implicitq
   ---------|--------
  /                  \
lit                  np1
 |           ---------|-------
 |          /                 \
the     np_head            rel_clause
      -----|-----          ----|----
     /     |     \        /         \
   < >   adj2   noun    lit         stm
    |      |      |      |       -----|----
    |      |      |      |      /          \
   art   first   bus   that   < >        do_phrase
                               |        -----|----
                               |       /          \
                              bus   verb           pp
                                     |        -----|-----
                                     |       /           \
                                   stop   prep          name
                                            |             |
                                            |             |
                                           at    studentersamfundet
```

2.1.3 Prolog

In section 2.1.1, parts of the area of computational semantics were outlined, and examples were given for vocabularies and models. Their purpose was to introduce representations that may easily be used to create a system capable of reasoning within its domain. Blackburn and Bos [2003] give descriptions of the implementation step using Prolog, a logical programming language excellent at this type of work. As BusTUC is written in SICStus Prolog[4] (which is an implementation of ISO-Prolog[5]) some examples are given below to show a glimpse of the structure of Prolog in practice.

Building on the examples in section 2.1.1, the vocabulary would in Prolog look like this:

```
relation(stopsat,2).
relation(bus,1).
relation(station,1).
constant(buss66).
constant(samfundet).
```

Using the vocabulary defined in Prolog above, the model becomes:

```
[bus(bus66),station(samfundet),stopsat(bus66,samfundet)]
```

Note that with Prolog, one here follows the Closed World Assumption (CWA) [Reiter, 1978], meaning that what is not modeled as *true* is always treated as *false*. For example, it is not ambiguous whether Bus 66 is also a station – the absence of "`station(buss66).`" means that Bus 66 is (in Prolog's eyes) *not a station*.

A practical query example from BusTUC is given as:

```
[which(A):::(dragvoll isa station,B isa bus,A isa time,
         dob/leave/B/dragvoll/C,srel/in/time/A/C,event/real/C)]
```

As explained in section 2.1.2, this is a logical request for a time `A` corresponding to bus `B` passing the "Dragvoll" station. The initial question has been transformed into a form with which a model established as above (though obviously a far more complex model) can be queried. This is the TUC Query Language (TQL). Prolog will go through its rules in order to find an `A` that fulfills the given conditions (being the time at which a bus `B`, leaves the `station` named `Dragvoll`).

2.2 BusTUC and associated systems

This section explores both BusTUC and various related projects. As BusTUC is a part of this project, this section provides it with both an explanation and a

[4] https://sicstus.sics.se/

[5] http://www.deransart.fr//prolog/docs.html

context.

2.2.1 BusTUC

The BusTUC system, described in [Amble, 2000], is built as a special application of the more general The Understanding Computer (TUC) system for language comprehension. TUC was introduced in [Amble, 2004], where a brief introduction to BusTUC was also provided, and was built to carry on the results of even earlier projects at NTNU that could easily be adapted to more specialized use.

BusTUC's application of TUC is made to process and answer queries about the bus routes in Trondheim. In 1999, it became a part of the services offered by the local bus transportation company in Trondheim, and its service is used in several other applications as well. The functionality of BusTUC is achieved through a large number of rules, implemented in Prolog (section 2.1.3), which the system churns through to provide an answer to each query. Given a question in one of its supported languages (Norwegian and English), it will go through its grammar rules in order to understand what information is required. By breaking down the query to comprehend it in this way, BusTUC lets users submit questions without any requirements for how the question is phrased. This allows users to write their question the way they feel is natural, and despite the large number of possible ways to phrase a question, the system will handle the rest.

BusTUC accomplishes the above through its three main components:

1. A parser system for natural language input

   The parser system contains a dictionary, a grammar, and a lexical processor.

2. A knowledge base

   The knowledge base has two parts: one for general semantics and one for the application.

3. A query processor

   The query processor contains a bus route database and a logic system for routing.

BusTUC's flow is shown in Figure 2.1. TUC converts a given natural language query to a TQL expression, which BusTUC then finds all corresponding rules for. It then instantiates the resulting program to find the specific entities involved (such as a specific bus), generates and runs an answer program, and passes the output to the user.

As section 2.1.2 explains, BusTUC uses a variant of Extraposition Grammar called Consensical Grammar – "CONtext SENSItive CompositionAL Grammar".

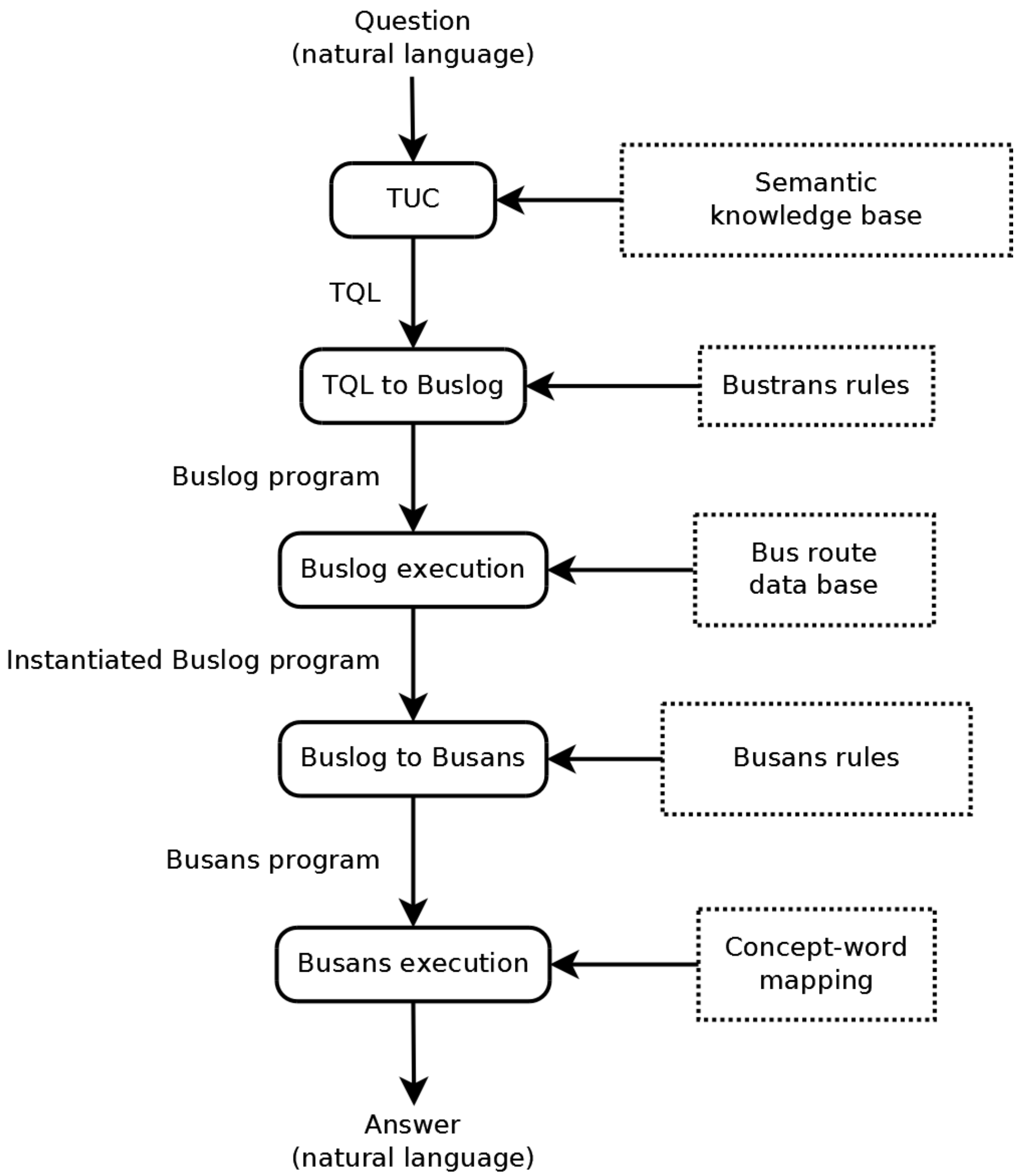

Figure 2.1: BusTUC architecture (adapted from [Bratseth, 1997]).

With Compositional Grammars, the semantics of phrases are composed of the semantics of their sub-phrases. BusTUC's parser system uses this grammar on statements, and converts questions into a statement-based form. For example, a question like "Which bus stops at Lade?" would become closer to "For which X is it true that X is a bus that stops at Lade?". The statement is in this case "X is a bus that stops at Lade". BusTUC then analyzes the statement to find out what fits the statement about "X". To disambiguate the input, the parser discards semantically incorrect interpretations early, prioritizes long (and semantically correct) interpretations of phrases, and is forced to make some committed choices along the way.[6]

The rules necessary for BusTUC to understand its copies of AtB's bus routes, such as for example definitions of bus stops, form one of the most important parts of the system. The semantic knowledge base contains information on the meaning of words and how they work, and is the source of most of the logic of BusTUC's "world". It is also the place where most changes are made, while the dictionary and grammar are kept relatively constant.

Much of the work performed by the query processor is to identify requested information about a certain event, such as the time of a departure. Below are some examples of queries that the resulting system can answer, along with the returned information. "Q" and "A" mark questions and their answers, respectively. When only one location is given, the other place (destination or origin) is assumed to be Trondheim's city center.

Q: *When does the bus leave Dragvoll?*

> **A:** Bus 9 passes by Dragvoll at 1:05 pm and at 1:25 pm and arrives at Sentrumsterminalen, 24-31 minutes later. Bus 5 passes by Dragvoll at 1:12 pm and arrives at Kongens gate K1, 18 minutes later. Bus 5 passes by NTNU Dragvoll at 1:13 pm and arrives at Kongens gate K1, 17 minutes later. Bus 36 passes by NTNU Dragvoll at 1:16 pm and arrives at Sentrumsterminalen, 14-18 minutes later. Bus 66 passes by NTNU Dragvoll at 1:17 pm and arrives at Munkegata M1, 32 minutes later. The hours indicate the earliest passing times.

Q: *From Nardo to City Syd*

> **A:** The station nearest to Nardo is Nardosenteret. Bus 8 goes from Nardosenteret at 1:16 pm to Prof. Brochs gate at 1:23 pm and bus 46 goes from Prof. Brochs gate at 1:41 pm to John Aaes veg at 1:49 pm. The hours indicate the earliest passing times.

Q: *What is the cheapest bus to Oslo?*

> **A:** I have only routes for buses in Trondheim.

The questions above were asked and answered through the "Bus Oracle" feature on AtB's web pages[7], which use BusTUC for this feature. However, to get a

[6]Such "cuts" are made at certain places in the grammar, blocking backtracking past the cut, but is a necessity because storing every possibility for backtracking would impact the memory usage and performance too severely.

[7]https://www.atb.no/ (as of June 11, 2015)

more visual demonstration of BusTUC's query breakdown, BusTUC's website[8] at Department of Computer and Information Science (IDI) can be used with the *tree* option selected. The tree in Listing 2.3 shows the output given when we request query breakdown information in the manner above, for the query *When does the bus leave Dragvoll?* from the table above. This gives a decent illustration of the way BusTUC understands queries.

Listing 2.3: Listing 2.1 repeated: Output tree and logic for "When does the bus leave Dragvoll?"

```
                     whenq
      ----------------|----------------
    /          |                        \
   lit         pp                        ynq
    |        ---|--            -----------|----------
    |       /      \         /                        \
  when    < >    np_head    lit                        stm
           |       --|-      |              ---------|---------
           |      /    \     |             /                    \
         prep   < >    < >  does       np_head               do_phrase
                 |      |              ----|---              -----|-----
                 |      |             /        \            /           \
                the    noun     determiner0   noun     vp_kernel        < >
                                   -|-          |       ---|---          |
                                  /   \         |      /       \         |
                                 lit  < >      bus   verb      name   adverbial1
                                  |    |              |          |
                                  |    |              |          |
                                 the  art           leave    dragvoll

[which(A):::(dragvoll isa station,B isa bus,A isa time,
             dob/leave/B/dragvoll/C,srel/in/time/A/C,
             event/real/C)]
```

Below the tree in Listing 2.3, we also see the resulting logical expression that is generated from the query: TQL. This is the form in which the question is handled when BusTUC attempts to produce the necessary information for the user, through rules established as described in sections 2.1.1 and 2.1.3.

2.2.2 Applications of BusTUC

As mentioned, there are several other services using BusTUC's functionality as well. This section introduces some extensions to and applications of BusTUC.

[8] http://busstuc.idi.ntnu.no/ (as of June 11, 2015)

The interfaces offered through AtB[9] and IDI's web pages[10] have already been introduced (section 2.2.1), and as a consequence of their simplicity, they require no further explanation. The same "bus oracle" functionality is also included in the popular Bartebuss[11] application for mobile devices. In fact, among the five Android applications mentioned in section 1.1, three use BusTUC, including the two most popular[12] among them.

Another relatively simple side of BusTUC is the Short Message Service (SMS) interface it offers. This works just like the "normal" BusTUC, but gives shorter answers to avoid making lengthy (or numerous) SMS messages necessary. Some applications that require longer descriptions are outlined below.

TABuss

Tore Amble Buss (TABuss) [Marcussen and Eliassen, 2011; Eliassen et al., 2012] is a mobile application for the Android platform, and provides helpful information to bus travelers. It is based on the work done by Raaum [2010], which created an Android application capable of taking real-time bus data, user Global Positioning System (GPS) location and bus stop GPS locations into account and use this to provide helpful information through BusTUC. Marcussen and Eliassen expanded and tested this to produce an improved result which is available on Google Play.[13]

Raaum's application provides a field for query input, along with a map with the user and bus stops marked, and text output. TABuss displays a similar input field, lists input suggestions and has separate answer and map screens. Both versions use destination-only input, as they automatically use GPS to get the departure location (as context for the query).

MultiBRIS

Made in parallel with TABuss, MultiBRIS ("A Multiple-platform approach to the Ultimate Bus Route Information System for Mobile Devices") [Andersstuen and Engell, 2011; Engell et al., 2012] is a context-aware multi-platform system for mobile devices, outputting bus route information in much the same way as TABuss. As with TABuss, the starting point for MultiBRIS was Raaum [2010], but unlike TABuss, his version exists on Google Play as a prototype, not a complete application.

[9] https://www.atb.no/

[10] http://busstuc.idi.ntnu.no/ (as of June 11, 2015)

[11] http://bartebuss.no/favoritter (as of June 11, 2015)

[12] Bartebuss and Busstider have the most downloads on Google Play among these five (10 000 to 50 000 and 5 000 to 10 000, respectively; the others have far fewer), as of June 11, 2015.

[13] Google Play: https://play.google.com/store/ (as of June 11, 2015)

Speech-based BusTUC

TaleTUC is the result of the combined work of Engell [2012] and Andersstuen and Marcussen [2012]. They designed TaleTUC as a proof-of-concept system where TABuss was extended to act as a client in a client-server architecture allowing TABuss to support voice-based querying of TABuss and BusTUC. Like the systems above, this is a context-aware application, and combines Case-based Reasoning (CBR) and Automatic Speech Recognition (ASR) to recognize the names of bus stops in the speech input. With TaleTUC in TABuss, the input speech is sent to the server, along with device ID and its location. The server has three modules, one for translation between text and speech, and two for case-based reasoning. These analyze the information and return the resulting interpretation to TABuss.

BusTUC also has a prototype speech-to-speech system in Norwegian. This uses the BUSTER Text-To-Speech (TTS) system [Johnsen et al., 2003, p.125–131] and the Norwegian speech engine “Nora” to answer the questions in synthesized speech [Engell, 2012]. It supports dialogue-like querying. This allows user to, for example, state where they want to travel *to*, at which point the system will ask them where they want to travel *from*. Once all necessary information is received, the system provides an answer in synthetic Norwegian.

BUSTER is a part of *Telebuster*, which has been used at NTNU before, in a project to provide automatic dialogue-based visitors’ guide, *Marvina* [Hartvigsen et al., 2007].

2.3 Similar systems

There exist a number of systems providing services similar or otherwise relevant to BusTUC, using various other approaches to answer the users’ information needs. This section outlines published literature on the topic, including both work from close to the time of BusTUC’s inception and recent developments. A significant difference is that the flexible natural-language input of BusTUC is mostly absent from these systems.

2.3.1 Dutch spoken dialog system

[Strik et al., 1997] describes the construction of a spoken dialog system for Dutch public transportation information, made to handle queries for routing information so the *human* support could focus on the other queries. Made for a considerably different scale than BusTUC, this service was given a schedule database for all public transport companies in the Netherlands. Its routing information was given in answer to origin and destination locations provided as input, which resembles a large portion of BusTUC’s queries. It was not made with BusTUC’s input flexibility, demanding that both origin and destination be explicitly given in answer to a series of fixed questions. In [Strik et al., 1997], a German prototype system was used as the foundation, and a "bootstrapping" approach was used, where they tailored the system to handle Dutch instead, then iteratively improved this based on small-scale user testing. With a functioning system as the starting point, this approach involves a requirement which cannot always be fulfilled, but remains relevant to improvements of BusTUC.

2.3.2 Hong Kong Public Transport Enquiry System

To allow travelers to navigate Hong Kong's complex public transportation systems with relative ease, a web application was created as documented in [Pun-Cheng, 2012]. Much like [Strik et al., 1997], it offers little flexibility for the input, though it allows for selection of origin and destination both in writing and using a digital map. Pun-Cheng also focuses on accurate computation for routing involving special points of interest in Hong Kong, something BusTUC also supports to some extent for Trondheim.

2.3.3 Siri

In recent years, digital speech-to-speech services have experienced an increased popularity. Apple's "Siri" may be largely to thank for starting much of the attention such services now receive from the general public. Siri is a digital assistant, released for the iPhone in 2011. It uses speech processing to interpret spoken requests, then retrieves and presents an answer to the input. Though Siri has received criticism for imprecise interpretation and slow results [Pogue, 2012], it quickly became a useful tool for many iPhone users. Siri is known to be context aware, adapting to its user over time [Geller, 2012], but Apple has not made much detailed information on Siri's technology available. With iPhones running the iOS 8 system, Siri can function as a travel assistant[14] if the user is in a supported area.

2.3.4 Google Now and Google Maps

Google's "Google Now"[15], unveiled in 2012, is Google's equivalent of Siri, and has received much praise, sometimes in contrast to Siri. It was named "Innovation of the Year" of 2012 by the magazine "Popular Science" and is also context aware. As Google Now works well with Google's other services, its quality as a route planner

[14]https://www.apple.com/ios/feature-availability/#siri-directions (as of June 11, 2015)
[15]http://www.google.com/landing/now/ (as of June 11, 2015)

is not surprising. It can even detect the user's travel patterns and email topics, and use that to offer useful information [Manjunath, 2014].

Google Maps[16] already offers route planning for pedestrians and car drivers, and if data is available for the area it will work for public transportation as well. Google Maps already has some BusTUC-like functionality, allowing text-based queries such as "from Moholt to Lade". This functionality puts it closer to BusTUC's support of natural language queries than most such systems, but the similarity ends there, as Google Maps does not support other ways of phrasing a "to-from" query than the one used in the example above. Furthermore, Google Maps does not currently include data on AtB's buses. Should this change, however, it would introduce Google Maps as a major competitor of BusTUC.

[16]https://www.google.no/maps/ (as of June 11, 2015)

Chapter 3

Method: Survey

To answer the first research question, it is appropriate to ask users and potential users, as they are the ones who ultimately decide the usefulness of the system, rather than the system's designers. "Usefulness" being a subjective measurement, it is most natural to perform a qualitative evaluation of this, rather than quantitative. What do people expect from NLP-based query systems for bus route information? This chapter describes the creation and distribution of a survey with the purpose of assessing this.

Three different surveys are mentioned in this chapter:

- The main survey used in the pre-study introduced in section 1.2.
- The main survey used in this thesis. This is referred to as the "new survey" when compared to others, and simply as the "survey" in later chapters where it is the only survey discussed.
- An initial survey attempt in the pre-study, referred to as the "first survey". It was conducted more locally than the others, but has little value beyond this due to its very limited number of participants.

3.1 Distribution

All surveys were distributed using tools on the web. The first survey was distributed through Facebook[1], a social network, using advice on timing.[2] Due to the social circle of the account used on the site, this led to a set of participants containing mostly people who had visited or lived in Trondheim. However, the number of participants gained was insufficient. This approach worked reasonably

[1]https://www.facebook.com/ (as of June 11, 2015)
[2]http://blog.surepayroll.com/post-pin-tweet-best-time-to-outreach/ (as of June 11, 2015)

well for [Nakken and Nascimento Bakke, 2015], another BusTUC project, but Facebook's algorithm for spreading posts does not treat everything equally, and people's enthusiasm for surveys while browsing Facebook is not necessarily high. [Nakken and Nascimento Bakke, 2015] had already tried personally conducting surveys at bus stops, but they found that the limited time (before people leave by bus), limited willingness to answer surveys, and limited parallelism (one person at a time) made it far too inefficient.

The manner of distribution was therefore chosen to be *SampleSize*[3], a survey-focused section of Reddit[4], an extremely diversely multifaceted online discussion platform. Given that the purpose of *SampleSize* is to provide and answer surveys of all kinds, and its acceptable level of activity, it quickly yielded results that were sufficiently numerous, but also few enough for manual treatment.

Participants were not materially rewarded for completing the survey, as the distribution method (*SampleSize*) typically does not have (or require) such an incentive.

3.2 Creation

The surveys were created and conducted using survey tools on the internet, and were conducted in English. In the pre-study, both surveys were created using Google Forms[5], a free online tool for conducting surveys. With the new survey, this was switched to Qualtrics[6] for its more advanced randomization features.

In the creation of the survey, advice from Kelley et al. [2003] on survey style was heeded. The surveys were made simple, providing little more than the most necessary information, and only the most central question was made mandatory. This is because when people can choose whether or not to answer a survey, they are far more likely to answer if it seems easy to do so – this was clearly confirmed specifically for the *SampleSize* community in a survey it did on and about itself.[7]

3.3 Learning from the earlier surveys

This section focuses on the weaknesses of the pre-study's sixth question, which asked participants to pick what they perceived as the most useful improvements to BusTUC, based on the information they were given. These weaknesses may

[3]http://www.reddit.com/r/samplesize (as of June 11, 2015)

[4]http://www.reddit.com/

[5]Google Forms, http://www.google.com/forms/about/ (as of June 11, 2015).

[6]www.qualtrics.com (as of June 11, 2015)

[7]http://www.reddit.com/r/SampleSize/comments/ticys/results_from_the_meta_survey_about_surveys/ (as of June 11, 2015)

have strongly influenced the results, and are therefore addressed in the creation of a new survey.

The results of improvement ranking in the pre-study are shown in Figure 3.1.

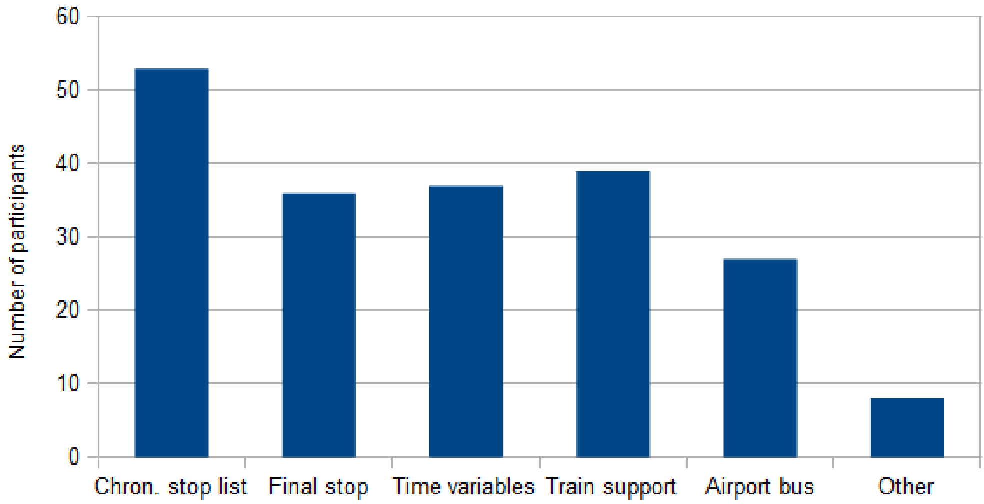

Figure 3.1: Improvement popularity in the pre-study.

3.3.1 Train ambiguity

In the pre-study, the second most popular option was train support. However, a problem becomes apparent when reading the queries it asks participants to write for BusTUC. The train-related queries appear to focus less on trains in and out of a city, and more on “subways”, which by many are colloquially referred to as trains. This becomes a major and problematic ambiguity in the survey. A subway system would indeed be a very natural inclusion in this type of route planner for a city, but while trains in and out of the city can be useful as well, it is a very different matter. As most participants of the survey on Reddit’s *SampleSize* were probably unfamiliar with Trondheim, they would not know that the city has no subway, making subway support irrelevant.

Comparing surveys

To investigate this problem further, the following subsection studies a part of the very first survey, which due to its distribution method had a significant percentage of Trondheim-dwellers among its participants. The first survey is not otherwise used because of its few responses (merely 16 in number), but if the popularity of the “train” option differs greatly from the one previously described, it *may* indicate that the ambiguity was an important factor in the *SampleSize* survey.

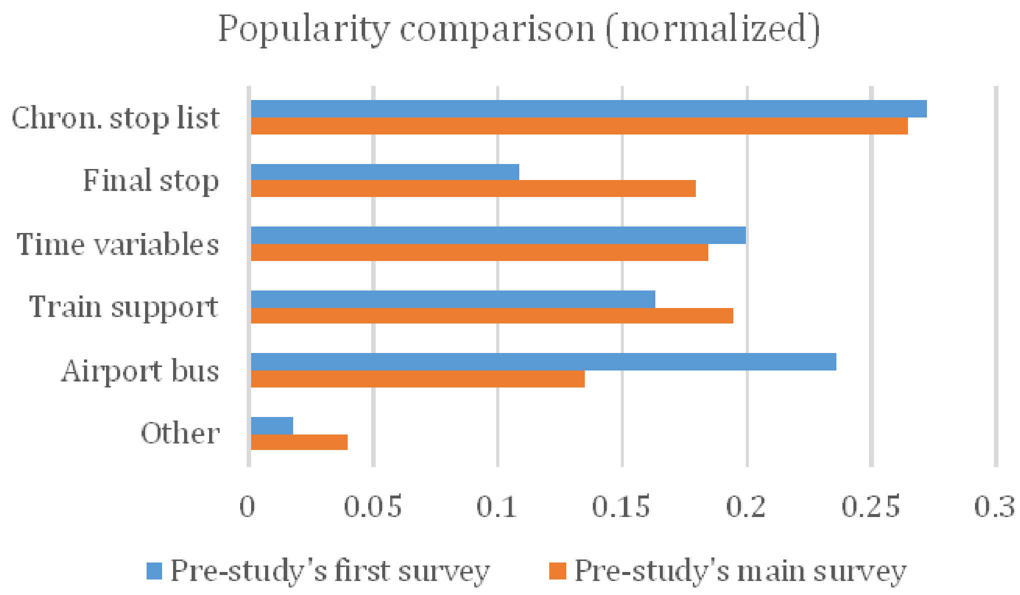

Figure 3.2: The popularity of improvements, in both the pre-study's main survey and its first survey, both normalized.

Figure 3.2 displays the results of the first survey's improvement suggestions alongside the corresponding results of the "previous" survey. Both have been normalized for the sake of the comparison.

Observe that while train support was the second most popular option in the pre-study's main survey, its first survey's (admittedly few) responses placed it in fourth place, below both time variables and airport bus support. This *could* indicate that the participants of the original survey, being mainly bus users in Trondheim (see Figure C.1 in Appendix C), rank train support lower because they know that there is no subway within the city.

Though the number of participants in the original survey is too low for any strong conclusions to be drawn, these are interesting and important observations to keep in mind.

Furthermore, there were no train-specific questions suggested in the first survey, further strengthening the hypothesis that ambiguity was a major factor in its popularity in the pre-study's main survey.

3.3.2 The order of alternatives

The question may also have created an advantage for the suggestions that were at the top of the list, as the pre-study did not randomize the order of the suggested improvements. One can assume that most participants read them in the order in which they were listed, and that many started choosing before having read them all. Having at first no competition, the sensible-sounding "chronological stop list" suggestion might make a participant think it an obvious choice. Subsequent suggestions would be presented with stronger competition in the mind of the participant, who could even have selected three options already and therefore be less motivated to select more. This might have been the case especially because the participants did not know the place in question. Had they known Trondheim, they might have stronger opinions on the subject, easily overpowering this tendency, as exemplified by the comparison in section 3.3.1.

3.3.3 GPS support

From what the literature review in chapter 2 revealed, GPS is not really BusTUC's task; BusTUC's service is primarily a text-to-text NLP system for finding information related to bus travel in Trondheim, though it is used as part of the service of several applications that provide GPS-based assistance. For example, TABuss (section 2.2.2) uses GPS in combination with BusTUC, but the fetching of the initial GPS data (translating "here" to a GPS position) is not the task of BusTUC itself.

However, this was not made sufficiently clear in the pre-study. Because of this, many of the suggested queries were of a type that required translating words like "here" to a GPS position. To ensure that the responses received are as relevant as possible to the system in question, it is important that the new survey makes this distinction clear to all participants.

3.4 The new survey

With the weaknesses of the pre-study discussed above in mind, a new survey was constructed and conducted. The resulting new survey is similar to the one in the pre-study, but with steps taken to ensure that the pre-study's problems do not bias or obscure the results. Thus, the new results may form the basis for work done on the BusTUC system. Based on feedback to the pre-study's survey, an additional suggestion was also added to the improvement ranking question.

3.4.1 Main changes

The main steps taken to address the pre-study's issues, based on the analysis above, are summarized below.

Order bias To ensure no competing alternatives experienced order bias, their order was randomized for each participant. An exception to this rule was the last option, "something else", which was always at the bottom. The static placement of the last alternative was chosen because this alternative referred both to all the other alternatives (with the implied "something else, *not among the alternatives above*") and to the next question, making it more natural and better for the survey flow to have it last.

Train ambiguity To minimize the possibility for confusion about the nature of Trondheim's train service, the city's absence of a subway system was mentioned in the introduction. To increase the likelihood of a participant reading the introduction properly, it was kept as short as possible, and for those who did not read it (or forgot), a short but clear "(not subway!)" notice was added to the "train information" alternative.

GPS queries Much like the train ambiguity, short and clear messages went a long way in removing the inaccuracies regarding GPS queries. A short explanation was added to the first BusTUC-related question, SQ5, saying that the computation of GPS locations is handled by applications *using* the system in question, not the system itself.

3.4.2 Survey content

The content of the new survey is described below (the complete survey can be seen as presented to participants in Appendix A).

SQ1 *Occupation*
This question gathered the necessary minimum of information about the participant, by asking if (s)he is primarily a student, employed, retired or "unemployed / other".

SQ2 *Frequency*
The purpose of this question is to gain a measure of the extent to which the participant is similar to the target market of the BusTUC system and the services it supports.

SQ3 *Location*
As traveling in the countryside can be quite different from travel in a city, capturing this information could be useful. This question thus gathers more information on the "relevance" of the participant.

SQ4 *Information sources*
How does the participant *currently* get information about bus routes?

SQ5 *Query suggestions*
Participants are asked to suggest a few questions for the system, given what they know.

SQ6 *Improvements*
This question asks the participant which improvements seem the most useful, with references to BusTUC's current functionality for context. At least three of them had to be selected. The alternatives were:

1. "Chronological list of stops for a bus (currently alphabetical)"
2. "Information about the final stop/end stop of the bus"
3. "Time variables, such as opening hour information (if something else, specify below) (ex.: '<place> to <shopping center> at opening hour')". (Other variables could be requested in SQ7.)
4. "Information about the trains out of the city (not subway!)"
5. "Support for the buses to/from the airport"
6. "Support for handicap information"
7. "Something else" (to be specified in the answer to SQ7)

SQ7 *Further Input*
Participants could enter further ideas here, or specify their desired functionality if they selected "something else" in question SQ6.

The first suggestion, finding the chronological order of bus stops when asked where a bus stops, originated from discussions with the supervisor and other bus travelers about BusTUC's features, strengths and weaknesses. This was also the case for the fourth and fifth suggestions, whereas the second and sixth suggestions came from feedback and queries given in the earlier surveys. The points of interest (POIs) supported by the information system described in section 2.3.2 led to the third ("Time variables") alternative, which is in essence a more advanced support for POIs.

Chapter 4

Survey results, analysis and discussion

This chapter presents the results of the survey described in chapter 3, along with an analysis and discussion of each part. Dividing the survey results into sections are the three main topics: the participants (section 4.1), the functionality demand (section 4.2), and the queries and further input (section 4.3). Additionally, section 4.4 discusses biases and assumptions, both those that were removed through the analysis and survey creation described in chapter 3 and those that remain or might remain.

4.1 Participants

The survey received 54 responses in total, presumably from 54 different individuals. This section shows the answers collected from the questions that were focused on establishing some basic background knowledge on the people who answered the survey.

4.1.1 Occupation (SQ1)

SQ1 *Which of the following best describes you?*

1. Student
2. Working
3. Retired
4. Unemployed / other

Of the 54 participants, 53 (98 %) answered this question. Among them were 32 (60 %) students and 18 (34 %) workers. It is likely that some were both, but the participants were asked to choose the option which best described them. None (0 %) described themselves as retired, while three (6 %) marked the "unemployed / other" option. Figure 4.1 displays the distribution.

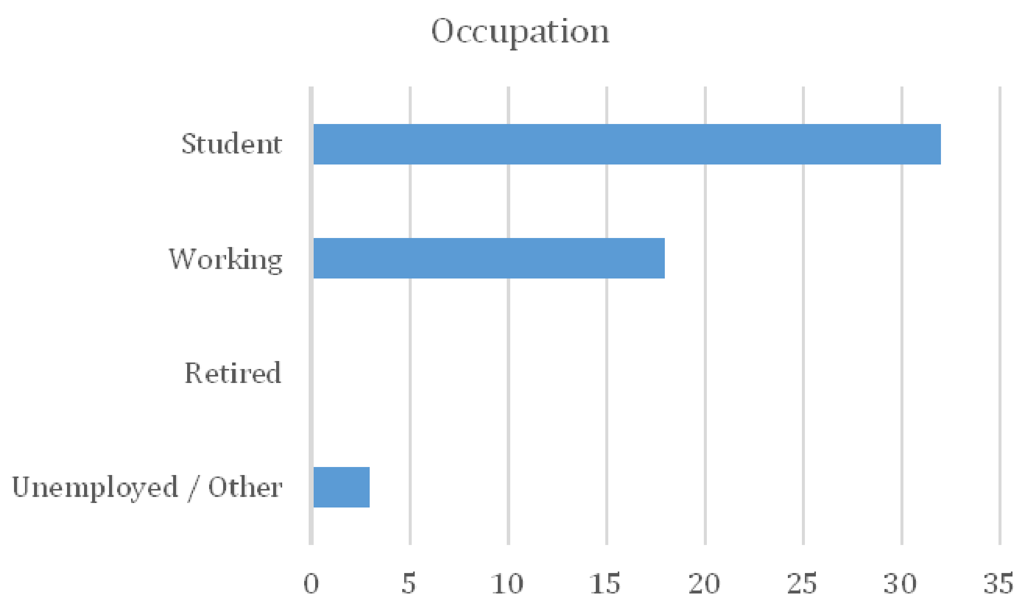

Figure 4.1: Answers to SQ1.

4.1.2 Frequency (SQ2)

SQ2 *How many times do you travel by bus during an average week? A rough estimate is fine.*

1. I do not use any kind of public transportation
2. I never travel by bus (but sometimes other public transportation)
3. Less than 1 (on average)
4. 1-4
5. 5-8
6. 9-12
7. More than 12

Among the 53 participants (98 %) who answered this question, five (9 %) do not use any kind of public transportation, while four (8 %) of them use some, but not buses. Twelve (23 %) use buses less than once a week, nine (17 %) use them one to four times a week, ten (19 %) use them five to eight times, and eleven (21 %) use them nine to twelve times a week. Two (4 %) of the participants travel by bus more than twelve times during an average week. Figure 4.2 provides a visual comparison of the numbers.

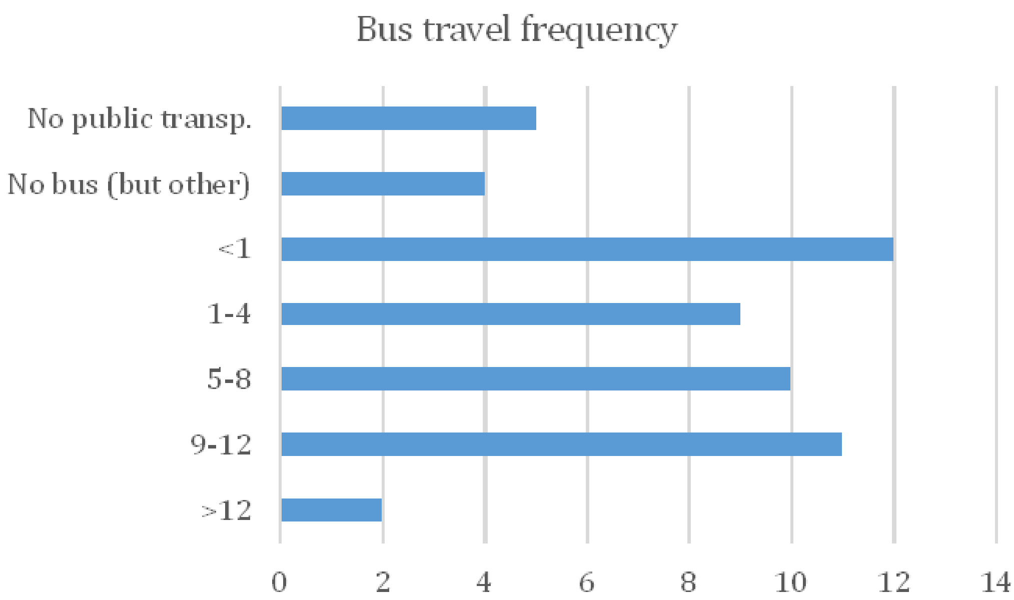

Figure 4.2: Answers to SQ2.

4.1.3 Location (SQ3)

SQ3 *Where do you travel by bus the most?*

1. In a city
2. Not within a city
3. I don't travel by bus

When asked about their primary location for bus travel, 42 (81 %) answered that they travel in a city, while four (8 %) travel outside of cities and six (12 %) do not travel by bus (Figure 4.3).

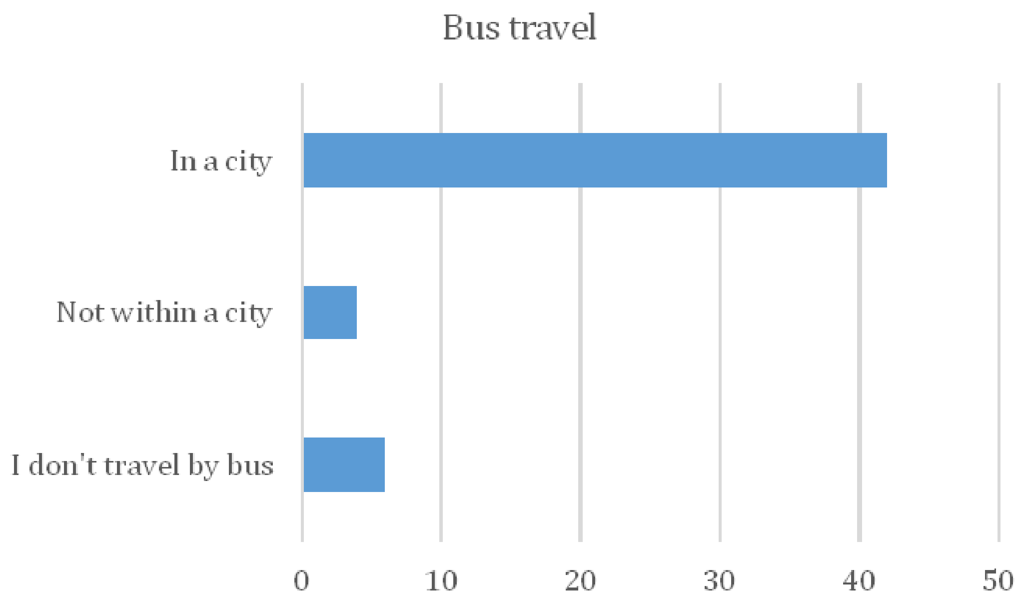

Figure 4.3: Answers to SQ3.

4.1.4 Information sources (SQ4)

SQ4 *How do you get information about bus routes / other public transportation?*

1. Information at the bus(/other) stop
2. Information on paper
3. Application on mobile device
4. Websites
5. I don't seek out information
6. Other: [Input field]

This question asked how participants obtain information about bus (or other public transportation) routes. Multiple answers were possible. The results (Figure 4.4) were: 36 (68 %) use websites, 30 (57 %) use information at the station or stop, 27 (51 %) use applications on mobile devices, nine (17 %) use information on paper, five (9 %) do not seek information, and three (6 %) used something else (”other”).

The three who selected “other” described their information sources as “friend”, “Greyhound.com” and “Google Maps”. Two of these are assumed to be errors on the part of the participant: Friends were indeed not listed among the static alternatives, but *Greyhound* is a website offering travel planning, making this answer belong to the "websites" option. As this participant had not already selected that alternative, it is treated as such in Figure 4.4. Google Maps, on the other hand, exists with the same name as both website and mobile application, causing this answer to correspond to "websites" or "application on mobile device" – or both. As this participant also selected both of those, the "other" entry does not add any further information and is therefore not included in Figure 4.4.

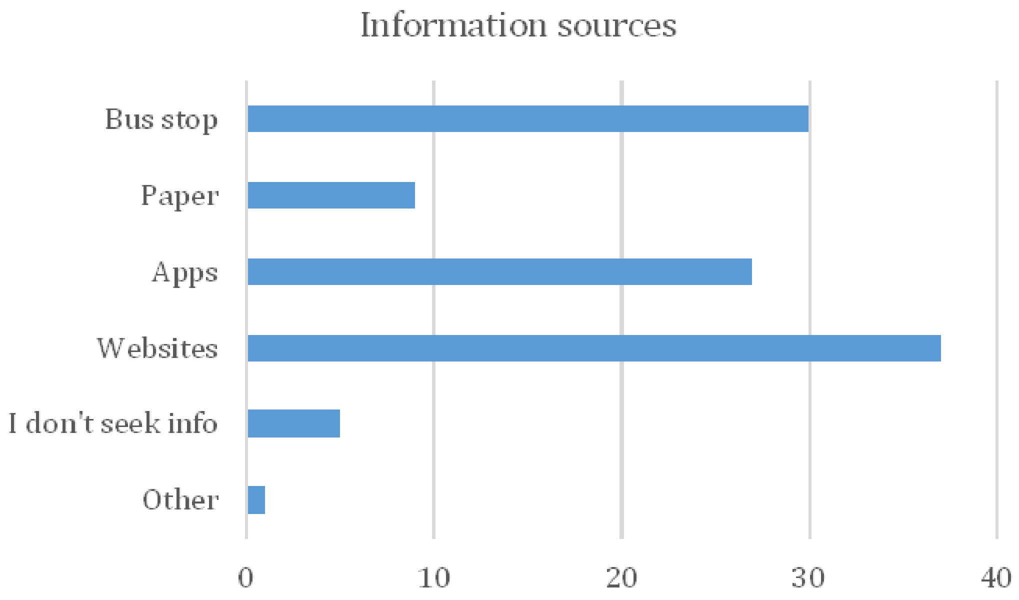

Figure 4.4: Answers to SQ4.

4.1.5 Representativeness

Questions SQ1 through SQ4 provided some background information on the participants. One goal of these questions was to gain insight into whether the participants were relevant or not. For example, if a majority of the answers were submitted by people who never or very rarely travel by bus, then the answers might not be sufficiently representative of the needs of BusTUC's users. Likewise, had most participants traveled mostly outside of cities[1], their bus-related information needs might differ from those of Trondheim's inhabitants.

Thankfully, 83 % of the participants travel by bus, and 81 % do so in a city (ignoring the single participant who refrained from answering these questions).

[1]It is worth mentioning that views of what a "city" is also differ between people. A person living in an area of Trondheim's population density, but somewhat close to Tokyo, might apply the term "city" to one but not the other.

Since a clear majority of the responses were from people traveling by bus within cities, the participants can be said to have habits and opinions that are relevant to Trondheim's bus services. It is somewhat peculiar that while six participants stated that they do not travel by bus in SQ3, the corresponding options in SQ2 showed nine in this group, despite SQ2 being answered by only one more participant. This is assumed to be an error on the part of a few participants, and all in all not particularly significant.

The first few questions also gathered some information on the participants' occupations, as well as their usual methods of finding route information. As students were the majority at 60 %, with workers filling the next 34 %, the distribution may be reasonably close to the customers along a number of Trondheim's bus routes, given the city's relatively large student population. However, there were only three who described themselves as "unemployed", and perhaps more importantly, the retired are not at all represented among the participants in this study. This bias towards younger age groups is probably a result of the distribution method, which was Internet-based and did not focus on covering all age groups. The elderly can be expected to have wishes and expectations that may differ from those of younger customers. Due to their absence in the set of participants, it may be useful to in the future conduct a completely separate survey targeting elderly bus travelers, to ensure that their voices are heard in the evaluation of improvements to BusTUC. They might be more dependent on BusTUC correctly answering their query than their younger counterparts, due to issues with health or technological ineptitude, so it is important not to forget them simply because they do not represent the majority of the market. Given the differences in the level of proficiency (and frequency) with which different generations typically interact with internet-based technology, it might not be unreasonable to assume that the working participants as a group also lean towards the younger generations. Because of cases like this, one must note that the survey results do not provide a *complete* description of demand in BusTUC's market.

The survey's fourth question revealed that the participants' main sources of information about bus routes were websites (68 %), the bus stop (57 %) and mobile applications (51 %). As BusTUC is used both on the web and in mobile applications, the participants appear to suit BusTUC's market in this respect as well.

4.2 Functionality demand

With the opening questions having established that the results of the survey can be seen as relevant to Trondheim's buses, the later questions can be used to gain an overview of the demand a service like BusTUC faces. The results of the voting on improvements (question SQ6) are presented in section 4.2.1, and analyzed and discussed in the later subsections.

The workload associated with each improvement varies, ranging from alterations of existing functionality (though not necessarily as minor as they may seem) to whole new domains to support.

4.2.1 Improvements (SQ6)

SQ6 *What possible additions do you think seem the most useful for this type of system?*

Please select 3 or more.

1. Chronological list of stops for a bus (currently alphabetical)
2. Information about the final stop/end stop of the bus
3. Time variables, such as opening hour information (if something else, specify below) (ex.: "<place> to <shopping center> at opening hour")
4. Information about the trains out of the city (not subway!)
5. Support for the buses to/from airport (not currently supported)
6. Support for handicap information
7. Something else is important to support (specify below)

Participants were asked to select at least three improvements they thought would be useful, among five suggestions and the option to add their own suggestion instead of one of the provided ones. Those that had their own suggestions did this as their answer to the final question (SQ7). The results are shown in Figure 4.5 and are, in descending order of popularity: 43 (80 %) participants voted for sorting lists of bus stops chronologically rather than alphabetically, 32 (59 %) desired support for queries using time variables (such as opening hours), 30 (56 %) voted for information about the final stop of buses, 28 (52 %) for information about buses to and from the airport, 23 (43 %) for information about handicap accessibility, 21 (39 %) for information about trains in and out of the city, and eight (15 %) of the participants pointed to their own suggestion provided in their answer to question SQ7.

4.2.2 Chronological lists and final stop information

The most popular alternative by far was the change from alphabetical to chronological bus stop lists, which 80 percent of the participants voted for. And with the new survey's randomized ordering of alternatives, it is this time certain that

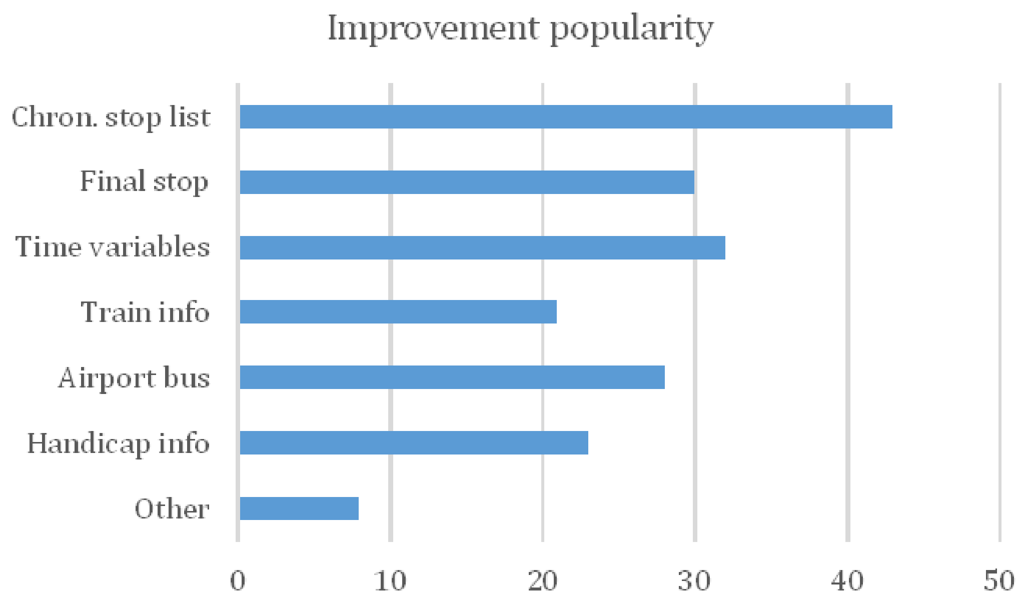

Figure 4.5: Answers to SQ6.

the order did not impart a bias on this result. It therefore appears that among the improvements suggested, the strongest demand is for such chronological lists. The second most popular alternative was support for time variables, only narrowly ahead of the popularity of information about bus destinations (final stops).

Given that the implementation of chronological bus stop lists might contribute to parts of a solution to final stop information in BusTUC's Prolog code (introduced in chapter 2), it might be natural for the implementation of one to follow the implementation of the other. The overwhelming support for chronological bus stops makes it a clear choice for implementation, and with the final stop alternative at 56 %, implementing both would be very reasonable.

4.2.3 Time variables

Despite its popularity, ranking second in the answers to question SQ6, the "time variables" option may have impractical side effects. For queries relying on information about opening hours (or similar time variables) to be useful, support for the opening hours of only a few places is not sufficient. The feature needs to be reliable, making a large number of supported time variables necessary. This in itself is not a problem. The problem becomes clear when one considers the fact that the time variables are, after all, *variable*; they may vary. And more importantly, they may vary independently. One would therefore have to maintain that

BusTUC's opening hour for each supported establishment is correct. Two options immediately present themselves: all of them could be checked at regular intervals, a method which scales badly and risks leaving some time variables incorrect for unacceptable amounts of time. Alternatively, arrangements could be made for the establishments to send their updated opening hour information themselves. However, this is likely to be forgotten regularly, and it would take a large amount of effort to promote the establishment of this procedure for the city's shopping centers, gyms and other points of interest. Unless a reliable and efficient arrangement can be found, this option may demand too much maintenance, or produce too many errors, to be a useful addition in the long term.

If implemented, time variables should also be accompanied by improvements to the way BusTUC handles points of interest, such as shopping centers. More such points of interest should be recognized as viable origins and destinations, and have their opening hours included in an eventual implementation of time variable support. As BusTUC can find bus stops based on street names, and knows *some* points of interest (such as the Nidaros Cathedral), it would not be unreasonable to expand this to include Trondheim's major shopping centers and police station. In this case, the queries would be mostly the same as ones BusTUC already supports, so the knowledge base would have to be expanded with the names and positions of the points of interest.

4.2.4 Trains

This might be one of the most demanding suggestions, as it shares less with BusTUC's existing domain than the others, but for the same reason it could be a major expansion of BusTUC's service. If successfully implemented, it would allow travelers to take the bus to the train and receive information about both parts of the journey from the same source. This would also include a train going to the airport, which is an alternative to the airport buses listed below. However, it received the fewest votes of all the suggested improvements.

4.2.5 Airport buses

Airport bus support also received a significant number of votes. As the domain of this improvement shares much with what BusTUC currently supports, it should fit the current service well.

Though the low number of responses to the very first survey attempt is too low for any strong conclusions, it is also interesting to note that the comparison of the two first survey versions (Figure 3.2) reveals a major difference in the popularity of airport bus support. This may be because the majority of the first survey's participants were students in Trondheim. Many students travel to their families' homes during the winter, summer and Easter holidays. As many do not own cars, this leads to visiting Trondheim's closest airport six times each year being reasonably common. For the participants in the later surveys (both "previous" and "new"), however, this might not be the case. It is therefore worth noting that the airport bus support might be more valuable to the population of Trondheim city than the main survey makes it seem.

To support this, the different companies providing buses to the airport would have to be contacted and an agreement made to maintain up-to-date route information from all of them. Introducing multiple sources like this would increase the frequency with which BusTUC would require updates to its route information.

4.2.6 Handicap information

The addition of handicap information also received support from a number of participants. Though it ranked fifth among the suggested improvements, its popularity was impressive for an alternative that most probably does not directly affect the majority of the participants. Many seem to agree that the significance of the feature to those who need it is large enough to be a priority.

This particular option was originally from a response to the pre-study's survey. Even then, there was only one question about the handicap friendliness of the buses, but to customers who need it, this information could be of the utmost importance. As section 4.1.5 mentions, one should be careful not to ignore the needs of certain groups merely because they are a minority.

As BusTUC appears to be unable to understand sentences about wheelchairs and handicaps, implementation would require the addition of not just the handling of such information, but also the definitions of the words involved. On the other hand, the output might be quite simple and apply to most or even all buses. The required additions to the semantic knowledge base should therefore be relatively small.

4.3 Queries and further input

In question SQ7, participants could elaborate on their selection of improvements and otherwise provide feedback, and in SQ5 they were asked to provide some questions for the system (based on an explanation given in the survey introduction). Section 4.3.1 describes the feedback gained with question SQ7. The queries received from participants (question SQ5) are presented in section 4.3.2, along with BusTUC's performance when tested with them. Both the queries and the feedback are then analyzed and discussed in section 4.3.3.

4.3.1 Further input (SQ7)

SQ7 *Any other notes on information you feel is important/useful for such a system to support?*

(If you chose "something else" above, please elaborate here)

The last question let participants provide further notes on information they felt would be important for a system like BusTUC[2] to support. This question also let participants elaborate on their answer to the previous question (requested of participants selecting the "something else" option).

In total, ten participants provided feedback here, 8 of whom elaborated on their choice of "something else" in question SQ6. The feedback received in this question was about the following subjects:

1. Delay information: real-time bus times
2. Frequency information: how often a bus passes a station
3. Transfer information
4. Station landmarks: to help visitors recognize their destination
5. Real-time bus GPS position

4.3.2 Query suggestions (SQ5)

Among the 54 participants, the 39 (72 %) who submitted query suggestions gave a total of 123 queries.

SQ5 *Please think of a few questions for the information system to interpret and answer, as you might need it to when traveling by bus.*

They should be about public transportation, and ones you think you might use.

Place names: *A/B/etc can be used as placeholders for place names.*

No GPS-based queries: *As such things are handled by applications using this system, not the core query system, GPS queries such as ("From* here *to Place_B") are not relevant here.*

A few examples are listed below, exactly as received. (Note that participants were asked to use placeholders where place names were required, hence "Bus A", "Place_B", "x" and similar oddities.)

[2]Based on the survey introduction's description of the system.

- “Next bus from Place_A to Place_B”
- “Price for Place_A to Place_B”
- “Is Bus A on time?”
- “When is the next bus to A from B?”
- “When will the next bus arrive at bus stop B?”
- “When will Bus x arrive at Station_A around 7 PM”

Running the queries

The suggested queries were all used to test the BusTUC system, by giving them as input.[3] Similar or identical queries were not removed, as common query types being relatively numerous in the query set lets the set more accurately represent the actual queries of new BusTUC users. Placeholder names such as “Place_A” in the queries were replaced with actual place names in Trondheim. The same was done to placeholder addresses, bus route numbers, times of day and shopping centers. The replacements used are listed in appendix B along with the queries (both original and edited versions) and the results of using them as input.

The results of testing BusTUC with the suggested queries are given in Table 4.1. Dividing the number of good results by the total number of queries and multiplying by 100 to obtain the percentage, the portion of queries that yielded good results from BusTUC was found to be $(43/123) * 100 \approx 35.0$ %.

Result quality	**Number**
Good	43
Bad	80
Total	123

Table 4.1: Result quality of the suggested queries as input BusTUC.

4.3.3 Queries and feedback topics

The survey’s fifth question yielded another type of improvement suggestion. It requested questions for the system, based on the description given in the survey. These questions show what the participants expect from the system, and can be seen as a practical set of descriptions of the demand on BusTUC and to some extent

[3]The testing was done on a locally run version of BusTUC.

other NLP-based route planners. While the quality of the suggested queries varied, they are not necessarily less important than the results of the voting in SQ6. The queries indicate what functionality is perceived as beneficial, and can be used as input. This makes them useful for both deciding what is best to implement and measuring the change brought by said implementation.

As the questions were left mostly untouched except for tweaks to make them relevant to Trondheim, some were written in more convoluted language than expected. This could be because participants were asked to create useful queries; this process is a creative one, and very different from the purposeful querying performed when one actually needs it. With a success rate of 35 % (section 4.3.2), BusTUC handled it reasonably well, considering the circumstances. Common spelling mistakes were also left as they were, as these will be part of BusTUC's normal input as well, making it reasonable to expect it to handle them.

As 65 % of the participants use websites to obtain bus information and 51 % use mobile applications, most of them are clearly accustomed to obtaining such information digitally. However, as section 2.3 shows, digital tools using natural language queries and answers to provide such services are not very common. Many participants might not have used such a service before, which could have affected the way they phrased their queries.

To some extent, the queries received might be influenced by the improvements listed in SQ6. Some participants might avoid those topics, knowing that they are not currently supported, while others might give queries testing those exact areas. Whether this would influence the query collection toward or away from the topics voted on, is uncertain. The question requesting queries (SQ5) was placed before the voting (SQ6), which most probably reduced this effect, as most participants would complete SQ5 before paying much attention to SQ6. All in all, it is assumed that it did not affect the queries' topics to any significant degree.

Main topics

Among the topics that BusTUC was unable to provide answers for, the most popular ones are described below, each with an example from the queries received. They show which types of queries there is demand for in a practical situation.[4] The topics of feedback given in answer to question SQ7 are also included.

Frequency: "How often does a bus go from A to B?"

> There were numerous queries about the frequency of buses. Despite the apparent usefulness of the topic to the participants, BusTUC's typical response was "I have no information about frequency."

[4] At least practical compared to question SQ6, and as practical as a survey like this is likely to get without having participants use the system themselves.

BusTUC's grammar can already understand such queries, so implementing this would require modifying the semantic knowledge base to support the handling of queries about frequency.

A natural answer to this type of query might be to specify the interval between buses, when the interval changes next, and what the interval is after the change.

Delay and traffic: "What is the estimated time of arrival in current traffic conditions?"

Because buses easily can be hindered by traffic, it is not surprising that their customers are interested in information when a bus is delayed. While some delay statistics for a given day and time might be somewhat useful, BusTUC users would probably prefer more precise information.

As mentioned in section 2.2.1, BusTUC keeps its own copies of AtB's bus routes. This means that it does not have to communicate with AtB servers to provide its service. Because of this, supporting information on current delays would require the construction of a module capable of communicating with AtB to retrieve such information from them. And sadly, the amount of such information that is available from AtB at this time is lacking. However, AtB has informed the BusTUC group that a new system for real-time GPS information is expected to become available within a year or so. If BusTUC were to be expanded to support such information, it would fulfill much of the demand that lies behind queries such as the example above. Alternatively, a separate service using BusTUC might be expanded to support it. It would in that case not benefit all applications of BusTUC, but it might still be the more natural option, similar to the case of GPS as presented in section 3.3.3.

Length of time: "How long will it take to get from A to B?"

Among the suggested questions for BusTUC were several that requested information about the time a bus trip takes, using the word "long". BusTUC, not realizing it was asked for a length of *time*, would simply answer "I have no information about lengths".

Because this type of query probably is more likely than questions about distance, it might be useful to the users if BusTUC's semantic knowledge base was altered to generally treat such questions as questions about time. Doing this might make the system mistakenly answer queries about the distance of bus trips, but this might be far less common.

A few topics from the feedback, that were excluded from the discussion above, were:

- Transfer information (already supported by BusTUC)
- "GPS" support (see section 3.3.3)
- Descriptions of nearby landmarks for every bus stop (too time-consuming for its minimal demand)

4.4 Bias and assumptions

The results show some interesting changes compared to the earlier versions, many corresponding to areas where the pre-study was deemed too biased. This section analyzes and discusses biases that were removed (section 4.4.1), as well as assumptions and possibly remaining biases (section 4.4.2).

4.4.1 Reduced bias

The inaccuracies and possible biases in the earlier versions of the survey diminished the certainty with which one could use the results. By constructing and conducting a new survey to remove these weaknesses, the significance of the findings was increased. Without the possible option order bias and ambiguities, the new results reveal several interesting changes, some of which can be expected to come from the differences between the surveys.

The clearest difference is how the train support dropped from being the second most popular option to being the sixth, a dramatic drop which can very reasonably be seen as confirmation that the ambiguity in the earlier surveys was strongly affecting the result.

Interestingly, the implementation of chronological stop lists retained its strong lead on the others, showing that it was indeed its perceived usefulness – not its position in the list – which led to its popularity.

4.4.2 Assumptions and possible biases

Few participants chose the "other" option. As it requires more independent thought and imagination than the other alternatives, this is not surprising. Furthermore, simply by virtue of being shown to every participant, the improvements suggested through the five first options gain an advantage. If not among them, even a truly excellent idea would only be considered by those who happen to think of it; on the other hand, the improvement suggestions that were provided were considered by everyone. Question SQ6 thus naturally gained a bias towards the suggestions that were provided. This might have been the case especially because the participants did not know the place in question; had they known Trondheim, BusTUC or other NLP-based bus information systems well, they might have stronger opinions on the subject, making them more likely to think of improvements on their own.

Additionally, the treatment of the queries given in answer to question SQ5 hinges on some assumptions. To be able to use the queries as input to measure BusTUC's success rate, they were manually edited. This process replaced placeholders with actual place names, buses and times. It also included modification of suggestions that were descriptions of queries rather than queries themselves, or contained multiple queries. A consequence of this is that the results of the query testing depend on the interpretation of each query by its editor. The success rate is therefore the result of a process which is at least partially subjective. In addition to this, there might be alternative substitutions than the ones used for the placeholders, for which the results would have been different. Hence, the treatment of query suggestions in this research depends on the assumptions that this subjectivity does not skew the results significantly, and that the substitution done suits the queries as the participants intended them.

Also worth mentioning is that, presumably, most of the participants do not know Trondheim or BusTUC. A reassurance is in this case that situations with public transportation usually translate reasonably well between cities, and that Trondheim's first-time users of BusTUC are also given only a brief description of the service. Even AtB's "Bussorakelet" website[5], gives an explanation that is roughly on the same level as the one included in the survey.

[5]https://www.atb.no/bussorakelet/ (as of June 11, 2015)

Chapter 5

Method: Implementation

This chapter details the implementation of some of the improvements described in earlier chapters. The survey results and analysis in chapter 4 form the foundation for the choice of implementations.

5.1 Chronological stop lists

Being the single most popular alteration in the survey (and in both the earlier surveys), the change from alphabetical to chronological stop order was a clear choice for implementation.

The procedure in the “Buslog execution” step in Figure 2.1 was modified to ensure that requests for bus stop lists no longer created the list by using a predicate involving sorting the elements. An alternative predicate was constructed, and the previous predicate’s build-in functionality was added afterward when needed. This change was applied to all cases where the list of bus stops for a given bus is created for printing.

The performance impact of the change was evaluated using a test collection containing 24 different queries for stop lists, both in English and Norwegian, and both in normal mode and SMS mode.[1] If the alterations were to greatly affect the performance of the system on these queries, it should show up in the results.

5.2 Handicap

As discussed in section 4.2.6, handicap information received an impressive amount of support considering its lack of relevance to (presumably) most participants. Its

[1]In SMS mode, the output is reduced to the bare essentials to minimize the length.

importance to those it *is* relevant to makes up for the size of this group of users, and also marks it as an important expansion of BusTUC's current functionality.

Initially, BusTUC only knew enough about handicaps to recognize some very limited variations of a few relevant words, and tell the user that it had no relevant information, recommending the AtB website instead. To solve this, it was first necessary to improve TUC's relevant vocabulary. Alterations were made to its dictionary, and much like what was described in sections 2.1.1 and 2.1.3, numerous logical rules such as

```
noun2(handikappet,handicapped).
```

(linking the two words together) were added to tie together the vocabulary, allowing queries about handicap information to successfully pass TUC's lexical analysis.

With a query successfully parsed to TUC Query Language (TQL) (section 2.1.3), BusTUC will go through its rules to attempt to find a procedure solving the problem defined by the query. In the case of handicap queries, the TQL would be something along the lines of Listing 5.1. BusTUC will attempt to find a rule defining the procedure for the given case. Two rules were defined for this step: one for users using BusTUC through SMS, and one for the other, "normal" cases. In essence, the rules require that the query specifies a problem related to the handicap information, and ensure that the answer program includes handicap information. When BusTUC later runs the answer program, the information printed is one of two types of output defined for handicap information: the "normal" version and the shortened SMS version.

Listing 5.1: TQL for "Can I take the bus in a wheelchair?"

```
[test:::('I' isa self,A isa wheelchair,B isa bus,
          dob/take/'I'/B/C,srel/in/wheelchair/A/C,
          event/real/C)]
```

For this feature, AtB provided more detailed information on handicap accessibility than what is present on their web pages.[2] As the information can be relevant not only to wheelchair users, but also other users such as those bringing prams, the handicap information rules were adjusted to react to a few other query types as well, such as queries about prams or what there is room for on the bus.

Additionally, to ensure that implicit questions such as simply "handicap" could also be handled, the system's grammar was expanded with the necessary logic for handling implicit questions about traveling by bus with a handicap.

Note that such single-word queries are currently only supported in Norwegian BusTUC because the underlying grammar in BusTUC is more complex and well-developed for Norwegian. Unsurprisingly, as BusTUC is made for a city in Norway, the Norwegian support has been more highly prioritized for a number of years.

[2]As of June 11, 2015

5.3 Final stops

Ranked a close third in popularity in the survey, information about buses' final stops is a good choice of implementation. Providing the destination (final stop) for a bus is typically useful for determining in which direction to take a bus route, for example, ensuring that the user takes bus *5* towards the university campus rather than away from campus and towards town. This could prove especially useful for customers who do not know the city well, as the destination is displayed on every bus, allowing the customer to immediately decide if it is the correct bus line *and* direction, without any knowledge of where the correct and incorrect final stops lie.

This information was therefore implemented as a part of the standard route planner's answer. The implementation changed mostly the "Buslog to Busans" and "Busans execution" steps in Figure 2.1 on page 14 (the two execution steps involved in the generation of the textual answer). The first of these was expanded with logic adding the generation of destination information to the relevant answer programs. For a given bus or route ID, this logic first fetches the route ID if necessary, then attempts to unify the final stop variable with an actual stop through a number of new predicates. It then becomes necessary to compute which end station is in the direction of the bus trip, to avoid sending users in the wrong direction. First looking up all end stations for the bus trip, the logic then makes use of a new predicate to obtain the list of stations for the bus trip. (Though it was thought that this could be reused from the *chronological bus stop lists* implementation, some differences in procedure prevented this.)

The next new step in the new program execution is a disambiguation of the bus trip's destination station to handle ambiguous station names such as "Studentersamfundet" (there are two "Studentersamfundet" stops). The departure station is treated similarly. This procedure is also designed to handle some inaccuracies in the data received from AtB, such as a destination being "Munkegata M1" for a bus which actually stops at "Munkegata M2".

With both the given stations (origin and destination) unambiguously specified, it is then possible to analyze the previously obtained list of stations to separate the relevant part (stations not yet passed) from the rest. This is then used to filter the list of end stations, obtaining the end station the bus is heading towards.

If the result of this is still more than one station, the answer program has been expanded with the necessary recursive logic to print all of them separated with slashes ("/"), to make the possible ambiguity of the information clear and avoid misinforming users.

5.4 Queries

The test queries received in answer to survey question SQ5 were run to test the system after its improvement, testing BusTUC like described in section 4.3.2. The change was measured in two ways:

Binary BusTUC's response was treated as either a success (1) or a failure (0), as done in section 4.3.2.

Points Each query-response pair was labeled according to the query type and the quality of the response (see Appendix B). These labels were then used to give scores similar to, but more nuanced than the binary grading, by valuing the "successful" labels differently. The labels and their scores were the following:

- **Good** Score: 1. An appropriate result was returned.
- **Better** Score: 1.2. An appropriate and improved result was returned.
- **Partial** Score: 0.5. The query was not answered completely, but some of the desired information was provided. (Example: frequency queries, answered with the next five departures)
- **Indirect** Score: 1. The query was not answered literally, but the information need behind it was most probably met.
- **Fail** Score: 0. BusTUC did non provide the requested information.
- **Timeout** Score: 0. The system timed out while analyzing the sentence.
- **Incompr.** Score: 0. TUC failed to transform the sentence to TQL and marked it as incomprehensible.

Chapter 6

Results

The results of the implementations detailed in chapter 5 are presented here. Section 6.1 describes the results of the implementation of chronological bus stop lists, while sections 6.2 and 6.3 do the same for handicap information and bus destination information, respectively. Section 6.4 details the results of testing the system with the queries from the survey, after the implementation.

6.1 Chronological stop lists

With the alterations described in section 5.1, BusTUC's output changed for queries concerning bus stop lists. The following shows how the output was before, and how it is after the implementation:

Input *Where does bus 63 stop?*

Previous output

Bus 63 goes to the stations
Ankers gate , Dalen Hageby , Festningsgata , Gudes gate , Gyldenløves gate , Hospitalskirka , Høgskoleringen , Ila , Ilsvika , Jonsvannsveien , Kalvskinnet , Kongens gate K1 , Kongens gate K2 , Prinsen kinosenter , Rosendal , Rønningsbakken , Saxenborg allé , Skansen , Strindheim , Strindheim 2 , Studentersamfundet 2 , Weidemannsveien .

New output

Bus 63 goes to the stations
Strindheim 2 , Strindheim , Dalen Hageby , Saxenborg allé , Rønningsbakken , Weidemannsveien , Gyldenløves gate , Festningsgata , Ankers gate , Jonsvannsveien , Gudes gate , Høgskoleringen , Studentersamfundet 2 , Prinsen kinosenter , Kongens gate K1 , Hospitalskirka , Kalvskinnet , Skansen , Ila , Ilsvika , Kongens gate K2 , Rosendal .

Table 6.1 shows the performance when tested before and after the change.

	Average performance (ms)
Before	3910.8 ms
After	4020.2 ms

Table 6.1: Average performance on stop-list query test set, before and after modification.

6.2 Handicap information

After the implementation of support for handicap information queries as described in section 5.2, queries for information about handicap support are understood by BusTUC. As this concerns a completely new type of query which does not use the same information or Prolog logic as the other queries, the performance of other queries was not affected enough to be reliably measured. As handicap information was not previously supported, there is no previous speed to compare to either. The output, however, is drastically changed, as demonstrated below.

Input Queries such as the following:
Can I take the bus with a wheelchair?
Is there room for a wheelchair in the bus?
I am handicapped. Can you help me get into the bus?
Is there room for a pram on the bus?

Previous output Variations of the following:
I have no information about wheelchair .
AtB has webaddress
http://www.atb.no

New output (normal)
All city buses except route 47 and 48 have ground-level entry. Other buses do not.
Buses with elevated entry are equipped with lifts.
There is room for prams and wheelchairs (max. 80x120 cm, 300 kg) in the middle of the bus.
Electrical wheelchairs are not necessarily supported.
The bus driver can assist with boarding.

New output (shortened for SMS)
Low entry: all city buses except 47 & 48. Others have lifts.

Room for wheelchairs of max. 80x120 cm, 300 kg. Driver can assist with boarding.

6.3 Final stop information

With information about buses' destinations incorporated into the answers when BusTUC is used as a route planner, the change is small on the surface, simply giving a quick piece of useful information for each bus. The change is shown in the comparison below, and Table 6.2 shows the change in performance using BusTUC's main set of test queries.

Input *I want to take the bus from Gløshaugen to Berg Studentby.*

Previous output

The station nearest to Gløshaugen is Gløshaugen Syd .
Bus 5 passes by Gløshaugen Syd at 7:48 pm , at 8:18 pm , at 8:48 pm , at 9:18 pm , at 9:48 pm and at 10:18 pm
and arrives at Berg studentby , 2 minutes later .
The hours indicate the earliest passing times.

New output

The station nearest to Gløshaugen is Gløshaugen Syd .
Bus 5 (towards Lohove) passes by Gløshaugen Syd at 9:18 pm , at 9:48 pm , at 10:18 pm , at 10:48 pm , at 11:18 pm and at 11:48 pm and arrives at Berg studentby , 2 minutes later .
The hours indicate the earliest passing times.

	Average performance (ms)
Before	78044 ms
After	81370 ms

Table 6.2: Average performance on general test set, before and after implementation of support for final stop information.

6.4 Queries

The quality of the responses is shown in Table 6.3, while Table 6.4 provides the total scores, calculated as explained in section 5.4. Using the binary scoring system, no change in the answer quality is detected, but measuring with the point system (section 5.4) reveals a 17.8 % increase of the score (from 41.5 to 48.9 points). Among the 123 queries, 37 had their answers improved, which amount to 30 % of the queries.

Label	Before	After	Binary score	Point score
Good	39	2	1	1
Better	0	37	1	1.2
Partial	3	3	1	0.5
Indirect	1	1	1	1
Fail	25	25	0	0
Timeout	2	2	0	0
Incompr.	43	43	0	0

Table 6.3: Scores for the different results.

Scoring system	Before	After
Binary score	43	43
Point score	41.5	48.9

Table 6.4: Scores for the different results.

Chapter 7

Analysis and discussion

This chapter analyzes and discusses the results in chapter 6 in the context of the earlier chapters. Section 7.1 focuses on the support of chronological bus stop lists, section 7.2 on handicap information and section 7.3 on the bus destination information, while the significance of the query suggestions is analyzed and discussed in section 7.4. In section 7.5, the topic is the general value and impact of improvements such as the ones implemented and the approach as a whole.

7.1 Chronological stop lists

With the output as shown in section 6.1, checking which stations a bus goes to will be easier, as the information will be presented in a way which is more intuitive to travelers: chronologically. There are situations in which one might want an alphabetically ordered list, but the overwhelming support this change received in the survey demonstrated that users expect such cases to be rare.

The change did not alter performance much; the average time required for the relevant test set to complete increased from 3910.8 to 4020.2 milliseconds. With 24 queries in the test set, this means merely an additional 4.56 milliseconds per query when run locally on a significantly weaker computer than the BusTUC server. More generally, it means a 2.8 % increase in the already short response time. Answering queries in the users' preferred way is easily worth such a slight increase in response time.

7.2 Handicap information

With a variety of different queries for handicap information now supported, BusTUC handicapped travelers can plan their bus travels with fewer worries. This also applies to others who might need to know what there is room for on the bus, such as those traveling with a pram. Furthermore, as the information received from AtB and provided in answer to handicap queries is slightly more specific than what is available on AtB's webpages[1], BusTUC will provide more precise information on this topic than what was previously available.

7.3 Final stop information

This implementation adds information about the direction of each bus, for each bus route suggestion returned by BusTUC. As a number of additional steps are required for BusTUC to compute this information – and with a few more needed to handle inaccuracies in the knowledge base generated from AtB's data – it is no surprise that it had a visible impact on the average performance. However, on the considerably stronger BusTUC server, the difference can be expected to be less noticeable, and the $(81370/78044 - 1) * 100 = 4.26$ % increase is worth it considering the clarity it adds to the output.

Bus travelers will with the change be given clear information on the direction in which they shall take the bus, not merely which bus line to take. This will be especially useful for those without substantial experience with the bus line they will be taking, such as visitors or simply locals traveling outside their everyday destinations. With information about the direction of the bus given as part of the answer, users can recognize their bus with immediate confidence, rather than wondering whether the bus is traveling in the right direction. This improvement might thus especially help travelers who are short on time.

Unfortunately, bus lines 36 and 66 – two versions of a bus line looping back to its beginning – do not have their "halfway" final stops marked in the data received from AtB at the moment, while the data used by the buses have these stops marked as final stops. They have therefore been excluded from this feature, but may easily be included once the data sets received better match the ones used for buses' displays. (If this does not happen, one *could* add special cases for these buses, but it would create a risk of erroneous information if the routes later were to change without the developers being specifically informed of this.)

7.4 Queries

Assigning a binary value to each query yields a success rate of $43/123 = 54.75$ %, but no change in the overall success rating of the queries. However, this is not unexpected, as all the implementations were changes to or expansions of existing functionality, rather than the addition of entirely new functionality. The exception

[1] As of June 11, 2015

to this is the support for handicap queries, but as they are of little personal importance to the average participant, they were absent from the query suggestions given in SQ5.

If one instead uses the point assignment system described in section 5.4, the success score increases by 17.8 %, from 41.5 to 48.9. Perhaps more usefully, we see that 37 of the 123 queries were labeled as “better” than they were before the changes. This is 30 %, indicating that the implemented functionality will improve the information provided in answer to a substantial portion of BusTUC’s usual queries.

Nevertheless, it is worth noting that as the case of handicap information demonstrates, a query’s importance does not necessarily equal its commonness. The point assignment method as a measurement of BusTUC’s progress therefore suffers from many of the same faults as the binary method, despite its superior precision. And with no previous measurements using this system, there is currently little to compare it to, making the percentage gain less useful for measuring the progress. Though this could be improved by consistent application of the same method of measurement to later expansions, it depends on the completeness of the test set: Later improvements not covered by the tests would require an expansion of the test set, which again would cause re-calculation of earlier scores to be required for consistent scores.

The participants’ possibly excessive creativity and the subjectivity of the query transformation, as discussed in section 4.3.3, also diminish the importance of this method of measurement.

Therefore, though the point system serves as confirmation that progress has been made, it may be better to apply a more flexible method of measurement in the future. For example, one might use the average daily success rate of the system over time. This would delay the measurement of improvements until some time after the new version of BusTUC is activated, but the flexibility may be worth it when measuring the progress, even though cases like handicap information would still be poorly measured.

Despite the above, the queries have proven useful, as their purpose was not only to measure the progress. They provided valuable insight into the expected needs of bus travelers, and where the creativity of a participant exceeds that of most actual uses of BusTUC, one might discover areas of improvement that the system’s more habit-bound users are unlikely to test. Because of this, eventual later expansions of BusTUC following this survey-first approach should not leave out the query gathering entirely.

7.5 Approach and usefulness

There have been other BusTUC projects using questionnaires to gather the opinions of users [Engell, 2012; Wollamo, 2013; Nakken and Nascimento Bakke, 2015; Jacobsson, 2015], but those surveys have focused on the user testing of the result, rather than on establishing what is the best direction in which to *seek* results to improve the services offered to users. For example, Wollamo also wrote about the usefulness of BusTUC to users, but the development was based on the assumption that users agreed with him on what would be useful; in this respect, the approach used in this project differs significantly from earlier BusTUC projects, as it measures the improvements' perceived utility before selecting them.

Furthermore: as the literature review revealed in section 2.2.2, a considerable number of applications (of varying degrees of completeness) include BusTUC's functionality as a part of the service they provide. Most of these are to some extent competitors vying for Trondheim's bus passengers. This project's approach, however, is to improve the usefulness of BusTUC itself, based on an assessment of what people expect and desire. The implementations detailed in this thesis therefore become available to end users regardless of which of the existing applications of BusTUC they use, thus providing useful improvements of the services available to users without having to struggle as a competitor of the existing applications.

Highly specialized applications, such as ones designed specifically to aid the elderly or the visually impaired, would face a different competition and might do very well with their target audience without challenging the most popular applications. The speech-related technology of TaleTUC (section 2.2.2) could be especially useful in such a setting. However, more general bus information applications would normally have to compete against the larger ones. Thus, for the general development, BusTUC is better served by a focus on improving and expanding its core functionality.

Furthermore, with every improvement of this type, the competition that exists between the various applications changes slightly in the favor of those that include BusTUC's natural language information service. This effect is further strengthened by how the implementations are grounded in research on what bus travelers wish for.

This approach to the development of BusTUC is therefore a more efficient way of improving the service BusTUC provides. After all, the usefulness of the system only counts if it actually reaches the users, and having to compete against already popular applications is a major obstacle – an obstacle that is not always necessary.

Chapter 8

Conclusion and future work

This chapter concludes the research and outlines future work. Section 8.1 summarizes the conclusions reached in the earlier chapters, with a focus on the research questions

8.1 Research questions and goals

The goals defined in section 1.3 deal with the discovery and implementation of useful features. They have been reached, and included in the summarization of the answer to research question RQ1 below.

RQ1 *What additional information is it useful for NLP-based transportation query systems like BusTUC to support?*

This question was answered by the survey and the analysis that followed. Several expansions of BusTUC's functionality were suggested, ranked by participants and analyzed based on the results, leading to the implementation of some of them. Additional potential improvements were identified with the help of participants' query suggestions and feedback. These improvements are summarized below, showing the additional functionality that bus travelers want services like BusTUC to support.

Chronological stop lists: This improvement received overwhelming support from the survey participants, gathering far more votes than any other. BusTUC was therefore modified to generate chronological bus stop lists for the relevant queries.

Handicap information: The responses and analysis confirmed the importance of this functionality. Information about wheelchairs and what buses have room for was added to BusTUC. New vocabulary and query analysis logic will let the next version of BusTUC support such queries as soon as it goes live.

Final stops: Final stop information was shown to be highly valued by the survey participants. The necessary logic was added for BusTUC to conclude which end stop each suggested bus is heading towards, and include this information as part of the suggestion.

Time variables: This feature received considerable support from survey participants, and was concluded to be a useful extension of functionality if implemented with a long-term, low-maintenance, scalable solution.

Frequency: Frequency information was identified as a possible improvement from the queries and feedback received in the survey. Future expansions of BusTUC functionality should consider including this.

Airport bus: Adding support for information about the main buses to the closest airport would be of use to users of BusTUC, and should be strongly considered in later work on the system. It is similar to the service already provided by BusTUC, but some more query types should also be supported, such as queries about airport bus prices. An agreement would have to be established with the companies in question to ensure updated information, and a program constructed for conversion of this information to BusTUC's logical format.

Additionally, the query testing has proven that there are still reasonable queries that BusTUC misunderstands. Improving this would reduce the frequency with which users experience errors while using the system.

RQ2 *Would BusTUC's usefulness to its users best be increased through a survey-first approach to expanding its functionality?*

While the exact procedure described by this thesis can not be expected to fit every future BusTUC project, its general approach is worth keeping. The work described in chapter 5 is firmly grounded in an analysis of the opinions bus passengers have on the subject, and applies to all services using BusTUC. With this approach to developing BusTUC further, one ensures that the results both reach the users and serve them well. It will therefore in many cases be a more efficient way of helping users than the construction of more mobile applications, which would compete against established applications with the very same BusTUC in their service. Though it was also concluded that the progress is not easily measured formally and consistently, the involvement of bus travelers in the preparation for each development functions as a safety net, ensuring that the development moves BusTUC noticeably forwards.

8.2 Future Work

This section explores the future work that emerges from the content of the report.

8.2.1 Improved comprehension

The queries showed that there are still many queries BusTUC (or more specifically, TUC) does not comprehend. Some of these are completely reasonable queries, while others are not queries one would usually expect BusTUC to receive. The first group should be supported, and ideally the second group should also be comprehended (even if it just means giving a negative answer). Looking into the current lacks and weaknesses of BusTUC's sentence analysis, and correcting it, could greatly improve BusTUC, especially for users not yet used to its limitations.

8.2.2 Further improvements

Not every improvement discussed and analyzed was implemented, and there are doubtlessly other potential improvements not yet uncovered or analyzed. Future work could focus on continuing the work and method of this thesis to further improve the service BusTUC offers to its users, regardless of their application of choice.

8.2.3 Real-time information

AtB expects to start using a new and improved system for real-time bus delay information in the near future. Once this is up and running, it would be beneficial to use the new system to fetch real-time delay information which can be provided as part of BusTUC's service. It might also be beneficial to stick to a survey first approach to this development too, as this would help the developers ensure that the information is provided in the ways the users find useful, rather than risk implementing it for queries they are unlikely to use.

8.2.4 Dialog

In combination with some or all of the planned implementations above, the user-friendliness of BusTUC can be expected to improve greatly if one were to add dialog support. As chapter 2 revealed, there has been a considerable amount of work done on both speech and dialog for BusTUC. Implementing dialog support into BusTUC itself would allow for a much more user-friendly interaction with BusTUC. There has already been an attempt at this, but it is not yet fully integrated. If this were to be finished, and then adapted to help where dialog is most needed, it would have great impact on the user experience. For example, for an ambiguous query like “Gløs til Berg”, where “Berg” can refer to several places, the user could just tell BusTUC which “Berg” was meant, rather than re-submitting the entire query with this information.

Bibliography

Amble, T. (2000). BusTUC: A Natural Language Bus Route Oracle. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pages 1–6, Stroudsburg, PA, USA. Association for Computational Linguistics.

Amble, T. (2004). The Understanding Computer. *Natural Language Understanding in Practice. Link: www.idi.ntnu.no/emner/tdt4275/NOTES/tulip.pdf.*

Andersstuen, R. and Engell, T. (2011). MultiBRIS: A Multiple-platform approach to the Ultimate Bus Route Information System for Mobile Devices. Technical report, Autumn project, Department of Computer and Information Science, Norwegian University of Science and Technology.

Andersstuen, R. and Marcussen, C. J. (2012). *TaleTUC : Automatic Speech Recognition for a Bus Route Information System.* MSc Thesis, Department of Computer and Information Science, Norwegian University of Science and Technology.

Bach, E. and Horn, G. M. (1976). Remarks on "Conditions on Transformations". *Linguistic Inquiry*, 7(2):265–299.

Blackburn, P. and Bos, J. (2003). Representation and inference for natural language. *A First Course in Computational Semantics*, pages iii–vi, 1–4, 14–27.

Bratseth, J. S. (1997). *Bustuc - A Natural Language Bus Traffic Information System.* MSc Thesis, Department of Computer and Information Science, Norwegian University of Science and Technology.

Colmerauer, A. (1978). Metamorphosis grammars. In Bolc, L., editor, *Natural Language Communication with Computers*, pages 133–188. Springer Berlin Heidelberg.

Eliassen, L. M., Sætre, R., Gambäck, B., and Marcussen, C. J. (2012). Context-Awareness and Real-Time Information in an Intelligent Smartphone Application.

Engell, R. A. T. B., Sætre, R., and Gambäck, B. (2012). A Multiple Platform Approach to Building a Bus Route Information System for Mobile Devices. In *12th International Conference on Innovative Internet Community Systems*, page 71. Citeseer.

Engell, T. B. (2012). *TaleTUC: Text-to-Speech and Other Enhancements to Existing Bus Route Information Systems*. MSc Thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, (Telektronikk, Volume 99, No. 2 2003, ISSN 0085-7130).

Geller, T. (2012). Talking to Machines. *Communications of the ACM*, 55(4):14–16.

Hartvigsen, O., Harborg, E., Amble, T., and Johnsen, M. H. (2007). Marvina–A Norwegian Speech-Centric, Multimodal Visitors' Guide. In *NODALIDA 2007 Proceedings*.

Jacobsson, E. (2015). *OsloTUC - Natural Language Bus Oracle for a new City*. MSc Thesis, Department of Computer and Information Science, Norwegian University of Science and Technology.

Johnsen, M. H., Amble, T., and Harborg, E. (2003). A Norwegian Spoken Dialogue System for Bus Travel Information. *Spoken Language Technology in Telecommunications*, pages 125–131.

Kelley, K., Clark, B., Brown, V., and Sitzia, J. (2003). Good practice in the conduct and reporting of survey research. *International Journal for Quality in Health Care*, 15(3):261–266.

Manjunath, K. U. K. (2014). *Location Based Context-Aware Systems*. MSc Thesis, University of Birmingham.

Marcussen, C. and Eliassen, L. M. (2011). TABuss: An Intelligent Smartphone Application. Fall project report, Department of Computer and Information Science, Norwegian University of Science and Technology.

Nakken, O. K. and Nascimento Bakke, S. (2015). *TransitVision: Approximating Vehicle Locations Using SIRI-SM Real-Time Data*. Master Thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Pereira, F. (1981). Extraposition Grammars. *Computational Linguistics*, 7(4):243–256.

Pereira, F. C. and Warren, D. H. (1980). Definite clause grammars for language analysis—a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, 13(3):231–278.

Pogue, D. (2012). Siri, Why Aren't You Smarter? *Scientific American*, 307(2):33–33.

Pun-Cheng, L. (2012). An Interactive Web-Based Public Transport Enquiry System With Real-Time Optimal Route Computation. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):983–988.

Raaum, M. (2010). *An intelligent smartphone application.* MSc Thesis, Department of Computer and Information Science, Norwegian University of Science and Technology.

Reiter, R. (1978). On Closed World Data Bases. In *Logic and Data Bases*, pages 55–76. Springer.

Strik, H., Russel, A., Heuvel, H. V. D., Cucchiarini, C., and Boves, L. (1997). A spoken dialog system for the Dutch public transport information service. *International Journal of Speech Technology*, 2(2):121–131.

Wollamo, M. Q. (2013). *Improving an Existing Natural Language Bus Route Information System : Adding Support for Geographic Expansion and Visualization in Maps.* MSc Thesis, Department of Computer and Information Science, Norwegian University of Science and Technology.

Appendix A

Survey questions

The complete survey described in chapter 3 is given here, in the same words as the version received by the participants. The headline for the survey on *Reddit SampleSize* (see section 3.1) had to conform to the rules for such headlines, and was therefore chosen to be *[Academic] Quick survey about public transportation (Everyone).* On Qualtrics (see section 3.2), the headline was *Quick survey about public transportation information.*

One question requires at least three options to be selected. Where this is not mentioned participants selected at most one option.

Introduction *Quick introduction:*

> We have a system for bus information, which understands normal sentences. If asked a question about local bus routes in natural language (2 are supported), it will interpret it and provide generate an answer (also in natural language). It provides information for bus travel in a city of less than 200 000 people. There is no subway system in this city.

Examples *Examples:*

> Q1: "When does bus X leave place_A?"
>
> A1: "Bus X passes by Place_A at 0:00 am and arrives at Default_Destination at 0:32 am. The hours indicate the earliest passing times."
>
> Q2: "From Place_A to Place_B"
>
> A2: "The station nearest to Place_A is Station_A. Bus X goes from Station_A at 11:46 pm to Station_C at 11:52 pm and bus Y goes from Station_C at 0:00 am to Place_B at 0:09 am. The hours indicate the earliest passing times."

SQ1 *Which of the following best describes you?*

1. Student

2. Working
3. Retired
4. Unemployed / other

SQ2 *How many times do you travel by bus during an average week?*

A rough estimate is fine.

1. I do not use any kind of public transportation
2. I never travel by bus (but sometimes other public transportation)
3. Less than 1 (on average)
4. 1-4
5. 5-8
6. 9-12
7. More than 12

SQ3 *Where do you travel by bus the most?*

1. In a city
2. Not within a city
3. I don't travel by bus

SQ4 *How do you get information about bus routes / other public transportation?*

1. Information at the bus(/other) stop
2. Information on paper
3. Application on mobile device
4. Websites
5. I don't seek out information
6. Other: [Input field]

SQ5 *Please think of a few questions for the information system to interpret and answer, as you might need it to when traveling by bus.*

They should be about public transportation, and ones you think you might use.

Place names: A/B/etc can be used as placeholders for place names.

No GPS-based queries: As such things are handled by applications using this system, not the core query system, GPS queries such as ("From *here* to Place_B") are not relevant here.

[Text input field]

SQ6 *What possible additions do you think seem the most useful for this type of system?*

Please select 3 or more.

(As you probably do not have experience with our system, a "gut feeling" of usefulness is fine.)

1. Chronological list of stops for a bus (currently alphabetical)
2. Information about the final stop/end stop of the bus
3. Time variables, such as opening hour information (if something else, specify below) (ex.: "<place> to <shopping center> at opening hour")
4. Information about the trains out of the city (not subway!)
5. Support for the buses to/from airport (not currently supported)
6. Information about handicap accessibility
7. Something else is important to support (specify below)

SQ7 *Any other notes on information you feel is important/useful for such a system to support?*

(If you chose "something else" above, please elaborate here)

[Text input field]

Appendix B

Queries from survey

Table B.1 contains the query suggestions given in answer to the fifth survey question. There were 123 suggested queries. The first column contains the queries as they were received. The second column contains the queries after translation and manual replacement of placeholders, so that they can be used in the Norwegian BusTUC (for example changing “from A to B” to “fra Nardo til Moholt”).

The marks for different types of results were as explained below:

Good An appropriate result was returned.

Better An appropriate and improved result was returned.

Partial The query was not answered completely, but some of the desired information was provided. (Example: frequency queries, answered with the next five departures)

Indirect The query was not answered literally, but the *information need* behind it was most probably met.

Fail BusTUC did not provide the requested information.

Timeout The system timed out while analyzing the sentence.

Incompr. TUC failed to transform the sentence to TQL and marked it as incomprehensible.

(**Irrelevant**) The question is in some way irrelevant.

The replacements used are as follows:

Moholt Primary area. Also used for "my house" (and similar) where GPS is not clearly mentioned (as one can expect people to know their home's closest bus stop well enough for it to be synonymous with "my home" in a bus context.

Samf Secondary area (abbreviation of "Studentersamfundet", understood by BusTUC).

Gløs, Lade Third area names (Gløs: abbreviation of "Gløshaugen").

Høgskoleringen 2 Primary address (used when a participant clearly wishes to use an exact address in the query).

Bus 5 Replaces bus numbers. (If multiple, bus 22 used.)

0900 Replaces time placeholders in queries (for example "[DEPARTURE TIME]").

Some of the suggestions have been modified more than others. For example, some were not structured as a question for the system, but rather as a suggestion of what BusTUC should be able to answer. These have their original query descriptions listed in *italics*. Common spelling mistakes (for example, "what's"→"whats") have been left intact, while a few that it is reasonable for BusTUC to reject (for example, "bus"→"bugs") have been corrected, as the user would most likely correct it and re-submit the query without changing its sentence structure. Furthermore, suggestions were not assumed to rely on GPS merely at the mention of a word like "here"; this assumption was made only when it was very clear that the user expects the system to know their current location.

Table B.1: Query suggestions

Unedited query	**Edited query**	**Before**	**After**
When will bugs 545 get here?	When will bus 5 get to Moholt?	Good	Better
How often does a bus go from A to B?	How often does a bus go from Moholt to Samf?	Partial	Partial
And B back to A?	How often does a bus go back from Samf to Moholt?	Fail	Fail
How long does it take to get from A to B?	How long does it take to get from Moholt to Samf?	Incompr.	Incompr.
How late does the bus run from A to B?	How late does the bus run from Moholt to Samf?	Fail	Fail
What is the stop before Station_A on the Bus X line?	What is the stop before Moholt on bus 5 towards Samf?	Incompr.	Incompr.

How often does Bus X run at Y time of day?	How often does bus 5 run at 0900?	Fail	Fail
How often does Bus X stop at Station_A at Y time of day?	How often does bus 5 stop at Moholt at 0900?	Fail	Fail
Does Station_A have benches/roof?	Does Moholt have benches/roof?	Incompr.	Incompr.
arrival time 2pm at A, what time do I have to get on the bus?	Arrival time 2PM at Moholt, what time do I have to get on the bus?	Timeout	Timeout
What's the last bus from origin X to destination Y on weekdays?	What's the last bus from Moholt to Samf on weekdays?	Incompr.	Incompr.
What's the first bus from origin x to destination Y on weekdays?	What's the first bus from Moholt to Samf on weekdays?	Incompr.	Incompr.
How do I pay the fare?	How do I pay the fare?	Fail	Fail
Is Bus X on time?	Is bus 5 from Moholt to Samf on time?	Fail	Fail
How long does Route X take?	How long does bus 5 from Moholt to Samf take?	Indirect	Indirect
When is the latest bus on route X?	When is the latest bus on route 5?	Fail	Fail
What time is the bus scheduled to arrive at Place A?	What time is the bus scheduled to arrive at Moholt?	Incompr.	Incompr.
What is the travel time between Place A and Place B?	What is the travel time between Moholt and Samf?	Incompr.	Incompr.
Will a bus traveling from Place A be available at 8PM?	Will a bus traveling from Moholt be available at 8PM?	Good	Better
How far away is the next bus right now?	How far away is the next bus from Moholt to Samf right now?	Incompr.	Incompr.
What is the estimated time of arrival in current traffic conditions?	What is the estimated time of arrival in the current traffic conditions for bus 5 from Moholt?	Incompr.	Incompr.
When is the next bus?	When is the next bus from Moholt to Samf?	Good	Better

When does this bus stop running?	When does bus 5 stop running?	Fail	Fail
How often does this bus come?	How often does bus 5 come?	Fail	Fail
Where should I get off to transfer to line X?	Where should I get off to transfer to bus 5?	Fail	Fail
What's the closest bus line to ?	What's the closest bus line to Moholt?	Incompr.	Incompr.
Whats the fastest way from A to B within the next $Time period	Whats the fastest way from Moholt to Samf within the next half hour?	Good	Better
Whats the soonest way from A to B	Whats the soonest way from Moholt to Samf	Incompr.	Incompr.
Next bus from Place_A to Place_B	Next bus from Moholt to Samf	Good	Better
Price for Place_A to Place_B	Price for Moholt to Samf	Incompr.	Incompr.
Place_A to Place_B at Time_Here	Moholt to Samf at 0900	Good	Better
Can I get to place_A from place_B?	Can I get to Moholt from Samf?	Good	Better
Is it faster to get to place_A from place_B or place_C?	Is it faster to get to Moholt from Samf or Lade?	Fail	Fail
Is place_A closer to place_B or place_C?	Is Moholt closer to Samf or Lade?	Incompr.	Incompr.
When does bus X get to place_A?	When does bus 5 get to Moholt?	Good	Better
How often does bus X leave place_A?	How often does bus 5 leave Moholt?	Partial	Partial
How long is bus X at place_A?	How long is bus 5 at Moholt?	Incompr.	Incompr.
Does bus X or bus Y get me from place_A to place_B sooner?	Does bus 5 or bus 36 get me from Moholt to Samf sooner?	Incompr.	Incompr.
Which bus gets me from place_A to place_B faster?	Which bus gets me from Moholt to Samf faster?	Incompr.	Incompr.
Is there a bus that visits place_A, place_B, ..., and place_Z before time_T?	Is there a bus that visits Moholt, Samf and Lade before 0900?	Timeout	Timeout

If Bus x arrives at place_A 5 minutes late , will i still be able to catch Bus y at place_A	If Bus 5 arrives at Moholt 5 minutes late, will I still be able to catch Bus 22 at Moholt?	Incompr.	Incompr.
What bus do I take to get to B?	What bus do I take to get to Moholt?	Good	Better
What bus routes go from here to B?	What bus routes go from Moholt to Samf?	Good	Better
How long will it take to get from A to B?	How long will it take to get from Moholt to Samf?	Fail	Fail
Is Bus A on time?	Is bus 5 from Moholt to Samf on time?	Fail	Fail
how long will it take me to get to place_a?	how long will it take me to get to Moholt?	Incompr.	Incompr.
what is the fastest bus that will take me to place_b?	what is the fastest bus that will take me to Samf?	Good	Better
How much do I pay for a trip from a to b?	How much do I pay for a trip from Moholt to Samf?	Good	Good
Can I pay Cash or Credit?	Can I pay Cash or Credit?	Incompr.	Incompr.
Do I need exact change?	Do I need exact change?	Incompr.	Incompr.
Is there an ATM near by?	Is there an ATM near by Moholt station?	Incompr.	Incompr.
Where is the nearest mall/shopping center/food stop/bus stop?	Where is the nearest shopping center?	Incompr.	Incompr.
What's the cheapest fare?	What's the cheapest fare?	Incompr.	Incompr.
How much delay does my tram/bus have?	How much delay does the next bus 5 from Moholt to Samf have?	Incompr.	Incompr.
Last bus X to reach place A by hh:mm.	Last bus 5 to reach Samf by 09:00.	Incompr.	Incompr.
What stops will this bus make?	What stops will bus 5 make?	Incompr.	Incompr.
What time does the bus arrive at stop A?	What time does bus 5 arrive at Moholt?	Good	Better
Will we pass through stop B?	Will bus 5 pass through Moholt?	Fail	Fail

Which side of the neighborhood does this bus pass?	Which side of the Moholt neighborhood does bus 5 pass?	Incompr.	Incompr.
Where is the bus stop for the same route, in the opposite direction?	Where is the Moholt bus stop for bus 5, out of town?	Fail	Fail
At what time is the last bus leaving from this stop?	At what time is the last bus leaving from Moholt?	Incompr.	Incompr.
Where should I change to bus [number or for destination]?	Where should I change to bus 5 for Samf from Lade?	Fail	Fail
Is this the bus for the airport?	Is bus 5 the bus for the airport?	Fail	Fail
Can I transfer to another bus for free?	Can I transfer to another bus for free?	Incompr.	Incompr.
When does the last bus leave [STOP]?	When does the last bus leave Moholt?	Good	Better
When does the last bus arrive at [STOP]?	When does the last bus arrive at Moholt?	Good	Better
How often do buses depart [STOP]?	How often do buses depart Moholt?	Partial	Partial
If I miss [DEPARTURE TIME], what alternate bus could I take?	If I miss bus 5 from Moholt at 0902, what alternate bus could I take?	Incompr.	Incompr.
When will I arrive at A?	When will I arrive at Moholt with bus 5?	Good	Better
How long will it take me to get to A?	How long will it take me to get to Moholt from Samf?	Incompr.	Incompr.
Is my bus on time?	Is bus 5 from Moholt on time?	Fail	Fail
When will my bus arrive?	When will bus 5 arrive at Moholt?	Good	Better
Is my bus late?	Is bus 5 from Moholt late?	Fail	Fail
When is the next bus from Place_A to Place_B?	When is the next bus from Moholt to Samf?	Good	Better
How much is the bus from Place_A to Place_B?	How much is the bus from Moholt to Samf?	Fail	Fail
How frequent are busses between Place_a and Place_B?	How frequent are busses between Moholt and Samf?	Incompr.	Incompr.

When is the first/last	When is the first bus to Moholt?	Good	Better
When is the first/last	When is the last bus to Moholt?	Good	Better
Where is the nearest bus stop to Location_A for me to travel to Place_B?	Where is the nearest bus stop to Høyskoleringen 2 for me to travel to Samf?	Incompr.	Incompr.
bus from Place_A to Place_B?	bus from Moholt to Samf?	Good	Better
Fastest route between A and B?	Fastest route between Moholt and Samf?	Incompr.	Incompr.
Which bus services run between A and B?	Which bus services run between Moholt and Samf?	Good	Better
Cheapest route between A and B?	Cheapest route between Moholt and Samf?	Incompr.	Incompr.
Is the bus running late?	Is the bus running late for Moholt?	Fail	Fail
What time does the bus come to placeholder?	What time does the bus come to Moholt?	Good	Better
Is this particular bus equiped with a bicycle rack?	Is bus 5 equiped with a bicycle rack?	Incompr.	Incompr.
How much does the bus cost?	How much does the bus cost?	Good	Good
Do I need exact change?	Do I need exact change?	Incompr.	Incompr.
How do I get from A to B?	How do I get from Moholt to Samf?	Good	Better
How do I get from here to home?	How do I get from Samf to Moholt?	Good	Better
All I care about is when the bus will arrive.	When will bus 5 arrive at Moholt?	Good	Better
How long until bus x?	How long until bus 5 to Moholt?	Incompr.	Incompr.
How much for a ticket?	How much for a ticket?	Incompr.	Incompr.
What time does the last bus leave?	What time does the last bus leave Moholt?	Good	Better
How do I get to A from B?	How do I get to Moholt from Samf?	Good	Better
When is the next bus to A from B?	When is the next bus to Moholt from Samf?	Good	Better

How often to buses leave A to get to B?	How often to buses leave Moholt to get to Samf?	Incompr.	Incompr.
Where do I catch bus X?	Where do I catch bus 5 to Moholt?	Good	Better
What times do buses leave A on the weekends?	What times do buses leave Moholt on the weekends?	Fail	Fail
When is the next bus over place B leaving from place A?	When is the next bus over Moholt leaving from Samf?	Good	Better
When is the last bus arriving before 10:30 at place B going to leave place A?	When is the last bus arriving before 10:30 at place B going to leave place A?	Incompr.	Incompr.
Why do you not stop when you see me running after your bus?	Why do you not stop when you see me running after your bus?	Fail	Fail
When is the next bus $x due at $location.	When is the next bus 5 due at Moholt.	Incompr.	Incompr.
What bus do I get from $location_a to get to $location_b by $time.	What bus do I get from Moholt to get to Samf by 0900.	Good	Better
What stations does bus A stop at?	What stations does bus 5 stop at?	Good	Better
What is the schedule for bus stop B?	What is the schedule for Moholt?	Good	Better
When will the next bus arrive at bus stop B?	When will the next bus arrive at Moholt?	Good	Better
How long will it take to get to A?	How long will it take to get to Moholt?	Fail	Fail
How long is the trip from Place_A to Place _B?	How long is the trip from Moholt to Samf?	Fail	Fail
How often does a bus stop at Place_A?	How often does a bus stop at Moholt?	Incompr.	Incompr.
Which bus lines go to Place_A?	Which bus lines go to Moholt?	Good	Better
When does the next bus come to Place_A?	When does the next bus come to Moholt?	Good	Better
When will Bus x arrive at Station_A around 7 PM	When will Bus 5 arrive at Moholt around 7 PM	Good	Better

Appendix C

Answers to the first survey

This appendix contains one of the answers to the pre-study's first survey. It was replaced by the later, improved versions because of its few responses, but is mentioned for comparison in section 3.3.1. As the cause of its relevance is that many of its participants were bus passengers in Trondheim, Figure C.1 is included to document this.

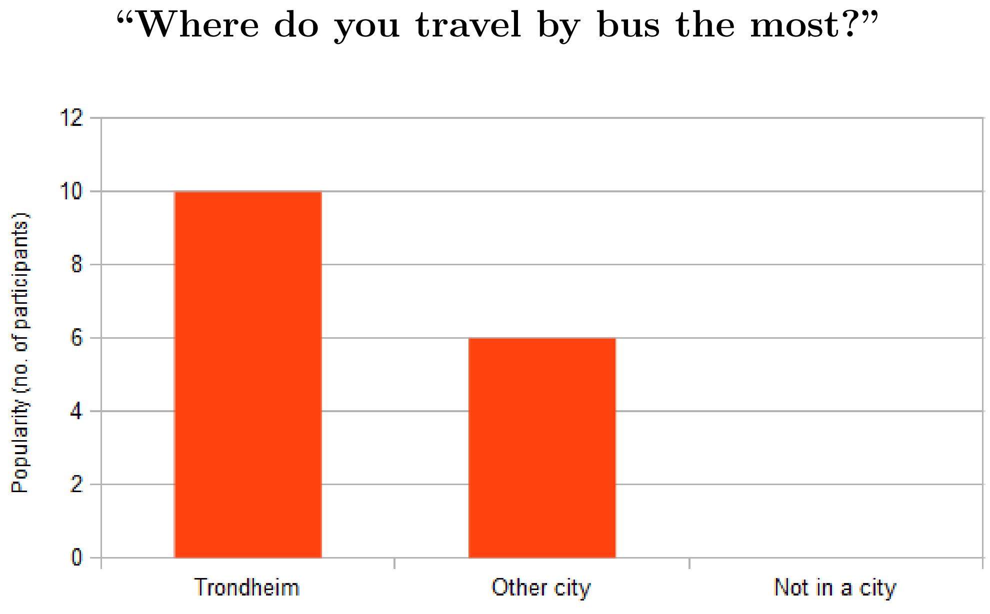

Figure C.1: Answers to the question about location in the pre-study's first survey.