



UPPSALA
UNIVERSITET

IT 15058

Examensarbete 30 hp
Juli 2015

Phonetic Analysis and Searching with Google Glass API

Vahid Abbasi

Institutionen för informationsteknologi
Department of Information Technology



UPPSALA
UNIVERSITET

**Teknisk- naturvetenskaplig fakultet
UTH-enheten**

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
<http://www.teknat.uu.se/student>

Abstract

Phonetic Analysis and Searching with Google Glass API

Vahid Abbasi

This project utilizes speech recognition Application Program Interface (API) together with phonetic algorithms to search Stockholm's restaurant names via Google Glass with higher precision.

This project considers the ability of phonetic algorithms and N-gram analyzer to retrieve the word and how it can be combined with automatic speech recognition to find the correct match. Significantly, the combination of these algorithms and the Google Glass limitation, e.g. its smallscreen, makes using a phonnetic filtering algorithm very helpful in getting better results.

Handledare: Birger Rydback
Ämnesgranskare: Matteo Magnani
Examinator: Edit Ngai
IT 15058
Tryckt av: Reprocentralen ITC

Acknowledgments

This is a master thesis submitted in Computer science to Department of Information Technology, Uppsala University, Uppsala, Sweden.

I would like to express my deepest gratitude to my supervisor Birger Rydback at Findwise for his patience in supporting continuously and generously guiding me with this project.

I would like to appreciate professor Matteo Magnani for reviewing my master thesis and offering valuable suggestions and comments.

Special thanks to Simon Stenström for initiating the work and providing key insights at Findwise.

I would like to thank Uppsala University and Findwise AB Company for providing me all the devices and resources necessary for the project.

Sincere thanks to my wife and my incredible parents who gave me support throughout the thesis work.

Contents

| | | |
|----------|-------------------------------------------------------------|----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | The Aim of This Work | 2 |
| 1.3 | Possible Solution | 2 |
| 1.4 | Overview | 2 |
| 2 | Background | 5 |
| 2.1 | Automatic Speech Recognition | 5 |
| 2.1.1 | Acoustic Phonetic Approach | 6 |
| 2.1.2 | Pattern Recognition Approach | 7 |
| 2.1.3 | Artificial Intelligence Approach (Knowledge Based Approach) | 7 |
| 2.2 | Search by Voice Efforts at Google | 7 |
| 2.3 | Difficulties with Automatic Speech Recognition | 8 |
| 2.3.1 | Human | 9 |
| 2.3.2 | Technology | 9 |
| 2.3.3 | Language | 9 |
| 2.4 | Metrics | 9 |
| 2.4.1 | Word Error Rate | 10 |
| 2.4.2 | Quality | 10 |
| 2.4.3 | Out of Vocabulary Rate | 11 |
| 2.4.4 | Latency | 11 |
| 2.5 | Phonetic algorithm | 11 |
| 2.5.1 | Soundex | 11 |
| 2.5.2 | Daitch-Mokotoff Soundex System | 12 |
| 2.5.3 | Metaphone | 12 |
| 2.5.4 | Double Metaphone | 13 |
| 2.5.5 | Metaphone 3 | 13 |
| 2.5.6 | Beider Morse Phonetic Matching | 13 |
| 2.6 | Google Glass | 14 |
| 2.6.1 | Google Glass Principles | 14 |
| 2.6.2 | User Interface | 14 |
| 2.6.3 | Technology Specs | 14 |
| 2.6.4 | Glass Development Kit | 15 |

| | | |
|----------|------------------------------------------------------|-----------|
| 2.6.5 | The Mirror API | 15 |
| 2.7 | Elastic-search | 15 |
| 2.7.1 | NGrams Tokenizer | 16 |
| 2.7.2 | Phonetic Filters | 17 |
| 3 | Related Works | 19 |
| 4 | Implementation | 21 |
| 4.1 | Motivation of selected Phonetic Algorithms | 21 |
| 4.2 | Motivation of Showing 6 Rows in Each Page | 22 |
| 4.2.1 | Google Glass Limitation | 22 |
| 4.2.2 | Users Behavior in Searching Areas | 23 |
| 4.2.3 | Conclusion | 26 |
| 4.3 | Data Set | 26 |
| 4.4 | Norconex | 27 |
| 4.4.1 | Importer Configuration Options | 27 |
| 4.4.2 | Committer Configuration Options | 27 |
| 4.4.3 | More Options | 27 |
| 4.5 | Elasticsearch | 27 |
| 4.6 | Demo Of Application | 28 |
| 5 | Evaluation | 31 |
| 5.1 | Test data | 31 |
| 5.2 | Quantity Tests | 32 |
| 5.3 | Result | 32 |
| 5.4 | Best Algorithm For Google Glass | 33 |
| 5.4.1 | Improving Precision and F-measure | 36 |
| 5.5 | Conclusion | 36 |
| 6 | Discussion and Conclusions | 39 |
| 6.1 | Development Limitations | 40 |
| 6.2 | Further Research | 40 |
| | List of Tables | 41 |
| | List of Figures | 41 |
| | Bibliography | 43 |
| | Appendices | 45 |
| A | List of Acronyms | 47 |
| B | Restaurant Names for Testing | 49 |
| C | Restaurant Names and Their Phonetic Sounds | 51 |

Chapter 1

Introduction

This is a Master thesis at the Department of Computer Science, Uppsala University done by Vahid Abbasi from January 2015 to June 2015. The thesis is in collaboration with FINDWISE AB company in Stockholm.

1.1 Motivation

Speech recognition is a significant invention and very applicable. Not only is a voice the most basic way for humans to communicate, but also because it is an efficient interface to digital devices. Google Glass is a novel device nowadays that uses speech as the main way of input. Although its current voice recognition capability is rather good, it is not perfect.

Any search in Google Glass is mainly controlled and activated via a voice, so that its functionality mostly depends on how well voice commands are detected; which means that when the algorithm does not detect a voice correctly, it does not perform current and further steps correctly either. Most failed detection's cases are where user calls names that are not from a native-language vocabulary (or most frequently used words), such as proper names (e.g. name of restaurants), because Google Glass might not find any words corresponding to the detected phoneme at the pattern matching steps of detection. It appears that one of the relevant searches request or applications for Google Glass would be looking for a place and having the relevant information about it.

In addition to this, Google Glass screen is very small and the algorithm filters out the candidate results rather strictly to give only one hit as the best, but it might not be the desired word. Considering restaurant name, the first candidate word is not always the best. Therefore, showing more hits could be useful so if the user was involved in the filtering process and got a few top relevant results, the correct answer would be revealed in fewer tries. Accordingly, using the current small screen more efficiently is worth considering with respect to all limitations and specifications for this technology, this kind of device limitation should be considered.

1.2 The Aim of This Work

In this thesis, voice recognition will be integrated into a search application by using Google Glass and focusing on matching the result to the relevant cases (possible restaurant names) and increasing user satisfaction. The target application is a restaurant search application to gather information and directions to a restaurant within Stockholm city, asked via the Google Glass voice.

Due to the fact that improving accuracy in the voice to text conversion is difficult and it may not be feasible for this thesis project, it would be interesting to see how a general solution for this application could lead to some improvements.

1.3 Possible Solution

The main goal of this thesis is to extend the current Google Glass voice detection algorithm and collect results and then check them against a relevant vocabulary database, in this case, the names of restaurants in Stockholm. As the set of restaurant names will have a finite number of options, it might be possible to specify the vocabulary and limit the results to these words, in the hope of achieving more accurate answers. This means that after getting the output from voice recognizer, the output is filtered using search index and all potential hits are shown to the user.

Phonetic algorithms are successfully used techniques in encoding words by their pronunciation. They are utilised in this project to encode the current detected names by Google Glass and all restaurant names indexed in a search index, which might be helpful to find more similarities by comparing phonetics than syntax.

Later, different phonetic algorithms are analyzed and compared (e.g. Soundex, Double Metaphone, etc.) in order to see which algorithm could work better in this application context. Eventually, the user is able to either select the desired name or retry the voice detection part. Then for the selected restaurant, some information (such as name, address and reviewed rank) and the possibility to navigate to the place on a map are provided.

1.4 Overview

In the second chapter, the background is reviewed and some well-known algorithms, phonetic algorithms, methods for evaluating the voice detection algorithm, and challenges of voice detection algorithms are explained. A brief description of Google Glass and Elastic-search is also introduced in this chapter. Readers who are already familiar with these concepts could skip chapter 2.

In chapter three, other related work is discussed along with their advantages and weak points.

Chapter four explains the implementation details of this project. The algorithm, setup details of tools and environment, the data source, the motivation of the selected algorithms and implementation phases are described.

1.4. OVERVIEW

In chapter five, algorithms performances are evaluated and how the application is tested through different use-cases is explained.

Finally chapter six contains the conclusion and the summary of the project. Some features are mentioned as a possible complementary features, which are lined up for the next version of this application.

Chapter 2

Background

This chapter briefly introduces important concepts and algorithms that can be used along this context. In the first section, automatic speech recognition and its categories are introduced. In the section two, search by voice efforts at Google is mentioned. In section three, common problems and limitations for voice detection are discussed. The next section, common metrics that are used for evaluating the automatic speech recognition are considered. Section five contains phonetic algorithms. Section six introduces Google Glass's interface, its technical specifications and Glass Development Kit. Finally, the last section includes Elastic-search engine and its possible setups that provide different filtering and indexing mechanism.

2.1 Automatic Speech Recognition

Automatic Speech Recognition (ASR) is known as the interpretation of human speech. Jurafsky [16] proposed a technical definition of ASR as a process of mapping acoustic signals to a string of words. He extended this concept to Automatic Speech Understanding (ASU) and the process was continued to produce a sort of understanding of the sentence.

Mats Blomberg [5] made a review on some applications using ASR and discussed the advantages that can be achieved. For example, the voice-controlled computer that was used for dictation could be an important application for physically disabled and lawyer, etc. Another application was environmental control applications, such as turning on a light and controlling the TV channels. In general, people with jobs where their hands are occupied, would greatly benefit from ASR-controlled applications.

In Speech Recognition, a phoneme is the smallest linguistic unit that can be combined together with other phonemes to create a sound "one" that would call a word. Words are made of letters, whereas the sound is heard, is made of phonemes. For example letters "o", "n", "e" that are combined together makes the word "one" and phonemes "W", "AH", "N" make a sound "one". Sample sonogram of a word 'zero' with marked boundaries for four phonemes is presented in Figure 2.1.

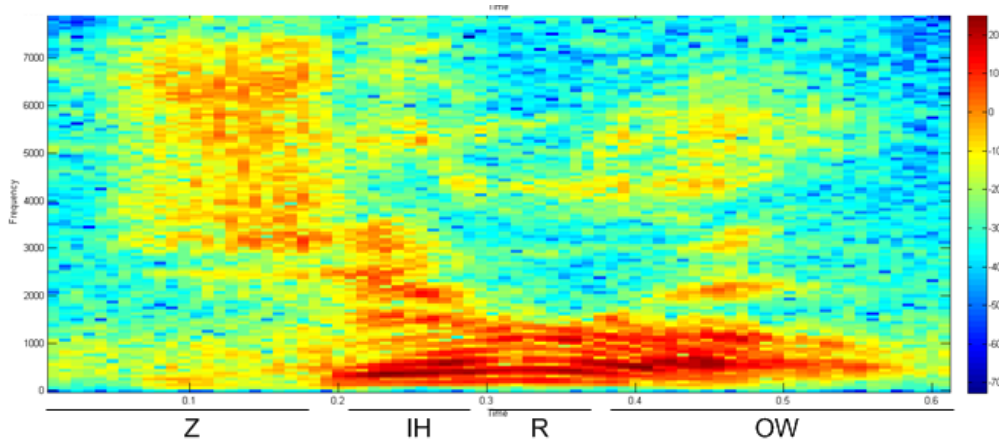


Figure 2.1: Phonemes 'Z', 'IH', 'R', 'OW' combined together form the word 'zero'.

2.1.1 Acoustic Phonetic Approach

In Acoustic Phonetic, each phoneme has its representation (or multiple representations) in the frequency domain, as illustrated in Figure 2.2. Some of them put more stress on low frequencies, some prefer high frequencies. Those sets of frequencies are called in acoustics "formant".

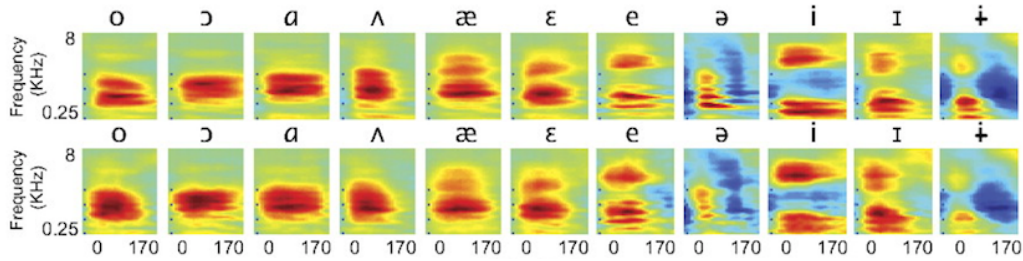


Figure 2.2: Sample phonemes and their representation in the frequency domain

During the recognition process, the speech signal has to be analyzed and the most probable distribution of the formant is retrieved [13]. It is based on assumption that, in language, there are finite and identifiable phonetic units that can be mapped out to a set of acoustic properties appearing in a speech signal. Even though the acoustic properties of phonetic units can be variable, the rules are not complicated and are learned by a machine with huge amounts of data.

This method contains several phases. In the first step, spectral measurements are converted to a set of features as descriptions for acoustic properties of the different phonetic units. In the segmentation and labeling phase, the speech signal is segmented into stable acoustic regions, and then one or more phonetic labels is attached to each segmented region which results in a phoneme lattice characterization of speech. At the end, a valid word is determined by considering the phonetic

2.2. SEARCH BY VOICE EFFORTS AT GOOGLE

label sequences that are produced by the previous step [30].

2.1.2 Pattern Recognition Approach

Another approach is pattern matching, which refers to two essential steps namely, pattern training and pattern comparison. In this approach, a well-formulated mathematical framework is used to establish consistent speech pattern representations. Then the pattern comparison is done based on a set of labeled training samples and a formal training algorithm [15, 17, 27].

This method compares the unknown speeches with each possible pattern. Patterns are made in training phase. It tries to determine it according to one of the known patterns. During the last six decades, Pattern Recognition approach has been the most important method for speech recognition [30].

2.1.3 Artificial Intelligence Approach (Knowledge Based Approach)

The next approach is a hybrid of the acoustic phonetic approach and pattern recognition approach. This knowledge-based method utilized some phonetic, linguistic and spectrogram information. In this approach, the recognition system was developed for having more speech sounds [2].

It provides little insight into the processing of the human speech and had an increased difficulty in analysis errors. However, it made a good and helpful linguistic and phonetic literature for better understanding of the human speech process. In fact, the recognition system and expert's speech knowledge are combined in this design. Usually, this knowledge is taken from a broad study of spectrogram and uses certain rules and procedures [2].

In this method, knowledge has also been used to guide the design of the models and algorithms of other techniques such as template matching and stochastic modeling. This form of knowledge application makes an important distinction between algorithms and knowledge that helps us in solving problems. Knowledge enables the algorithms to work better. In most successful strategic designs, these kinds of systems have significant contributions. It plays an important role in the selection of a suitable input representation, the definition of units of speech, or the design of the recognition algorithm itself [2].

2.2 Search by Voice Efforts at Google

Historically, searching for information through voice recognition is a common way historically that is not only specific to Computer or web domains. Already thirty years ago, people could dial directory assistance and ask an operator for a telephone number [29].

One of the most significant efforts was 800-GOOG-411 [3] when Google integrated speech recognition and web search in the form of an automated system.

This machine was used to find and call businesses. It was based on store-and-forward technology so that instead of direct interaction between the user and the operator, a request (city and state) was stored and later played it into the operator. In between, a search was constrained to businesses in the requested city [29].

Next version of GOOG-411 was released in 2008. It was able to search for a single utterance so there was no need to split apart the locations and businesses. The interaction became faster and provided greater flexibility for the users in the way they stated their needs. In this approach, the challenging part was extending the language model to a wider range other than to those businesses in the requested city[31].

Another Google voice detection effort emerged for iPhone to include a search via a voice feature named Google Mobile App (GMA). It extends the domain of a voice search from businesses on a map to the entire World Wide Web. Unlike GOOG-411, GMA is not very domain-dependent and it must handle anything that Google search can do. Apparently, it is more challenging due to the large range of the vocabulary and complexity of the queries [29]. The basic system architecture of the recognizer behind Google voice detection is illustrated in Figure 2.3.

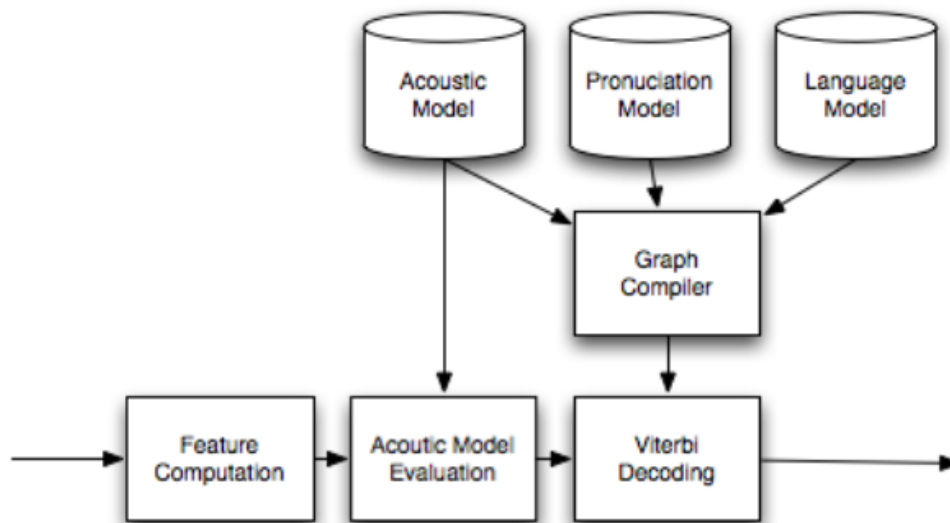


Figure 2.3: Basic block diagram of a speech recognizer

2.3 Difficulties with Automatic Speech Recognition

When considering the difficulties of automatic speech recognition, it is divided into three different categories. The first category is the role of humans in the process;

2.4. METRICS

second is related to the technology used for retrieving the speech, and lastly are the concerns of ambiguity and other characteristics and properties of the language.

2.3.1 Human

Human comprehension of speech is not comparable to automatic speech recognition in computers. Humans use both their ears and their knowledge about the speaker and the subject context. Humans can predict words and sentences by knowing the correct grammatical structures, idioms and the way that they say things. Also, body language, eye contact and postures are other advantages that human have. In contrast, ASR just employs the speech signal, statistical and grammatical models to guess the words and sentences [14].

Speaker variability is the other potential difficulty in retrieving the speech. Each speaker has a unique voice, speaking style and unique way to pronounce and emphasize words. Variability is also increased when the speaker shows his feelings such as sadness, anger or happiness etc. while speaking [14].

2.3.2 Technology

Technology has affected the process of converting speech to text. Speech is uttered in an environment of sounds and noises. These noises and echo effects should be identified and filtered out from the speech signal by modern and well-designed tools. Channel variability is another issue in ASR that can be revealed by the fact that the quality of microphones can affect the content of acoustic waves from the speaker to the computer. Moreover, for matching a large amount of sounds, words and sentences that are generated by the ASR computer, having good analysis tools and a comprehensive lexicon are necessary. These examples show how technology plays a role in the speech recognition process [14].

2.3.3 Language

One of the main issues in ASR is language modeling where it is needed to specify the differences between spoken and written language. For instance, grammar in spoken language differs from written language [1].

Another issue in modeling a language is ambiguity, where it cannot be decided easily which set of words is actually intended. This is called homophones, e.g. their and there. Word boundary ambiguity is another issue in modeling a language, where there are multiple ways of grouping phones into words [12].

2.4 Metrics

For evaluating the quality of a system, it is required to choose proper metrics. The metrics provide the better understanding of the system and the problems. Some

of the well-known metrics in this literature is reviewed here, to get an idea of how proposed model can be evaluated.

2.4.1 Word Error Rate

Word Error Rate (WER) measures the performance of an ASR. It compares words outputted by the recognizer to reference what users say [29]. Errors in this method are substitution, insertion and deletion. This method is defined as:

$$WER = \frac{S + D + I}{N}$$

where

- S: is the number of substitutions
- D: is the number of deletions
- I: is the number of insertions and
- N: is the number of words in the reference

Following are some examples to clarify WER:

- **Deletions:** consider the sentence for recognition is "Have a nice time", but ASR guess is "Have a time"
in this case 'nice' was deleted by the ASR, so a deletion happened.
- **Insertion:** consider the sentence for recognition is "what a girl!" but ASR guess is: what a nice girl
in this case 'nice' was inserted by the ASR, so a insertion happened.
- **Substitution:** consider the sentence for recognition is "Have a nice day", but ASR guess is "Have a bright day"
in this case 'nice' was substitution by the ASR, so a substitution happened.

2.4.2 Quality

While searching based on voice recognition, leaving one word out may not affect the final result. For instance, if user searches for "what is the highest mountain in the world?" The missing functioning words like "in" generally do not change the result [29]. Likewise, the misspelling of a plural form of a word (missing s) might not have an affect on the search either. These cases are considered in Web-Score metrics, which is based on measuring how many times a search result varies from a voice query. Using this Web-Score can specify the semantic quality of a recognizer [29].

2.5. PHONETIC ALGORITHM

2.4.3 Out of Vocabulary Rate

Out of Vocabulary (OOV) shows the percentage of words that are not modeled by the language model. The importance of this metric is in surrounding words where the recognizer cannot recognize the word because of subsequent poor predictions in the language model and acoustic misalignment. It is important to keep this rate as low as possible [29].

2.4.4 Latency

Time from when the user finishes speaking until the search result becomes visible on the screen is defined as latency of the system. It is affected by many factors such as the time to perform query, time it takes to render the search result and etc. [29].

2.5 Phonetic algorithm

Retrieving name-based information is complicated due to misspellings, nicknames and cultural variations. Languages are continuously developing over time and in many languages, especially English, how words are written might be different to how words are pronounced. Therefore, orthography is not a good candidate, whereas, phonology can reflect the current state of a language. Therefore, there is a demand to have a set of phonetic representations for each word of a language [6].

2.5.1 Soundex

In 1918, Russell developed Soundex algorithm. It was the first algorithm that tried to find a phonetic code for similar sounding words and was based on the fact that the nucleus of names consists of some sounds, which inadequately define names. These sounds may consist of one or more letters of the alphabet. As many names might have different spelling, the main idea of Soundex is to index a word by how it is pronounced rather than alphabetically written. It is explained in more detail, as Soundex is the base of other phonetic algorithms [8].

The Soundex code of a word consists of a letter followed by three digits. The letter is the first alphabet of the word and digits are in range of 1 to 6 that indicate different categories of specific letters according to table 2.1. If the word is not long enough to generate 3 numbers, an extra 0 is added to its end and if the generated code is longer than 3 numbers, the rest is shrunk. For instance, 'W252' is the code corresponds to 'Washington' and 'W000' is code for 'Wu' [8].

Soundex has some problems, as the accuracy is not good enough. There is a possibility of irrelevant matches (false positive), which causes low precision and it is the user who should check the correct word among many useless words after receiving the encoded results. This is common problem for many key-based algorithm [24].

Table 2.1: Soundex Coding Table

| Code | Letter | Description |
|------|------------------------|-------------------------|
| 1 | B, F, P, V | Labial |
| 2 | C, G, J, K, Q, S, X, Z | Gutturals and sibilants |
| 3 | D, T | Dental |
| 4 | L | Long liquid |
| 5 | M, N | Nasal |
| 6 | R | Short liquid |

The other problem is its dependency on initial letter. For example, if looking for Korbin but entering Corbin, no answer will be given [24].

2.5.2 Daitch-Mokotoff Soundex System

Randy Daitch and Gary Mokotoff developed Daitch-Mokotoff Soundex algorithm. They mainly added many Slavic and Yiddish surnames to Soundex system. Daitch-Mokotoff Soundex also contains refinements that are independent of cultural considerations [7].

In D-M algorithm, names are coded to six digits, each digit representing a sound listed in a coding chart. Just as the Soundex algorithm, it will be filled by zero when a name lacks enough coded sounds for six digits. It always encodes letter A, E, O, U, J, Y, and I, when they appear as the first letter or when two of them comes before a vowel. In the rest of the cases, they are skipped. In cases where there is more than one word involved, the system will encode it as a single word and link it together. Further more, there are some other rules in the implementation of the D-M algorithm that are ignored [7].

2.5.3 Metaphone

Lawrence Philips developed next phonetic algorithm, named Metaphone, with the intention to cover deficiencies in the Soundex algorithm. It is far more accurate than Soundex. Because it understands the basic rules of English pronunciation [25]. Similar to the Daitch-Mokotoff algorithm, it is based on sequences of letters rather than the single letter. It also encodes the whole name rather than just the initial part of it. In addition, it converts more English pronunciation rules such as when 'C' is pronounced as 'S' or pronounced as 'K' which is a big improvement when encoding similar words. However, it has some problems, for instance, encoding Bryan (BRYN) to Brian (BRN) or MacCafferey to MKKF. Also, retaining Soundex' choice of preserving the first letter does not work perfectly for words that start with vowels, e.g. "Otto" would not be matched to "Auto" [25].

2.5. PHONETIC ALGORITHM

2.5.4 Double Metaphone

While Metaphone worked well when the user knows the spelling of a word, it needed to be improved for uncommon words such as pharmaceutical words. Double Metaphone is a new version of the same algorithm with more accurate results [25]. It gives back a code shared between similar sounding words and a secondary code for cases with multiple variants of surnames from common ancestry and other unclear instances. A good example is the name "Smith", it is encoded to SM0 and XMT as primary and secondary codes, respectively, while XMT is the primary code of the name "Schemidt" as well [25].

It uses a complex rules to cover many irregularities in the English of Slavic, Germanic, Celtic, Greek, French, Italian, Spanish, Chinese, and other origin. Its main idea is to return two keys for words that can be reasonably pronounced more than one way, and that's why the new version is called Double Metaphone. This algorithm focuses more on the pronunciation of the words as the first key and will then usually return the native sound in the second key due to the fact that many last names have been changed to a more common spelling [25].

It is excellent at matching names and giving back consonant groups automatically with the common Anglicization, even if they do not really sound the same according to the usual standards of phonetic similarity, which means for instance it will match "Smith" to "Schmidt," "Filipowicz" to "Philipowitz," and "Jablonski" to "Yablonsky". Utilizing the strengths of both Soundex and Double Metaphone is the main idea of thorough searches in WorldVitalRecords.com website [25].

Searching via these algorithms is vital for the changes to the American English, as over time using words from other languages, it ensures that WorldVitalRecords.com is able to give a truly international approach to its searching measures [25].

2.5.5 Metaphone 3

Metaphone 3 is also designed so that it returns an approximate phonetic key for words and it generates same keys for similarly pronounced words. It considers a certain degree of 'fuzziness' to compensate variations in pronunciation, as well as misheard pronunciations. It is a commercial product, but is sold as source code[23].

2.5.6 Beider Morse Phonetic Matching

Beider-Morse system attempts to solve problems in previous algorithms. Specifically the number of matches found is often extremely large and consists of many names that could not be relevant at all. The Beider Morse system reduces the number of irrelevant matches by two steps. First the language is determined by spelling of a name, and then pronunciation rules are applied based on the detected language. It also considers the entire name rather than just some initial letters of it [4].

2.6 Google Glass

Google Glass is a pair of interactive eyeglasses that shows information in its optical display. It is a smart phone-like technology that communicates in a hands-free format with natural voice commands. It supports making phone calls, sending messages and taking photos and videos and delivering search results, for example by saying "take a picture", it captures what is in the users view. Other possible voice commands are shown in different menus for each application to guide the users to commit correct commands for supported functionality [10].

2.6.1 Google Glass Principles

Designing and using Glass is very different than present mobile platforms. While users these days are using several devices that store and display information for specific time periods, Glass works best with information that is simple, relevant, and current. It is not intended to be a replacement of a smart-phone, tablet, or laptop by transferring features designed for this device, instead, it focuses on how complement other services and delivers an experience that is unique. It should offer its features in a way that engages the user to use it when they want or live without taking away from it. The main idea is to be more useful by delivering information at the right place and at the right time for each task that leads to increased engagement and satisfaction. Otherwise it could be very annoying when unexpected functionality and bad experiences happen such as sending content too frequently and at unexpected times. Therefore, the intended creation of Glassware¹ should be clear to users and never pretend to be something that is not [10].

2.6.2 User Interface

The main user interface contains the comprised of 640 * 360 pixel cards that are exposed as the time-line (see Figure 2.4). It is possible to scroll through different sections of the time-lines to reveal cards. The Home card is the default card that users see when they wake Glass up and the most recent items reside closest to this card, such as replying to a text message or sharing a photo. Its user interface provides many features such as a standard way to present different cards and system-wide voice commands as a common way to launch Glassware. Once Glassware gets launched via voice commands, it is also possible to navigate the time-line with the touch-pad. Each card can also have menu items that let users carry out actions [10].

2.6.3 Technology Specs

Google Glasses look like an interactive pair of eyeglasses with smart phone-like display and natural language voice command support as well as Blue-tooth and

¹Glasswares are apps and services designed especially for Glass, built with Glass design principles.

2.7. ELASTIC-SEARCH



Figure 2.4: card time-line in Glass

Wi-Fi connectivity. Google Glass is compatible with both Android and Apple iOS mobile devices [10]. Other features of Glass are as follow:

- **Display:** High resolution with a 25 inch screen from eight feet away.
- **Camera:** capturing rate for photos 5 MP and Videos 720p
- **Audio:** Bone Conduction Transducer
- **Connectivity:** Wi-Fi - 802.11 b/g 2.4GHz and Blue-tooth
- **Storage:** 16 GB Flash. 12 GB of usable memory, synced with Google cloud.
- **Battery:** For one day of typical use (more battery intensive for video recording).

2.6.4 Glass Development Kit

The Glass Development kit (GDK) is an add-on to the Software Development kit (SDK) that lets us build Glassware that runs directly on Glass and by using both Android SDK and GDK, it is possible to utilize the vast array of Android APIs and also design a great experience for Glass [10].

2.6.5 The Mirror API

This REST API lets us build Glassware by using the chosen language and providing web-based APIs [10].

2.7 Elastic-search

The Elastic-search engine is a real-time search and analytic engine that is a flexible and powerful open source. The Elastic-search engine gives us the ability to move easily beyond simple full-text searches while architecturally it is useful in distributed environments where reliability and scalability are essential. It delivers most promises of search technology by its rich set of APIs, query DSLs and clients for the most popular programming languages. It indexes large amounts of data so that it is possible to run full-text searches and real-time statistics on it. For indexing, searching and managing any cluster settings, JSON file format is used over HTTP API and a JSON reply is taken back [9].

In fact, the Elastic-search can be treated as a NoSQL data store, with real-time search and analytical capabilities and with default document-oriented and high scalability. To have a tolerant cluster, it automatically divides data into shards, which are then replicated and equally balanced across the available servers in the cluster. This makes it easy to add and remove servers on the fly [9].

Elastic search analyzes each phrase during indexing and querying time. Configuration of analyzers can be done through the index analysis module. It can be used mainly for two cases, first when a document is indexed in order to both break indexed (analyzed) fields and second to process query strings. Analyzers are composed of different parts:

- **Tokenizer:** for breaking a string down into a stream of tokens or terms. As a simple tokenizer, the string might be split up into terms wherever a white-space or punctuation is met
- **TokenFilters:** Tokenizer sends a stream of tokens to it and it can modify, delete or add more tokens them, (e.g. lower casing, remove stop words, synonyms, respectively).

This process makes it possible to apply some additional algorithms on the data such as: stemming, stop words elimination, synonym filter, phonetic filter factory and many others.

2.7.1 NGrams Tokenizer

N-gram models are widely used in statistical natural language processing. Using an n-gram distribution, phonemes and sequences of phonemes can be modeled in speech recognition. N-gram is a sequence of characters constructed by taking a sub-string of a given string. In this thesis, the "NGrams" tokenizer can be used to generate tokens from sub-strings of the restaurant name value. As the received string from ASR is not precise enough, this application uses "NGram-tokenizer" to generate all of the sub-strings that later will be used in the index look-up table and be converted to the phonetic code, which will increase the chance of finding the correct restaurant name [9].

To setup the NGram tokenizer, following settings should be considered:

- **Min-gram:** Minimum size in code-points of a single n-gram, default value is 1
- **Max-gram:** Maximum size in code-points of a single n-gram, default value is 2
- **Token-chars:** Characters classes to keep in the tokens, Elastic-search will split on characters that do not belong to any of these classes, default keep all characters

2.7. ELASTIC-SEARCH

As an example for given string "Kebab Kungen", a tokenizer with min-gram equal to 8, max-gram equal to 10 and token-chars equal to letter and white-space, it produces these strings: "kebab ku", "kebab kun", "kebab kung", "ebab kun", "ebab kung", "ebab kunge", "bab kung", "bab kunge", "bab kungen", "ab kunge", "ab kungen", "ab kungen ", "b kungen", "b kungen " and " kungen ".

2.7.2 Phonetic Filters

Elastic defines six phonetic filters. Each one of them, given a sequence of characters (word), converts it into a tag according to its pronunciation. Those codes can be used to determine if two words sound alike. Filters supported by elastic were built on the basis of three most common solutions: Soundex, Metaphone and Carevphone:

- **Soundex:** was developed in the early twentieth century for indexing names. It creates a code that preserves first letter of the word and follows it by three digits. It is commonly used during genealogical research.
- **DoubleMetaphone:** is an improvement of Soundex, published in 1990. It covers many inconsistencies of the language and works well for words in general, not only names.
- **Caverphone:** was designed in 2002 at the University of Otago for name matching purposes, especially 19th and 20th century electoral rolls. The algorithm has been optimized for the New Zealand accents.

Phonetic filtering helps to match documents that were misunderstood by speech recognition algorithms.

Example: A name of a restaurant such as "Lion Bar" could be transcribed as "Leon Bar", since both words 'Kebab Kungen' and 'Chabab Kundun' sound very similar. The user's search for 'Lion Bar' would not find this relevant document while phonetic filters convert words 'Lion Bar' and 'Leon Bar' to "L516" (Soundex) or "LNPR" (DoubleMetaphone) codes.

Phonetic algorithms help to match similar words (like "Leon Bar" and "Lion Bar" or "Caina" and "Keine"). By utilizing the property of the filters as mentioned earlier, the phonetic algorithms can concatenate adjacent words and apply proper algorithms, since they analyze the sequence of characters and do not perform a dictionary look-up.

Chapter 3

Related Works

Morbini et al. [21] compared some speech recognizers, specifically cloud based systems, for developing domain-restricted dialogue systems, and analyzed their applicability for this regard. He showed better ASR performance comes with better natural language understanding performance and in general, the accuracy of local customizable systems is not better than distributed speech recognition systems, even if they are not specified to a domain. He didn't however evaluate a clear system. According to his effort, a combination of cloud-based recognizers and domain-restricted local system might lead to an overall better system.

Another speech recognition system is Sphinx-4, a modular, easily extensible, embeddable and open-source software developed at CMU [32]. Its decoder uses time synchronous Viterbi search using the token-pass algorithm [33] and the front-end includes voice activity detection. Its acoustic model is often based on the HUB-4 data set [11] and in the English MFCC features is used [19]. This mode is famous as the weak open source that needs changes in the feature representation, meaning, as well as the search space or linguist representation.

Ziolko [35] worked on Levenshtein distance costs. He used different insertion, deletion and substitution to held-out data to generate more accurate word hypotheses from a dynamic time warping-based phoneme recognizer [18]. Zgank and Kacic [34] tried to estimate acoustic confusability of words. They transformed word tokens to their phoneme representations and then they normalized Levenshtein distances between them and computed the confusability scores. Therefore, they estimated ASR performance by giving some currently available user commands. In contrast, this thesis intends to improve ASR results by comparing a word's phonemes using elastic search indexing for individual words.

The other interesting techniques are post-processing approaches. Milette and Stroud [20] optimized and processed the results from Google ASR on Android. They made a list of commands and by using a phonetic indexing algorithm to transformed the results from Google's speech recognition. Through this technique, they optimized the set of particular names rather than any words from the languages. However, multiple target commands might be reduced to the same representation.

Chapter 4

Implementation

This chapter discusses the motivation of selected algorithm for Google Glass application and contains a brief introduction to the environment used for development and the programming language.

4.1 Motivation of selected Phonetic Algorithms

This application is intended to search for a restaurant. Considering the current voice detection algorithm, Google Glass fails at detecting the searches in some cases, mainly because restaurant names are not totally modeled by the language model and might be pronounced considerably differently by different users. To be able to improve the detection accuracy, an algorithm is needed so it works fine with names and considers variations of pronunciation, as well.

Soundex, the first phonetic algorithm, is still a very well-known algorithm because of its simplicity and generality. Although its accuracy is not high in general detecting cases, it could be a good complement to the Glass detection algorithm in this application. There is a possibility of false positives and while using Soundex, the user should check the correct word among many useless words, but this application can do the checking in our encoded restaurant database. The way it encodes words, focusing on the beginning of a name and categorizing the rest based on similar sounding letters, will not strictly filter out many cases. This loose encoding might lead a higher chance of hitting on valid restaurant name in the next indexing phases.

Both Metaphone and Double Metaphone algorithms ignore non-initial vowels in their encoding and make same code for voiced/unvoiced consonant pairs, e.g. B/P or Z/S in the English language. This is useful in cases where adjacent vowels and consonants have been transposed, or voiced and unvoiced consonants are so close together in sound, but on the other hand the result set are huge. Metaphone 3 follows the same encoding, but adds the more flexibility to set the encoding in-order to include non-initial vowels, or to map voiced/unvoiced pairs to different encoding, or both. If the indexes have been prepared for a combination of these exact encoding

and the regular encoding, it will be more flexible in returning results to the user. If there are very few matches, the less exact matching might be return according to the existing standard of fuzziness, or, if there are too many matches, or plenty of "exact" matches, a more focused set could be returned.

Metaphone 3 has greater accuracy than Soundex, but because it is not an open source algorithm, the performances of Soundex and DoubleMetaphone algorithms are compared and experimented for this project in Google Glass to compensate the result into the valid range of restaurant names.

This research does a post-processing of results from Google ASR on Google Glass. The result from Google's speech recognition is transformed using a phonetic indexing algorithm such as Soundex or DoubleMetaphone and searching on domain knowledge (indexed restaurant) leading to improved ASR performance.

4.2 Motivation of Showing 6 Rows in Each Page

According to study of the Glass limitation and search result, this section will explain the motivation of showing 6 rows in each page of Google Glass.

4.2.1 Google Glass Limitation

Google Glass allows us to view applications and web pages using a display that projects information just above your line of sight. The official specs describe the Glass display as a "25-inch HD screen viewed from 8 feet away." So right up front, viewing distance matters Figure 4.1 [10]. This feature, in order to have a readable text, should use the bigger font size than what is expected.



Figure 4.1: Google Glass Display

Glass has a unique style, so it provides standard card templates, a color pallet, typography, and writing guidelines for us to follow. There are several available card layouts provided by the GDK to give users a consistent user experience, however, it

4.2. MOTIVATION OF SHOWING 6 ROWS IN EACH PAGE

is possible to design our own custom layouts by considering the Glass user interface's standard layout [10]. Glass defines dimensions for a set of common cards regions to make it easy to design and display different cards consistently Figure 4.2.

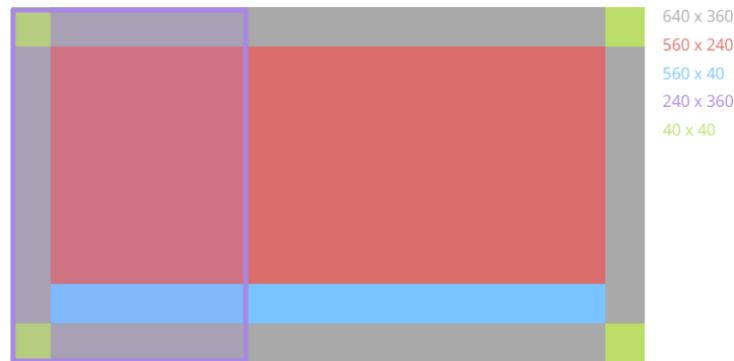


Figure 4.2: Card Layout Standard

- **Google Glass Typography** Glass displays all system text in Roboto Light, Roboto Regular, or Roboto Thin depending on font size.
- **Main Content** The main text content of the card is in Roboto Light with a minimum size of 32 pixels and is bounded by padding. Text that is 64 pixels and larger uses Roboto Thin.
- **Footer** The footer shows supplementary information about the card, such as the source of the card or a time-stamp. Footer text is 24 pixels, Roboto Regular.
- **Full-bleed Image** Images work best when they are full-bleed and do not require the 40px of padding that text requires.
- **Left Image or Column** Left image or columns require modifications to padding and text content.
- **Padding** Time-line cards have 40 pixels of padding on all sides for the text content. This allows most people to see your content clearly.

Considering the screen size of the Glass (640*360) and consistency of font styles, Google uses and suggests the Roboto Light font with size 32dp as a default font style, which implies 6 different rows to show the information with good quality readability on the Google Glass.

4.2.2 Users Behavior in Searching Areas

Apart from screen limitation and the fact that the space for text is limited, it will be interesting to see how users usually behave in searching areas.

- **Google Organic Click-Through**

The organic click through rate is one of the best metrics to measure online businesses' performance in SERPs (Search engine results page) and to illustrate the users' search behavior.

The question of how organic click-through rate is evolved into search engine's results came up almost a decade ago and is still left unanswered [28]. But there are different studies in this area showing how different types of search results influence users' behavior and what role user intent plays in determining the distribution of clicks [28].

In addition to the well-known tweaks marketers make to influence organic CTR such as titles, meta descriptions and snippets, there are several other elements such as ads, user intent, user device also have an impact on organic CTR.

- **Previous CTR Studies**

For having the overall view of the click-through rates on organic search results let's take a look at the research, which is done, in this area Figure 4.3 [28].

| Position | AOL 2006 | Enquiro 2007 | Chitika 2010 | Optify 2010 | Slingshot 2011 | Chitika 2013 | Catalyst 2013 | Caphyon 2014 |
|---------------|--------------|--------------|--------------|-------------|----------------|--------------|---------------|----------------|
| 1 | 42.3 | 27.1 | 34.35 | 36.4 | 18.20 | 32.5 | 17.16 | 31.24 |
| 2 | 11.92 | 11.7 | 16.96 | 12.5 | 10.05 | 17.6 | 9.94 | 14.04 |
| 3 | 8.44 | 8.7 | 11.42 | 9.5 | 7.22 | 11.4 | 7.64 | 9.85 |
| 4 | 6.03 | 5.1 | 7.73 | 7.9 | 4.81 | 8.1 | 5.31 | 6.97 |
| 5 | 4.86 | 4.0 | 6.19 | 6.1 | 3.09 | 6.1 | 3.5 | 5.5 |
| 1 to 5 | 73.55 | 56.6 | 76.65 | 72.4 | 43.37 | 75.7 | 43.55 | 67.6 |
| 6 | 3.99 | 4.1 | 5.05 | 4.1 | 2.76 | 4.4 | 1.63 | 3.73 (6 to 10) |
| 7 | 3.37 | 4.1 | 4.02 | 3.8 | 1.88 | 3.5 | 1.09 | N/A |
| 8 | 2.98 | 3.2 | 3.47 | 3.5 | 1.75 | 3.1 | 1.04 | N/A |
| 9 | 2.83 | 2.8 | 2.85 | 3.0 | 1.52 | 2.6 | 0.44 | N/A |
| 10 | 2.97 | 3.6 | 2.71 | 2.2 | 1.04 | 2.4 | 0.51 | N/A |
| Top 10 | 89.69 | 73 | 95 | 89 | 52 | 92 | 48 | 71.33 |

Figure 4.3: CTR In Different Year

It is important to emphasize with having the major differences (see Figure 4.4) in the methodologies applied for each study, the user preferred to stay in the first page and click only on 5 top hits.

In order to have a closer look at user behavior in the selecting the result in search, searching with specific intent and in different devices will be considered. In Figure 4.5, users of mobile phone or desktop are more interested in 5 hits in the first page [28].

4.2. MOTIVATION OF SHOWING 6 ROWS IN EACH PAGE

| | AOL 2006 | Enquiro 2007 | Chitika 2010 | Optify 2010 | Slingshot 2011 | Chitika 2013 | Catalyst 2013 | Caphyon 2014 |
|-----------------|--------------------------|------------------------------|---------------------------------------------------|-----------------------------|--------------------------------------|------------------------------|---------------------------------------------------|------------------------------|
| Keywords | 21,011,340 | N/A | N/A | 1,000 | 324 | 17,500 | N/A | 465,020 |
| Websites | 500,000 users | 1,000 subjects | > 300,000 | 250 | 200 | 59 | > 300,000 | 5,176 |
| Verticals | Various | B2B | Various | B2B | Various | CPG | Various | Various |
| Timeframe | 3 months in 2006 | March 2007 | May 2010 | Dec 2010 | Jan 2011 – Jul 2011 | Oct 2012 – Jun 2013 | May 2013 | July 2014 |
| Search Engine | AOL | Google | Google | Google | Google & Bing | Google | Google | Google |
| Location | N. America | N. America | N. America | US | US | US | N. America | US and International |
| Research Method | Web log parser | Eye-tracking | Based on ad impressions served within our network | N/A | GA & Keyword Planner data processing | GWT data processing | Based on ad impressions served within our network | GWT data processing |

Figure 4.4: Different Research In Different Area

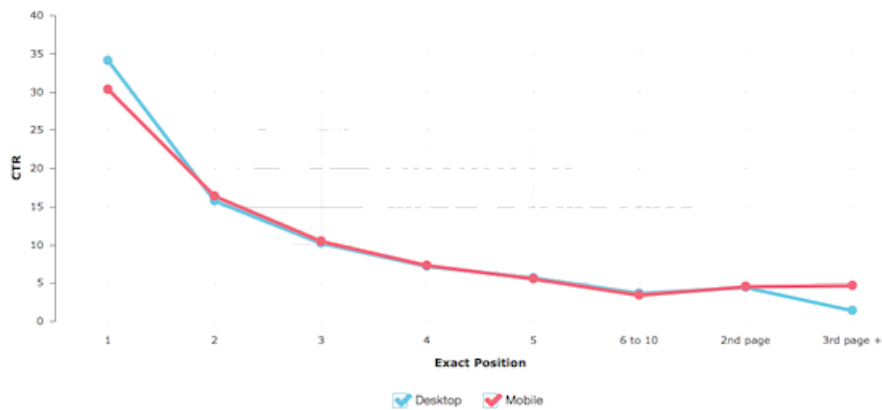


Figure 4.5: Search Result In Different Device

The next Figure 4.6 shows the average result for searching in, with different intent such as location, commercial and informational.

The interesting factor in most cases is that most users stay in the first page and do not proceed to the second page to explore more results as they most likely have expectations for the top 5 results. Using Google Glass as the target device in the specific search domain, searching for a restaurant name, this study will be a good reference while considering the impact of this device

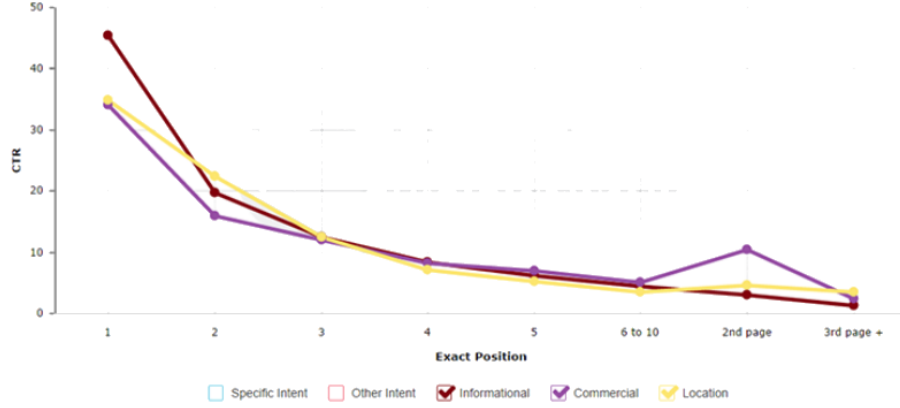


Figure 4.6: Search Result With Specific Intent

features on CTR.

4.2.3 Conclusion

In one hand, the Glass limitation and its suggestion for page/card layout comes to 6 rows for displaying information in one page/card. On the other hand, considering the users behavior for searching on different devices, the first page and mostly the 5 top hits are most likely to be checked by users. Therefore, 5 is chosen as the key value to show the results, evaluate and compare the phonetic algorithms in this application.

4.3 Data Set

There are many useful sources available on-line which provide information about city restaurants, for instance www.enrio.se, www.foursquare.com and www.restaurangkartan.se. They all provide good information about the menu, quality of food, rank of restaurant and visitors' reviews in addition to the basic information such as the address, contact information, place on a map, and so on. www.restaurangkartan.se is quite popular for sharing experiences in restaurants and is therefore used as a database reference in this project.

Some information about the restaurants name, address, phones number, rank and top reviews are collected. Of course, there is more to extract but due to limitation of the Glass screen, only the essential information will be visualized. To be able to have an on-line database, a web crawler is made to go through a each page of www.restaurangkartan.se and collect all information that is needed.

4.4 Norconex

Norconex HTTP Collector is used as a web crawler in this project to collect information about the restaurant. There are many helpful features in Norconex that are used in this thesis. It is multi-threaded and has language detection tools, which have the ability to configure the crawling speed. It provides filtering of documents based on URL, HTTP headers, content, or meta-data and is able to store crawled URLs in different database engines [22].

4.4.1 Importer Configuration Options

To extract text out of documents, HTTP collector has an integral part named importer module. It provides document manipulation options, filtering options and facilities that are distributed with the HTTP collector [22].

4.4.2 Committer Configuration Options

The Committer module takes the text extracted out of the collected documents and submits it to a target repository (e.g. search engine). A Committer implementation is customized to the target repository that is defined using configuration options, which are specific to each Committer [22].

4.4.3 More Options

Norconex provides many more options to structure the configuration files properly, such as creating reusable configuration fragments and using variables to make our files easier to maintain and more portable across different environments [22].

The Norconex XML configuration file consists of one filter for going through the web-pages and some filters for taking information that is needed such as name of restaurant, address, Tel etc. in order to finally take the text extracted out of all collected documents in the Committer module and submit it to the Elasticsearch engine.

4.5 Elasticsearch

Elastic is a powerful open source platform for indexing and searching among text documents. It makes it easy to approach advanced features such as, hit word highlighting or faceted search (arrange search results in groups of some important fields like author or date).

When documents are indexed in Elasticsearch, its analyzers, tokenizers and token filters are used to build an inverted index. Theoretically, this is a dictionary containing a list of terms (tokens), together with references to the documents in which those terms appear. There is an "analyzer" for each field in the mapping, which contains of a tokenizer and some token filters. The analyzer is responsible for

transforming the text of a given document field into the tokens in the look-up table used by the inverted index. When a text search is performed, the search text is also analyzed (usually), by comparing them against the resulting tokens (in inverted index), the referenced documents are returned in match cases [9].

4.6 Demo Of Application

Similar to all applications in the Google Glass, the Find Restaurant application is triggered by voice. By saying "Ok Glass" the main menu Figure 4.7 is displayed and the application is on the list of applications that Glass supports. The user can run the application by either calling its name or selecting it from the touch-pad panel. Figure 4.8 will be shown as the main page/card in this application.

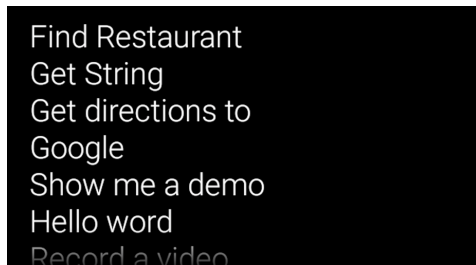


Figure 4.7: Menu In Google Glass

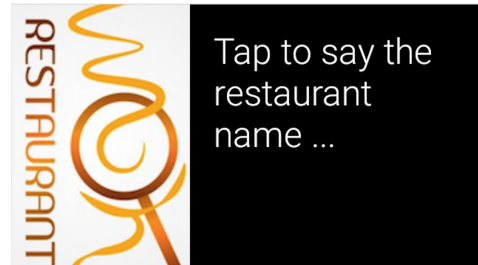


Figure 4.8: First card in application

When tapping on this card, voice recognition is triggered Figure 4.9 and the user can say the name of the restaurant. After taking the detected string by Glass API, database will be searched for finding the closest valid restaurant name.

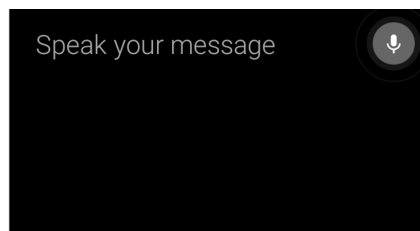


Figure 4.9: Voice trigger for saying the name of restaurant

When possible matches are found, they will be listed on the following card and allow the user chose the best hit, Figure 4.10.

The user should tap on the card or say, "Ok Glass" to see the options menu in order to either select the desire restaurant or ask to retry the search (see Figure 4.10). The Retry option might be useful when the algorithm doesn't show the proper restaurant in the suggested list, when the restaurant is not stored in the database at all, or when the user wants to try a different name.

4.6. DEMO OF APPLICATION



Figure 4.10: Selected Menu

By selecting or tapping on a restaurant name, a card containing some information about the selected restaurant and its picture will be shown in Figure 4.11. The content of this card is retrieved on-line from "www.restaurangkartan.se" web page. There are some options, like getting directions and reading more about the restaurant, provided for the user on the menu list of this card which can be selected via tapping on the card or saying "OK Glass" as usual, Figure 4.12.

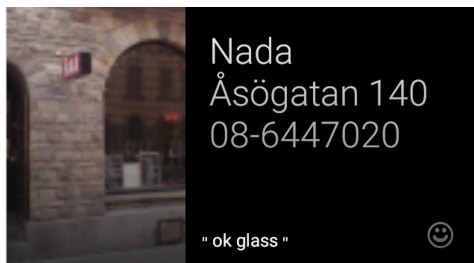


Figure 4.11: Restaurant Information

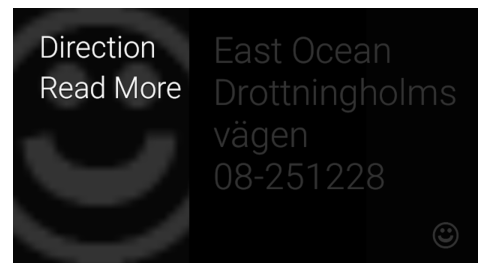


Figure 4.12: Options

By selecting the "Direction" option, Figure 4.13 will pop up that shows the direction from the user's position to the restaurant address, providing the GPS is on. This card has different options for getting to the destination, such as bike, train or personal car.

By selecting "Read More", the corresponding restaurant web page from "restaurangkartan.se" will be shown containing more information such as reviews, rating and a short description. At any stage, the user can get out from the application by doing the top down gesture on the Glass touch-pad.

CHAPTER 4. IMPLEMENTATION

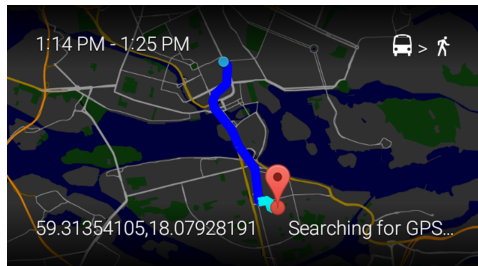


Figure 4.13: Direction

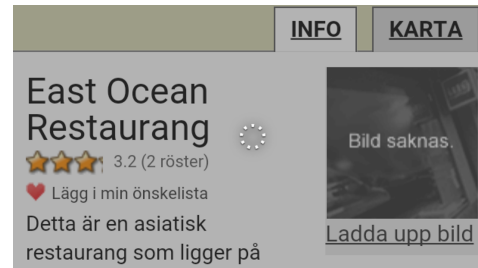


Figure 4.14: Read More

Chapter 5

Evaluation

For the evaluation of this project, both audios containing human speech and texts with true output for comparison are required. The simplest way is by involving people who use the device and produce the audios and store the corresponding names as the reference texts.

5.1 Test data

Due to the large number of possible restaurant names, a wise choice of comparison models is crucial. It might be impossible or incredibly difficult to decode all of the possible words, but a proper approximation for a given case is still useful. A test of 300 restaurant's names was made. All the cases were pronounced by 10 persons and 30 names for each implying different voices, reading speed etc.. In order to check the ability of phonetic algorithm to encoding different phonetic sounds, almost all voices were covered in selected restaurants for each person. An example of these diversities is shown in table 5.1. The complete table is shown in Appendix C.

Table 5.1: Phonetic Diversities Sound

| Phonetics Symbols | Restaurant Name |
|-------------------|-------------------|
| t,s,sh,n,l,i | Oriental Sushi |
| g,d,n,r | Angry Diner |
| b,t,r,e | Brother Tuck |
| b,l,i | Blue Chili |
| h,m,r | Maharajah |
| f,h,l,r | Food For Pleasure |
| f,v,h,j | Fair View House |
| t,w | Thai Take Away |

5.2 Quantity Tests

This application is evaluated using two quantity tests. The first interesting test is to see how many times using phonetic filters and N-grams compensated for the errors of speech to text conversion and is reported the correct name among the suggestions. The second test explores where the answer is located in suggested hits.

In the first test, the total percentage of the tries that ended up with the desired results is calculated. To investigate at which place the result is located, an average rank of all results ordered by probability of the correctness is used.

5.3 Result

A base line algorithm is considered as a reference to evaluate the ability of different phonetic algorithms in retrieving the isolated proper name (restaurant name). In base line algorithm, the string, taken from Glass voice recognition API, is divided by its white-spaces and searched in our database. Then the percentage of searches ended up with the correct restaurant name is reported as a reference.

Each of the search algorithms, base line, Soundex and Double Metaphone, uses two different tokenizers, whitespace and N-gram Tokenizers. In the current test, N-gram with min 5 gram and max 7 gram is used.

Without considering the screen size limitation of the Glass, Figure 5.1 compares the percentage of hits in the algorithms. Overall, there is a significant difference between the results of base line algorithm and phonetic algorithms.

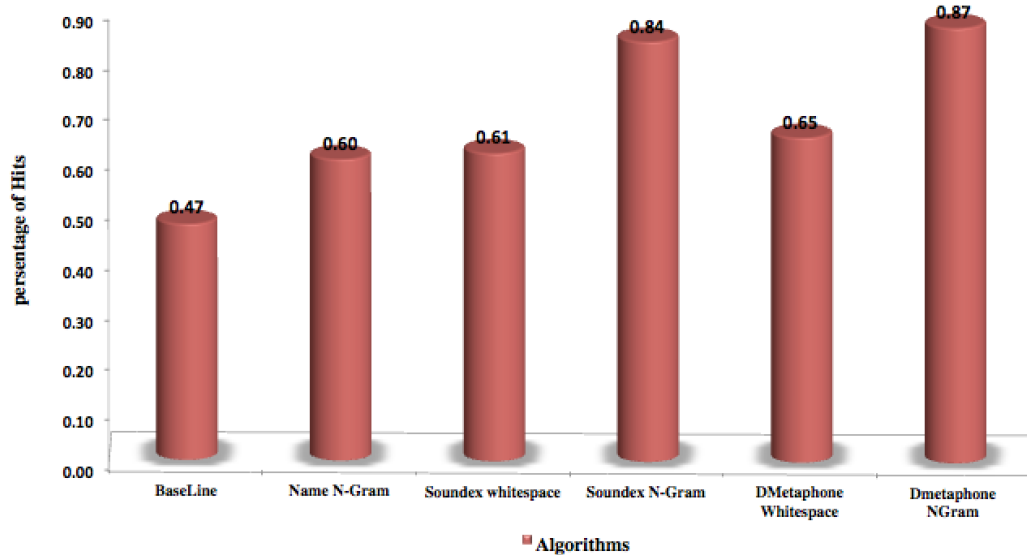


Figure 5.1: Percentage Of Hits Without Limitation Of Glass

To begin with, nearly 87% of searches hit the correct restaurant names in Double

5.4. BEST ALGORITHM FOR GOOGLE GLASS

Metaphone algorithm using N-Gram, which is the best, compared to the other tests. It performs better than the base line algorithm and performs better than Soundex algorithms in both White-space and N-gram with 61% and 84%, respectively.

It is worth mentioning that for all algorithms, the average of the ranks is less than 12, where Double Metaphone N-Gram is 11.14 and base line Whitespace is 4.06 (see Figure 5.2).

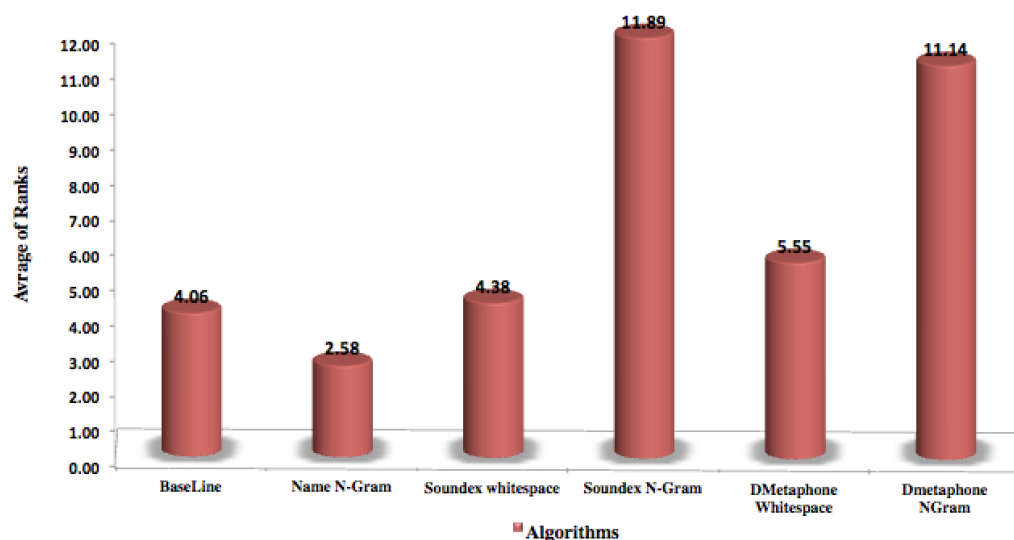


Figure 5.2: Average Of Ranks Without Limitation of Glass

In conclusion, Double Metaphone algorithms work better than base line and Soundex algorithms, in the term of total percentage of hits and average of ranks.

5.4 Best Algorithm For Google Glass

Another interesting test is to see how the screen limitation may affect the choice of the most suitable algorithm for Google Glass. The Glass screen is small and showing all results of a search on a small card is not feasible. Therefore, this constraint needs to be considered. This device is designed for showing short information of a search; at the same time traversing too many cards to find the correct result might be annoying. Therefore, the results are limited to 5 top hits and shown in one card on the Glass screen based on the discussion in section 4.2.

Figure 5.3 shows the percentage successful searches with limited hits considering the Glass limitation. Again, there is a significant difference between results of phonetic algorithms and base line algorithm.

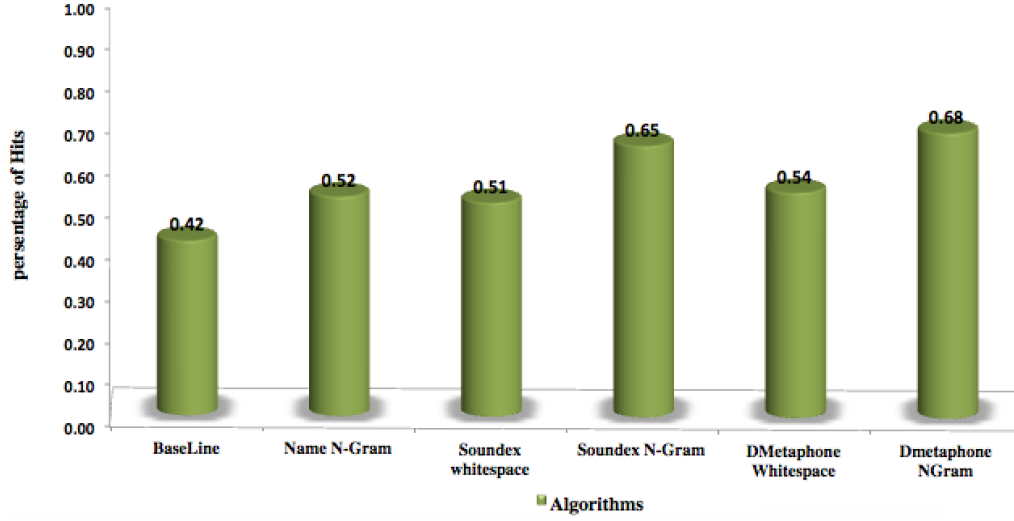


Figure 5.3: Percentage Of Hits With Limitation Of Glass

Figure 5.4 shows the average of ranks for 5 hits in each algorithm. It is noticeable that all algorithms have a good average of ranks, less than 2.

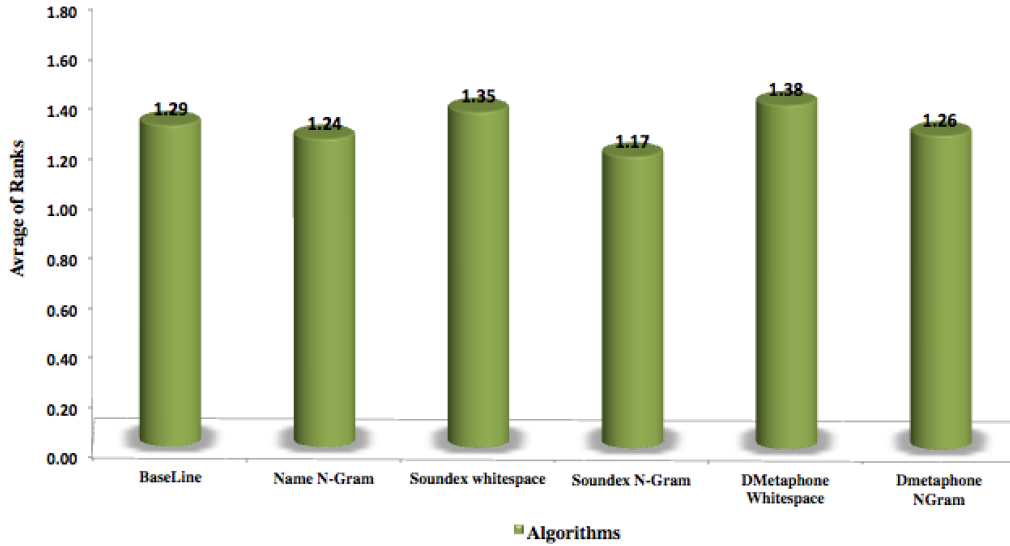


Figure 5.4: Average Of Ranks With Limitation of Glass

Besides analyzing the average of rank and percentage of hits, all algorithms used in this study are evaluated by more factors such as the precision and F-measure [26]. The F-measure is a weighted average of precision and recall. Considering the importance of recall in this case compared to precision, high weighted factor (0.9)

5.4. BEST ALGORITHM FOR GOOGLE GLASS

for recall, and less for precision (0.1) are used.

Figures 5.5 and 5.6 show precision and F-measure in each algorithm. Although Metaphone algorithm's precision is not the best between all algorithms, it has a great level in the F-measure instead.

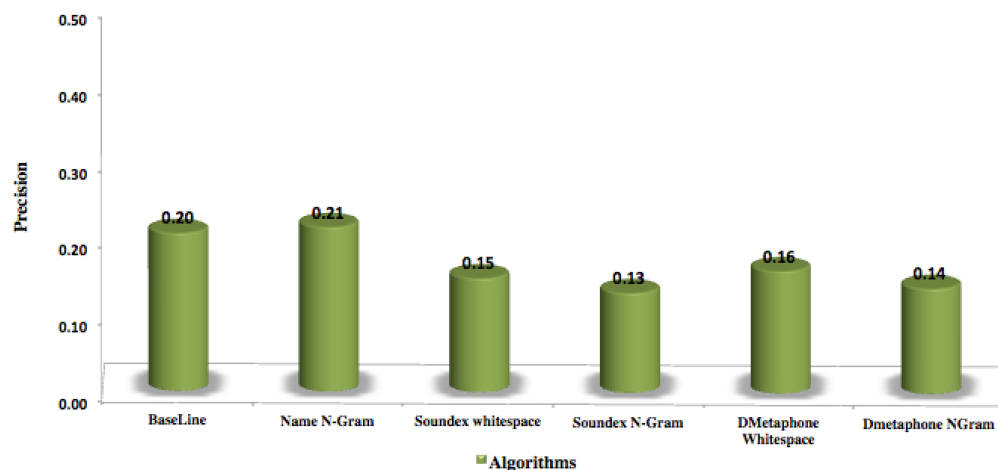


Figure 5.5: Precision

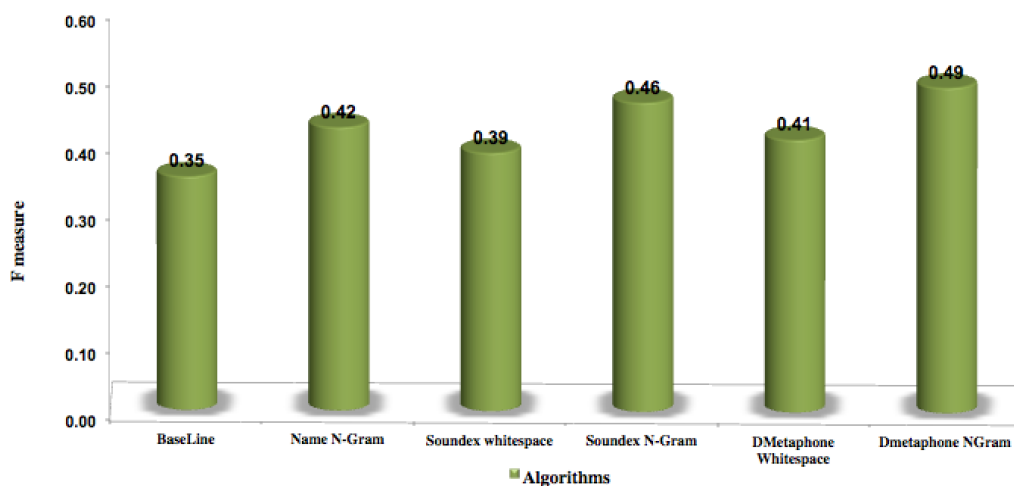


Figure 5.6: F-measure

To be able to select the best algorithm, apart from all these metrics, it is important to see how well each algorithm performs compared to the base line in terms of covering those restaurants that are found. According to Figure 5.7, DoubleMetaphone algorithm covers 93% of found restaurant by base line.

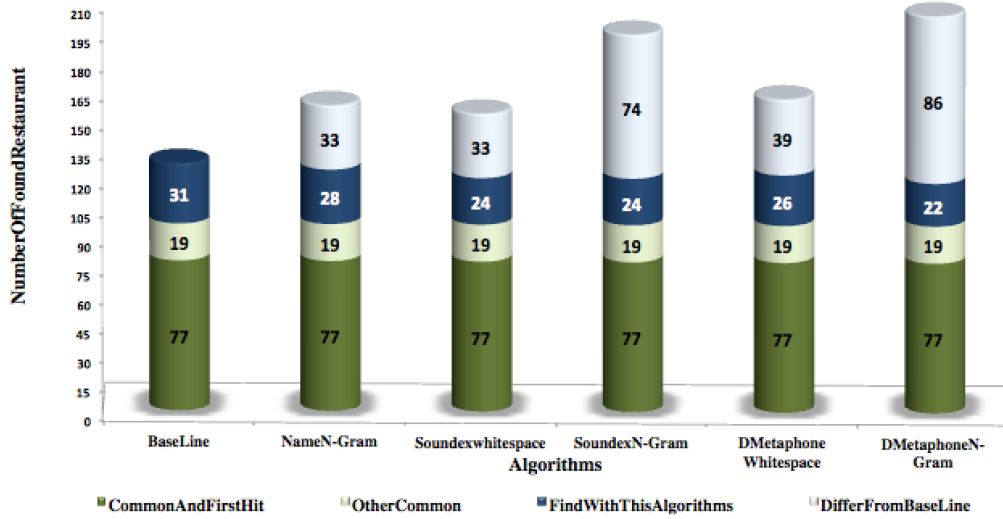


Figure 5.7: Covering Restaurants by Each Algorithm

5.4.1 Improving Precision and F-measure

Apart from the Glass limitation screen, showing even less than 5 hits can increase the usability of Glass. In order to show the minimum number of results, the outputs of different test cases were investigated. To determine if there were any connections between the ranking scores and hits that could lead to less numbers of hits on the screen, the results showed that when the algorithms found the searched hits from the first rank, the score was almost double the second hit. In these cases, one hit would be shown to the user. Figures 5.8 and 5.9 demonstrate how using this limit can improve the precision and F-measure metrics. Moreover, showing fewer hits to the user would increase the usability of this application.

5.5 Conclusion

As discussed earlier, Double Metaphone N-gram algorithm has the best recall (in term of ranks) and it is undoubtedly important in this kind of application with screen size limitation and with only one result being correct. The other advantage is that Double Metaphone found 93% of hits that the base line algorithm found and had a high F-measure. Therefore, it is selected as the best algorithm for this application in Google Glass among those was examined.

5.5. CONCLUSION

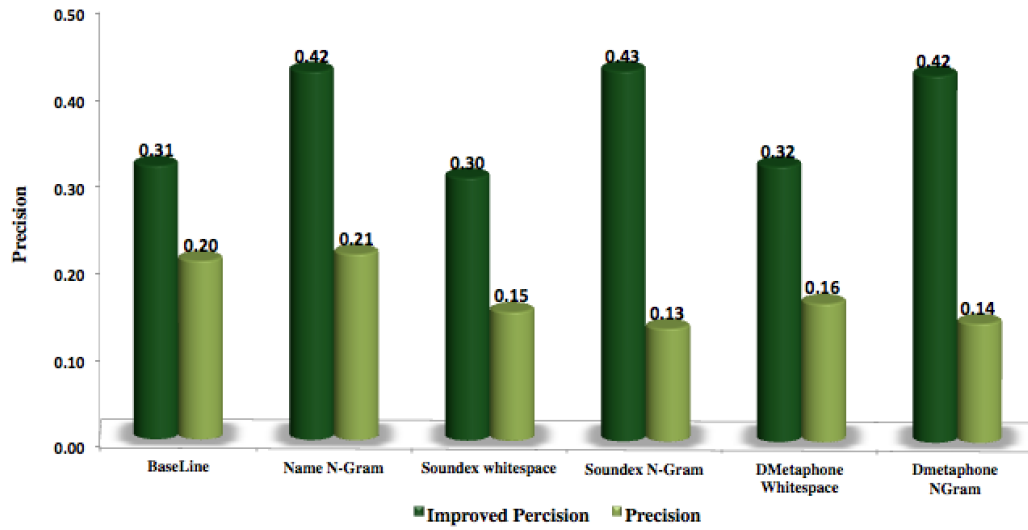


Figure 5.8: Improved precision

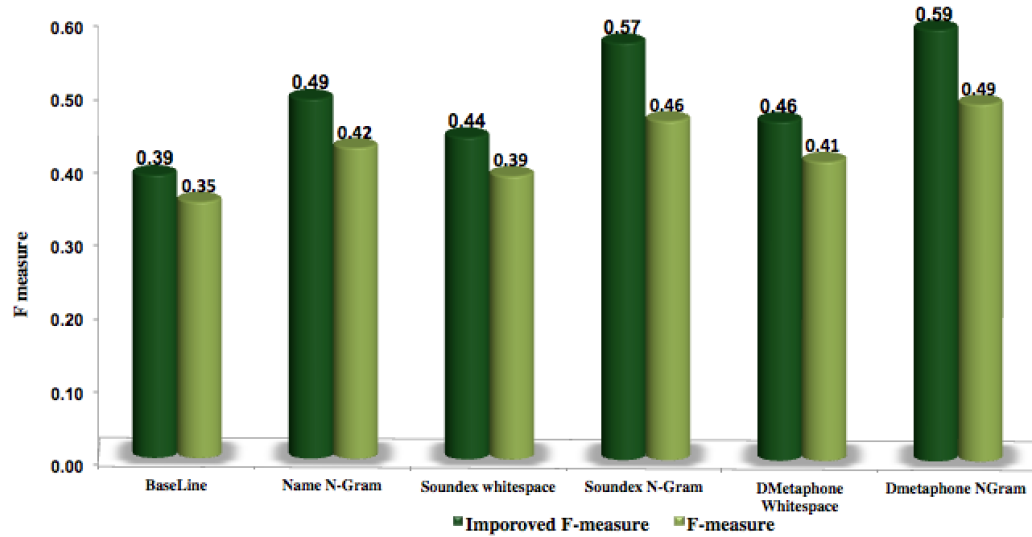


Figure 5.9: Improved F-measure

Chapter 6

Discussion and Conclusions

This project utilized speech recognition API and phonetic algorithms together to perform searching Stockholm's restaurant names via Google Glass with higher precision. During the project, there have been tests run for several filters and Tokenizers to figure out the most suitable combination for the Glass interface. The following conclusions were made.

As one of the main weakness in automatic speech recognition, specifying the proper names is a challenge and in the target project, it came to the name of restaurants in Stockholm city. However, by using a proper phonetic algorithm, the percentage of hits were increased when one searched for the restaurants name. It was important to consider two categories of the restaurant name, those words from the English dictionary, specially if they had ambiguity in the way they could be pronounced such as Blue Chili, Lion Bar, Doctor Salad and etc. and the names that were not originally native English word such as Gröndals Sushikök, Hamburgerprinsen, Mondo Tapas Och Bar and etc.. For instance, searching for the Lion Bar restaurant, Glass API's gave Leon Bar. While BaseLine algorithm was not able to find the correct match, DM-phonetic found the document by using the same code (LN, PR) for both names in less than a fifth hit.

The ability of phonetic algorithms became more clear while searching for non English names. Due to the current limitation of Glass that the supporting language and voice recognition API's library are limited to English, while searching for Hamburgerprinsen, Glass gave us Hamburger Pension instead, however, the same DM-phonetic code for both names (HMPR) helped to find the Swedish restaurant name.

The other features that were affected by choosing the best algorithm for Google Glass was the screen size limitation and its resolution. Investigating of the user behavior while searching on the different devices and considering the Glass limitation, it was shown that demonstrating top 5 hits at most was more effective and there were evaluated algorithms for them.

In Find Restaurant application, involving the user in selecting the desired search result increased the possibility of finding the restaurant name, which would ulti-

mately increase the user satisfaction.

In overall, the ability of phonetic algorithms and N-gram analyzer in retrieving the word were considered and how they can be combined with automatic speech recognition and helped to find the correct match were described. Specifically in the target application, current automatic speech recognition did not work well enough in searching proper names such as restaurants. Therefore, chance of finding the intended query in our infinite database was increased by filtering the result using phonetic algorithm and different analyzers. The combination of these algorithms and the Glass limitation (when showing as few results as possible on the small screen) makes using a phonetic filtering algorithm very helpful in gaining better results. This idea is usable for any devices with small screen sizes and low resolution such as personal digital assistant (PDA).

6.1 Development Limitations

- **GDK Issue:** It was not found clearly specified on the Google Glass web site [10] that which SDK APIs are compatible and cooperative with Glass, even-though it was stated that "By using the Android SDK and GDK, we can leverage the vast array of Android APIs and also design a great experience for Glass". However, some features were found in the voice recognition API that are not compatible and do not cooperate with Google Glass at all, for example other languages.
- **Google Glass Issue:** Debugging program on Glass is difficult as it is essential for device to be cold. However, while running the program and debugging it, the device will quickly heat up. This makes the programming complicated and possibly annoying.

6.2 Further Research

- **Other Language on Google Glass:** Supporting more languages in Google Glass could be interesting work in the future for these kind of applications. Searching for words via different languages would increase the chance of finding the correct result as it is very common that restaurant names are from different languages. By the time this thesis was done, there was no support for other languages, however this might be added in the near future, as it is expected to be more international. Alternatively, using a third-party recognition system such as Nuance's advanced speech recognition could be rather helpful.
- **Quality of Inputs:** It may be worth while to do pre-filtering audio signals and have some noise reduction. Obviously, the higher the quality of the input data is, the better the ASR results could be generated for the steps to follow.

List of Tables

| | | |
|-----|--------------------------------------|----|
| 2.1 | Soundex Coding Table | 12 |
| 5.1 | Phonetic Diversities Sound | 31 |

List of Figures

| | | |
|------|-------------------------------------------------------------------------|----|
| 2.1 | Phonemes 'Z', 'IH', 'R', 'OW' combined together form the word 'zero'. . | 6 |
| 2.2 | Sample phonemes and their representation in the frequency domain . . | 6 |
| 2.3 | Basic block diagram of a speech recognizer | 8 |
| 2.4 | card time-line in Glass | 15 |
| 4.1 | Google Glass Display | 22 |
| 4.2 | Card Layout Standard | 23 |
| 4.3 | CTR In Different Year | 24 |
| 4.4 | Different Research In Different Area | 25 |
| 4.5 | Search Result In Different Device | 25 |
| 4.6 | Search Result With Specific Intent | 26 |
| 4.7 | Menu In Google Glass | 28 |
| 4.8 | First card in application | 28 |
| 4.9 | Voice trigger for saying the name of restaurant | 28 |
| 4.10 | Selected Menu | 29 |
| 4.11 | Restaurant Information | 29 |
| 4.12 | Options | 29 |
| 4.13 | Direction | 30 |
| 4.14 | Read More | 30 |

| | | |
|-----|----------------------------------------------------------|----|
| 5.1 | Percentage Of Hits Without Limitation Of Glass | 32 |
| 5.2 | Average Of Ranks Without Limitation of Glass | 33 |
| 5.3 | Percentage Of Hits With Limitation Of Glass | 34 |
| 5.4 | Average Of Ranks With Limitation of Glass | 34 |
| 5.5 | Precision | 35 |
| 5.6 | F-measure | 35 |
| 5.7 | Covering Restaurants by Each Algorithm | 36 |
| 5.8 | Improved precision | 37 |
| 5.9 | Improved F-measure | 37 |
| C.1 | Restaurant Name | 51 |
| C.2 | Restaurant Name | 52 |

Bibliography

- [1] Jens Allwood, Maria Björnberg, Leif Grönqvist, Elisabeth Ahlsén, and Cajsa Ottesjö. Corpus-based research on spoken language, göteborg university. In *FQS–Forum Qualitative Social Research*, volume 1, pages 19–27, 2002.
- [2] MA Anusuya and Shriniwas K Katti. Speech recognition by machine, a review. *arXiv preprint arXiv:1001.2267*, 2010.
- [3] Michiel Bacchiani, Françoise Beaufays, Johan Schalkwyk, Mike Schuster, and Brian Strope. Deploying goog-411: Early lessons in data, measurement, and testing. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 5260–5263. IEEE, 2008.
- [4] Alexander Beider and Stephen P Morse. Phonetic matching: A better soundex. *Association of Professional Genealogists Quarterly*, 2010.
- [5] Frédéric Bimbot, Mats Blomberg, Lou Boves, Gérard Chollet, Cédric Jaboulet, Bruno Jacob, Jamal Kharroubi, Johan Koolwaaij, Johan Lindberg, Johnny Mariethoz, et al. An overview of the picasso project research activities in speaker verification for telephone applications. In *Eurospeech*, 1999.
- [6] Joe Celko and SQL Joe Celko’s. for smarties: Advanced sql programming. *Morgan Kaufmann Publishers Inc., San Francisco, CA, USA*, 53:54, 1995.
- [7] Maxime Crochemore, Christophe Hancart, and Thierry Lecroq. *Algorithms on strings*. Cambridge University Press, 2007.
- [8] Arlene H Eakle and Johni Cerny. *The source: A guidebook of American genealogy*. Ancestry. com, 1984.
- [9] Elastic Faculties. Elastic, url = <http://www.elastic.co/guide/en/elasticsearch/reference/1.5/analysis.html>, 2015.
- [10] Google Faculties. Google glass, url = <https://developers.google.com/glass/>, January 2015.
- [11] John Garofolo, Jonathan G Fiscus, and William M Fisher. Design and preparation of the 1996 hub-4 broadcast news benchmark test corpora. In *Proc. DARPA Speech Recognition Workshop*, pages 15–21. Citeseer, 1997.

BIBLIOGRAPHY

- [12] Ben Gold, Nelson Morgan, and Dan Ellis. *Speech and audio signal processing: processing and perception of speech and music*. John Wiley & Sons, 2011.
- [13] John F Hemdal and George W Hughes. A feature based computer recognition program for the modeling of vowel perception. *Models for the Perception of Speech and Visual Form*, Wathen-Dunn, W. Ed. MIT Press, Cambridge, MA, 1967.
- [14] Janet Holmes et al. *An introduction to sociolinguistics*. Routledge, 2013.
- [15] Fumitada Itakura. Minimum prediction residual principle applied to speech recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 23(1):67–72, 1975.
- [16] Daniel Jurafsky and H James. *Speech and language processing an introduction to natural language processing, computational linguistics, and speech*. 2000.
- [17] Chin-Hui Lee and L Rabiner. A frame-synchronous network search algorithm for connected word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(11):1649–1658, 1989.
- [18] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.
- [19] Paul Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Pattern recognition and artificial intelligence*, 116:374–388, 1976.
- [20] Greg Milette and Adam Stroud. *Professional Android sensor programming*. John Wiley & Sons, 2012.
- [21] Fabrizio Morbini, Kartik Audhkhasi, Kenji Sagae, Ron Artstein, Dogan Can, Panayiotis Georgiou, Shri Narayanan, Anton Leuski, and David Traum. Which asr should i choose for my dialogue system. In *Proceedings of the 14th annual SIGdial Meeting on Discourse and Dialogue*, pages 394–403, 2013.
- [22] norconex Faculties. norconex, <http://www.norconex.com/collectors/collector-http/configuration>.
- [23] Vimal P Parmar and C K Kumbharana. Study existing various phonetic algorithms and designing and development of a working model for the new developed algorithm and comparison by implementing it with existing algorithm (s). *International Journal of Computer Applications*, 98(19):45–49, 2014.
- [24] Frankie Patman and Leonard Shaefer. Is soundex good enough for you? on the hidden risks of soundex-based name searching. *Language Analysis Systems, Inc., Herndon*, 2001.
- [25] Lawrence Philips. The double metaphone search algorithm. *C/C++ users journal*, 18(6):38–43, 2000.

BIBLIOGRAPHY

- [26] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.
- [27] Lawrence R Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*, volume 14. PTR Prentice Hall Englewood Cliffs, 1993.
- [28] Advanced Web Ranking. Google organic ctr study 2014. 2015.
- [29] Johan Schalkwyk, Doug Beeferman, Françoise Beaufays, Bill Byrne, Ciprian Chelba, Mike Cohen, Maryam Garret, and Brian Strophe. Google search by voice: A case study.
- [30] Dat Tat Tran. *Fuzzy Approaches to Speech and Speaker Recognition*. PhD thesis, university of Canberra, 2000.
- [31] Charl Van Heerden, Johan Schalkwyk, and Brian Strophe. Language modeling for what-with-where on goog-411. 2009.
- [32] Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. Sphinx-4: A flexible open source framework for speech recognition. 2004.
- [33] Stephen John Young, NH Russell, and JHS Thornton. *Token passing: a simple conceptual model for connected speech recognition systems*. Citeseer, 1989.
- [34] Andrej Zgank and Zdravko Kacic. Predicting the acoustic confusability between words for a speech recognition system using levenshtein distance. *Elektronika ir Elektrotechnika*, 18(8):81–84, 2012.
- [35] Bartosz Ziółko, Jakub Gałka, T Jadczyk, and D Skurzok. Modified weighted levenshtein distance in automatic speech recognition. In *Proc. of 16th Conf. on Applications of Mathematics in Biology and Medicine*, pages 116–120, 2010.

Appendix A

List of Acronyms

ASR Automatic Speech Recognition

ASU Automatic Speech Understanding

WER Word Error Rate

OOV Out of Vocabulary

GDK Glass Development kit

SDK Software Development kit

Appendix B

Restaurant Names for Testing

Dapino, East Ocean, Food For Pleasure, Gröndals Sushikök, American Bar And Steakhouse, Boulebar Surbrunn, Cumin Club, Bishop Arms Söder, Ásens Krog, Minami, New Peking City, Oxelblom, Hamburgarprinsen, Indian Palace, Alviksgrillen, Kungfu Köket Sushi Och Dumplings, Kebab Kungen, Texas Longhorn Hornsbergs Strand, West Wong, Villa Fridhem, Queens Head, Potato Twister, Sankt Pauls Pizzabutik, River Ganges, Östermalms Fisk Och Pizza, Youran Sushi, Ecobaren På Centralbade, Espresso House Fridhemsplan, Falafelbaren, Govindas Indisk Vegetarisk, Grekiska Fast Food, Ashima Valhallavägen, Bröderna Bertlids, Ciao Ciao Due, Lilla Greta, Äppelvikens Krog, Mondo Tapas Och Bar, Noodle House, Ostra Matsal Och Bar, Happy India, Il Tempo, Kalaset Strand, Tea And Toast, Wkb Södermalm, Juiceverket Götgatsbacken, Vendetta, Prime Burger, Panda Express, Shahrzad Persiskt Kök, Onzbeh Grill, Doctor Salad, Frosseri, Great India, Atlas, Björk Bar Och Grill, Cirkus, Lorenzo, Golfbaren Kristineberg, New Tokyo, Oh Mamma Mia, Habibi Bar Och Café, Juiceverket Götgatsbacken, Karla Café, Tvátusen Å En Natt, Wkb Marieberg, Videgården, Pizzabutik Stora Essingen, Samrat Of India, Rasmus Grill Brommaplan, Fair View House, Gröna Dalen, Alviks Matbod, Adam Och Albin Matstudio, The Flying Horse, Banthai Fast Food, Angry Diner, Brother Tuck, Little Persia, Maharajah, Narknoi, Oxid Bar Tapas, Hoummus Libanesisk, Koreana, The Rice, Vita Italiana, Panini Birger Jarlspassagen, Syster O Bror, Spisa Hos Helena, Rama Kummin Koriander, Chez Betty, St Görans Pizza Och Salladsbar, Sankt Pauls Pizzabutik, Aberdeen China Restaurant, Fatbursbrunns, San Leandro Tapas, Lion Bar, Medo Sushi, Nancys Fresh food, Oriental Sushi, Halv Grek Plus Turk, Kimchi, Thai Oasen, Piazza Italia, Ramblas, Taco Bar, Ashoka, Butsaya Thai Take Away, Blue Chili, Debaser Humlegården, Mississippi Inn, Melanders Fisk Och Vilt, The Central Bar På Götgatan, Taste Of Thai, Pasta Gudfadern, Planet Food, Side Track Bar, Stekpannan, Den Gyldene Freden, Rost Stockholm, Pizza House, Caina, Cuckoo, Libanais Soir, Leas Mat, Mormors Kök Polsk, Hamburger Bar, Hilma Sofia, Tellus Pizza, Svartengrens, Pong Hornsberg, Vinkäl-

APPENDIX B. RESTAURANT NAMES FOR TESTING

laren Grappe, Teatergrillen, Kungsholmens Bowling, Husmanskosten, Ourkitchen, Divinus Foods, Edo Japansk, Farang, Gyrella, Adam och Albin Matstudio, Bun Bun Foodtruck, Curry Spice City, La Bodega Tapas Bar, Matprofilen, Nagano Japansk, Star Bar, Kebab Kungen, Två Drakar, Sibriens Soppkök, Stora Vikingen, Indian Stories, Adventure Education Of Europe, Äppelvikens Krog, Big Ben Pub, Sushi Now, Rost Stockholm, Stolen, Tasty Thai, Zink Grill, Bar Central, Un Poco, Rice Asian, Pong Buffé, New Peking City, Lucky City, Lilla Garden, Baron Steakhouse, Charles Dickens, En Till Grek, Esplanad Pizzeria, Folkunga Sushi Bar, Hint, Jimmy's Marina Steakhouse, Mejram, Opal Bar Bistro Söderhallarna, Solidaritet, Wkb Marievik, Hung Fook, Två tusen Å En Natt, Queens Head, Noodle House, Amazing India, Black And Brown Inn, Ecobaren På Centralbadet, Firewall, Carotte Catering, Gyrella, Indira, Kina Thai Gården, Lucky Wok, Nha Trang, Pärla Sushi, Quiche Company I Stockholm, Spice Garden, Tokyo By Night, Noor Halal, Ät Gott, Bönan Och Prinsen, Drottning Kebab, Gingerella, Hubertus, Karlssons Café, Medelhavets Kolgrill, Norrport, Pub Diset, Rival Hotel, Sushi Yama, Undici, Zuzhi Sushibar, Lelevo, Arena Globen, Capital, Därmed Pasta, Esperanza, Henricks, Kung Albert, Linje Tio, Noodle Mama, Ramblas Mariatorget, Stockholms Matvarufabrik, Villa Fridhem, Little Persia, The Picts Bar, Thai Oasen, Amys Café, Crazy Daisy, Essinge Wok, Grottan Pub, Jimmy's Steakhouse City, Kung Carls Bakficka, Mumbai, Pocket, River Ganges, Tasty Thai, Yuan Da Du, Itamae, Thai Oasen, Don Ramon, Art Sushi, Bluesbaren, Flower City, Jing Sushi Och Café, Lorenzo, Oriental Sushi, Risbergs, Texas Longhorn Sjöstaden, Corner Club, Kung Albert, Kung Albert, El Clasico, Alviksgrillen, Ballbreaker Kungsholmen, Bar Cobra, Crazy Horse, Eriks Bakficka, Golden Hits, Hamburgarprinsen, Honey Honey, Indiska Källaren, Kacha Thai Dining, Lemon, Linas Bar, Ming Garden, My Food Story, Nickys Thai, Pizza Butik Specialisten, Prime Burger, Pub Golden Glandore, Riche, Rytterbaren, Seoul, Sushi Bar Niko, Sushi Zen, Udon Sushi, Vasaaptit, Wok Take Away, Yanbos Kök, Big Mamma Chinese, Linkbar, Tomato I Stockholm, Josefina, Taste Of Vietnam, Tokyo By Night, Pure Nightclub, Shimla, Creperie Fyra Knop, Loopens Marina, Årstadals Deli, Mosebacke Etablissement, Nybroe Smörrebröd, Hosteria Tre Santi, Internationella Pressklubben, Korvmagasinet På Nytorget, Västbergakällaren, Dramatenterrassen, Debaser Humlegården, Frökens På Söder, Alfredos Bakficka, Brasserie Stuket, Bryggeriet.

Appendix C

Restaurant Names and Their Phonetic Sounds

| Phonetics Symbols | Restaurant Name | Phonetics Symbols | Restaurant Name |
|------------------------------|------------------------------|-------------------------------|-----------------------------|
| 'kɔrnər klʌb | Corner Club | Halv Grek plʌs tɑrk | Halv Grek Plus Turk |
| klʌŋ 'ælbərt | Kung Albert | 'kɪmʃi | Kimchi |
| klʌŋ 'ælbərt | Kung Albert | tai Oasen | Thai Oasen |
| el Clásico | El Clasico | pi' æzə 'tæliə | Piazza Italia |
| Ballbreaker Kungsholmen | Ballbreaker Kungsholmen | Ramblas | Ramblas |
| bər 'koʊbrə | Bar Cobra | 'təkoʊ bər | Taco Bar |
| 'kreizi hɔrs | Crazy Horse | Ashoka | Ashoka |
| 'eriks Bäckficka | Eriks Bäckficka | Butsaya tai teik ə 'wei | Butsaya Thai Take Away |
| 'goʊldən hits | Golden Hits | blu 'ʃili | Blue Chili |
| 'həni 'həni | Honey Honey | də' beisər Humlegården | Debaser Humlegården |
| Indiska keiällaren | Indiska Källaren | 'misi' sɪpi in | Mississippi Inn |
| Kacha tai 'daiɪŋ | Kacha Thai Dining | 'meləndəz fisk ɔk Vilt | Melanders Fisk Och Vilt |
| 'lemən | Lemon | ðə 'sentrəl bər plʌ ðjɔtɡatan | The Central Bar På Götgatan |
| 'linəz bər | Linäs Bar | teist əv tai | Taste Of Thai |
| miŋ 'ɡɑrdən | Ming Garden | 'pɔstə Gudfadern | Pasta Gudfadern |
| mai fud 'stɔri | My Food Story | 'plænət fud | Planet Food |
| 'nikiz tai | Nickys Thai | said træk bər | Side Track Bar |
| praɪm 'bɑrgər | Prime Burger | Stekpannan | Stekpannan |
| pʌb 'goʊldən Glandore | Pub Golden Glandore | den Gyldene Freden | Den Gyldene Freden |
| riʃ | Riche | rost 'stɔk hoʊlm | Rost Stockholm |
| soʊl | Seoul | 'piʃsə haʊs | Pizza House |
| 'suʃi bər 'nikoʊ | Sushi Bar Niko | Caina | Caina |
| 'suʃi zen | Sushi Zen | 'kuku | Cuckoo |
| Udon 'suʃi | Udon Sushi | Libanais sɔɪr | Libanais Soir |
| wɔk teik ə 'wei | Wok Take Away | liz mæt | Leas Mat |
| Yanbōs keiōkei | Yanbōs Kōk | 'hæmbərgər bər | Hamburger Bar |
| big 'mama ʃai' niz | Big Mamma Chinese | 'hilmə sɔʊ' fiə | Hilma Sofia |
| tə' mei, too ai 'stɔk hoʊlm | Tomato I Stockholm | Tellus 'piʃsə | Tellus Pizza |
| joʊsi' fiənə | Josefina | Svartengrens | Svartengrens |
| teist əv vi, et' nɔm | Taste Of Vietnam | pɔŋ Hornsberg | Pong Hornsberg |
| 'toʊki, ɔʊ bai nait | Tokyo By Night | Teatergrillen | Teatergrillen |
| pjɔr 'nait klʌb | Pure Nightclub | Kungsholmens 'boʊliŋ | Kungsholmens Bowling |
| Årstadals 'deli | Årstadals Deli | Husmanskosten | Husmanskosten |
| Mosebacke Etablissement | Mosebacke Etablissement | Ourkitchen | Ourkitchen |
| Nybroe Smörrebrödi | Nybroe Smörrebröd | Divinus fudz | Divinus Foods |
| Internationella Pressklubben | Internationella Pressklubben | 'idoʊ Japansk | Edo Japansk |
| Korvmagasinet på Nytorget | Korvmagasinet På Nytorget | Farang | Farang |
| viästbergakällaren | Västbergakällaren | Gyrella | Gyrella |
| də' beisər Humlegården | Debaser Humlegården | 'ædam ɔk 'ælbɪn Matstudio | Adam och Albin Matstudio |
| Frökenz på esödar | Frökens På Söder | bʌn bʌn Foodtruck | Bun Bun Foodtruck |
| 'æli' freidoʊz Bäckficka | Alfredos Bäckficka | 'kʌri spais 'siti | Curry Spice City |
| 'bræsəri Stuket | Brasserie Stuket | la boʊ deigə 'təpəs bər | La Bodega Tapas Bar |

Figure C.1: Restaurant Name

APPENDIX C. RESTAURANT NAMES AND THEIR PHONETIC SOUNDS

| Phonetics Symbols | Restaurant Name | Phonetics Symbols | Restaurant Name |
|-------------------------------------|-------------------------------|--------------------------------------|----------------------------------|
| 'indian 'stɔːrɪz | Indian Stories | mi'namɪ | Minami |
| æd'ventʃər ,edʒə 'keɪʃən əv 'jɔːrəp | Adventure Education Of Europe | nu 'pi'kiŋ 'sɪti | New Peking City |
| big ben pʌb | Big Ben Pub | 'indian 'pæləs | Indian Palace |
| 'sʊʃɪ nəʊ | Sushi Now | Kʊŋfu keɪəkət 'sʊʃɪ ək 'dʌmplingz | Kungfu Köket Sushi Och Dumplings |
| rost 'stɔk, hoolm | Rost Stockholm | Kebab Kungen | Kebab Kungen |
| 'stoolən | Stolen | 'teksəs 'lɔŋ, hɔːn hɔːnsbergs strænd | Texas Longhorn Hornsbergs Strand |
| 'teɪstɪ tʰaɪ | Tasty Thai | west wɔŋ | West Wong |
| zɪŋk grɪl | Zink Grill | 'vɪlə frɪdhem | Villa Fridhem |
| bər 'sentɹəl | Bar Central | kwinz hed | Queens Head |
| ən 'pɔːkoo | Un Poco | pə'teɪ, too 'twɪstər | Potato Twister |
| raɪs 'eɪʒən | Rice Asian | Sankt pɔlz Pizzabutik | Sankt Pauls Pizzabutik |
| pɔŋ bʌfɛ | Pong Buffé | 'rɪvər 'gændʒɪz | River Ganges |
| nu 'pi'kiŋ 'sɪti | New Peking City | ˌɛ'spresoo haos frɪdhemsplan | Espresso House Fridhemsplan |
| 'lʌki 'sɪti | Lucky City | Bröderna Bertlids | Bröderna Bertlids |
| 'lɪlə 'gɑːdən | Lilla Garden | tʃəʊ tʃəʊ du | Ciao Ciao Due |
| 'bæərən 'steɪk, haʊs | Baron Steakhouse | 'lɪlə 'grɪtə | Lilla Greta |
| tʃɔːrɪz 'dɪkənʒ | Charles Dickens | Äppelvikens krog | Äppelvikens Krog |
| en tɪl grek | En Till Grek | 'mandoʊ 'tʌpəs ək bər | Mondo Tapas Och Bar |
| hint | Hint | nudəl haʊs | Noodle House |
| 'dʒɪmɪ'ɛs mə'rɪnə 'steɪk, haʊs | Jimmy's Marina Steakhouse | 'hæpi 'ɪndiə | Happy India |
| Mejram | Mejram | ɪl 'tɛm, pɔː | Il Tempo |
| 'oʊpəl bər 'bɪstrɔː ɛsədərhallarna | Opal Bar Bistro Söderhallarna | tɪ ænd toʊst | Tea And Toast |
| hʌŋ fook | Hung Fook | ven'detə | Vendetta |
| 'ti'viätusen Å en Natt | Tvätusen Å En Natt | praɪm 'bɜːɡər | Prime Burger |
| kwinz hed | Queens Head | 'pændə ɪk'spres | Panda Express |
| 'nudəl haʊs | Noodle House | ʃər'zæd Persiskt keɪəkɛɪ | Shahrazad Persiskt Kök |
| ə'meɪzɪŋ 'ɪndiə | Amazing India | 'daktər 'sæləd | Doctor Salad |
| blæk ænd braʊn ɪn | Black And Brown Inn | frosseri | Frosseri |
| 'faɪrwəl | Firewall | greɪt 'ɪndiə | Great India |
| Carotte 'keɪtərɪŋ | Carotte Catering | 'ætɪləs | Atlas |
| ɪn'dɪrə | Indira | 'bi'dʒeɪərək bər ək grɪl | Björk Bar Och Grill |
| Kina tʰaɪ dʒɪlɑːrden | Kina Thai Gården | ˌsɪrkʊs | Cirkus |
| 'lʌki wɔk | Lucky Wok | lə'renzoo | Lorenzo |
| Nha træg | Nha Trang | Golfbaren Kristineberg | Golfbaren Kristineberg |
| kɪʃ 'kʌmpəni əɪ 'stɔk, hoolm | Quiche Company I Stockholm | nu 'toʊki, oo | New Tokyo |
| 'toʊki, oo bər nait | Tokyo By Night | oo 'mama 'miə | Oh Mamma Mia |
| 'noʊər Halal | Noor Halal | hə'bɪbi bər ək kʌfɛ | Habibi Bar Och Café |
| 'kɔːlsənʒ kʌfɛ | Karlsöns Café | 'kɔːlə kʌfɛ | Karla Café |
| pʌb dɪsɛt | Pub Diset | 'ti'viätusen Å en Natt | Tvätusen Å En Natt |
| 'raɪvəl hoo'tel | Rival Hotel | Pizzabutik 'stɔrə ɛssɪŋən | Pizzabutik Stora Essingen |
| ə'rɪnə glɔben | Arena Globen | Samrat əv 'ɪndiə | Samrat Of India |
| 'kæpətəl | Capital | fɛr vju haʊs | Fair View House |

Figure C.2: Restaurant Name