A tool for linking Bliss symbols to WordNet concepts

Sandra Derbring

Uppsala University
Department of Linguistics and Philology
Språktetnologiprogrammet
(Language Technology Programme)
Master's thesis in Computational Linguistics

Supervisors:
Mats Dahllöf, Uppsala University
Katarina Mühlenbock, DART
Abstract

This thesis describes the development of an application that aims at linking symbols used for alternative communication with their corresponding concepts. The application would be a contribution to an ongoing project at DART (a data resource team for disabled people) concerning concept coding and a help for maintaining ontology resources. The user will choose a set of image files that represent symbols as input for the application. The file names, which represent the concepts of the symbols, are then adjusted to suit WordNet, looked-up, tagged and ranked. A frequency for each part of speech of a file name is retrieved and the most frequent is presented to the user along with its most probable definition. Together with the image of the symbol, the user is supposed to decide if the system’s choice is correct or else correct it through the interface provided.

The application’s results was compared to a gold standard, consisting of the intended part of speech to each word. Factors that affected the results were the consideration of verb markers, the editing of the file names to suit WordNet and the development of a more sophisticated ranking method than an early naïve approach. A smaller set of test data was used to validate the results.

The results show that the ranking scores a recall at approximately 90% for each part of speech. The words that are not tagged are functional words or domain specific words that WordNet does not store. The precision score for nouns is almost perfect while the scores for adjective and adverb are lower due to the fact that some functional words are tagged here. The ranking method gives more reliable results than its naïve predecessor. Additions to the present ranking method and more extensive tests may improve the results.
List of Figures

2.1 CCF ................................................. 12
2.2 SYMBERED ........................................ 12
2.3 Bliss-words. Left to right: afraid, compare, finally, drive ....... 15
3.1 Synsets for the word 'dog' .................................. 18
3.2 WordNet’s frequency file ..................................... 19
4.1 An activity diagram .......................................... 24
4.2 StoCC .................................................. 25
4.3 StoCC processing a range of words ......................... 25
List of Tables

3.1 Different kinds of C-words ........................................... 19
4.1 Adjustment of file names ........................................... 27
4.2 An example of a post in the output file .......................... 27
4.3 Search results for 10 different words ......................... 29
5.1 Edited folder with verb markers ................................. 32
5.2 Verb markers considered ........................................... 32
5.3 Edited folder .......................................................... 33
5.4 Testdata .................................................................. 34
5.5 A test case of the system’s ranking results ................... 34
B.1 Instructions ............................................................. 41
B.2 Error handling .......................................................... 41
Acknowledgement

I would like to thank my supervisor Mats Dahllöf for advice and guidance in the work with this thesis.

Thanks are also due to everyone at DART, especially Katarina Mühlenbock for offering the opportunity and Mats Lundälv for his support, feedback and providing of material.

Further, thanks to people who have responded to queries on the NLTK and the STPKOM mailing list.

Finally, thank you to Mattias and Ingrid for their proofreading and feedback.
1 Introduction

Many people with communicative or cognitive difficulties use augmentative and alternative communication (AAC). One way of substituting speech is to communicate with artificial graphical symbol systems instead of natural language. In the last years, there has been a growing interest in improving the situation for users of symbol systems by facilitating their access to web services. DART, a computer resource centre (DAAtaResursTeam) for disabled people in Gothenburg, has been participating in several projects concerning these matters.

1.1 Purpose

This aim of the thesis is to describe the development of a system that links symbols that are used in augmentative and alternative communication to concepts in WordNet, a lexical database developed at Princeton University. WordNet is a formal representation of a set of concepts and their relationships to each other. This will hopefully facilitate maintenance of the corresponding ontology resources in an ongoing concept coding project conducted by DART. The purpose of the application is to present the user with the parts of speech and definitions of a set of symbols, represented by image files. The aim is to find the most probable part of speech and definition for all files, where the file name represent the corresponding word, and let the user edit them if necessary.

1.2 DART

DART was established in 1988 as a part of the Regional Children’s Habilitation Unit, in the Section for Neurology, Neuropsychiatry and Habilitation at the Department of Paediatrics of Sahlgrenska University Hospital. About 12 staff members - speech therapists, occupational therapists, pedagogues and engineers - work interdisciplinary to support children with special communication needs. DART works on three levels. The team provides assessments and counselling for people needing tailored computer systems or other alternative means of communication. It also educates teachers, parents and therapists of a child with special needs about different systems or solutions for that individual child. A third area is research and development of software and methods in this field.
1.3 Outline

This thesis describes the development of a system that links symbols used for alternative communication to their concepts. Chapter 2 gives an overview of previous projects and research on concept coding along with presentation of resources used throughout the thesis work. In chapter 3, the issues that have to be handled, encountered in the preparatory work, are defined. Chapter 4 describes the implementation of the system and the creation of the graphical user interface developed to facilitate the usage. In chapter 5, a description of the results and factors that affected them can be found together with a test case. Chapter 6 holds a discussion around the results as well as the different parts of the application and suggests some further improvements along with the final conclusions.
2 Background

Many people with language impairment communicate with artificial graphical symbol systems instead of natural language. There are several different systems, Blissymbolics, Pictogram and PCS (Picture Communication Symbols) perhaps being the most common ones. All those systems, together with other speech substitutes, are said to be augmentative and alternative communication (AAC). Some of them have a more complicated syntax and grammatical structure than others, but in all systems, the symbols correspond to semantic concepts. In this chapter, a description of the works and developments important for the theory behind this thesis is presented. Projects that DART has participated in are illustrated along with more general theories of semantics. The structure of Blissymbolics is explored, and so is the structure of WordNet and of the file names used for this task. A list of important concepts and terminology is provided for the user in Appendix refterminology.

2.1 Previous Projects

In the last years, DART has - among other things - focused on projects concerning access to Internet services for symbol language users. The team have participated in the World-Wide Augmentative and Alternative Communication project (WWAAC) and created the Concept Coding Framework Interest Group, whose work the symbol concept editor project (SYMBERED) is based on. SYMBERED aims at developing an editing tool for symbol based languages and at converting concepts to representation in different languages. A subtask is to convert a concept to a symbol and to assign a code to each concept, defining its meaning; so called concept coding. This is now done by hand but it is desirable to make it at least a semi-automatic process. As creating this system is the subject of this thesis, these projects are described below.

2.1.1 WWAAC

The World-Wide Augmentative and Alternative Communication project aimed at making the Web accessible to people with communication impairment. Almost all ways of using the Web today is based on reading and writing as the information is text-based. This excludes many people, for whom the Web really could be a resource, particularly those with difficulties expressing themselves, speaking, perceiving and remembering.

The WWAAC project would therefore develop tools to increase the access to Web services concerning communication. The project has been worked on by a number of actors, each in charge of a different part. DART has been re-
sponsible for the navigation and data management support. During the project, tools for text and task support, standards for encodings and guidelines for web presentation for the target user groups were implemented.

Among the services developed, the emphasis lay on the most common web activities; e-mail communication and reading information on the web. Adjustments were made to personalize the web browsers and to make it possible to communicate via symbols (WWAAC Final Report, 2004).

The project aimed at creating a system allowing symbol users to communicate with non-symbol users. It used symbol-to-speech and symbol-to-text rules, which produces correct speech or text output if the symbols come in correct order. To make it easy for users to structure their messages in a syntactically correct way, a solution with speech synthesis and a limited list of correct structures has been implemented. The user has thus a fixed number of sentences to choose from, each word possible to exchange for another. Speech synthesis can be used to listen to the resulting sentences.

Another task was to get speech/sound representation for the different technologies - web browsers and email programs - to support the user's navigation in the different systems and their subtasks and flexible support for different input possibilities, which depends on how the user communicates with the computer - a pointing device, the keyboard or a switch control.

The achievement most relevant for the present task, was the development of the concept coding framework (CCF) (Lundälv et al, 2006). One of the aims - to share and communicate with symbols over the web - is associated with difficulties as there are no encoding standards. As a solution to this, a concept coding framework was introduced. It contains a basic set of concepts built-up from wordlists in a number of languages and from the symbol systems Blissymbolics, Pictogram and PCS. For the project, this was supporting the linguistic structures of the e-mail communication (WWAAC Final Report, 2004) - and more generally, it has become a project of its own, developing a vocabulary in the form of two reference ontologies (a representation of sets of concepts and their relationships in a specific domain).

2.1.2 CCF

The Concept Coding Framework (CCF) is in short a structure for handling concept based language in a standardised way and to maintain a restrictive vocabulary. As an offspring from the WWAAC project, it was created to make it easier for users of different symbol based systems to communicate with each other over the web. Since there are many encoding standards that differ from each other, CCF will represent concepts in many different languages, both text and symbol based (Lundälv et al, 2006).

As existing systems use different technologies, CCF is aiming to be a bridge between them - and not a replacement - and is therefore designed to adjust to all standard technologies as shown in figure 2.1. CCF focuses on being used over the Internet for web services and activities, but can also be integrated with software and systems to support or integrate features (Lundälv et al, 2004).

CCF is based on two reference ontologies, Base Reference Ontology (BRO) and Complementary Reference Ontology (CRO). BRO is based on WordNet and built-up similarly with concept definitions and synsets with relations. CRO
contains words not found in WordNet, like pronouns, prepositions, conjunctions and other function words. These are connected to other ontology resources, like different language representations, symbol system bases and a concept code definition list. Words are converted into a semantic content in form of codes as in example 1 and 2 and transformed into the ontology best suited. The first number in the code represents the word’s part of speech (1-4 represents noun, verb, adjective and adverb, respectively). The two last numbers stands for the concept’s definition (Lundalv et al, 2006).

(1) cc-great-4004
(2) cc-coffee-1001

Data is read and saved into Resource Description Framework format (RDF). As of today, the database contains about 600 concepts, the small number being due to lack of structures and tools to update it automatically. The work described in this thesis aims at creating such a tool and thus develop this database.

2.1.3 SYMBERED - a symbol concept editor

SYMBERED is based on results from the WWAAC project, especially from the CCF work with standard concept codes, which this project had a close cooperation with. This Nordic project aimed at developing a user friendly editing tool to create web pages based on concept coding. This will lead to the possibility to show web material with adjustable graphic symbols for people using alternative communication. Adults and children with communication difficulties will hence have an area to share and take part of information, educational material and literature. The material is based on Swedish, Finnish and English texts connected with concept codes, which can be converted into different graphic symbol systems or text representations, as shown in figure 2.2 (Lundalv et al, 2006). Initially, this tool will support a limited concept vocabulary in six natural languages and two symbol systems. This will be possible to extend later. The conversion between different concept representations or from the concept codes to a specific representation is only made concept for concept and presupposes no grammatical or syntactical processing. This is often sufficient to understand the meaning of an utterance but if a complete translation is wanted, it will need further adjustments.
Altogether, the SYMBERED project has created a set of tools to develop and a possibility to share knowledge about concept coding and its possibilities. This has also become a common resource for symbol support which can be used in several adaptions and further projects. The test texts, tools and documents will be published on a project web site, the resources will also be distributed as part of service packages. Along with the tool itself, a concept and representation manager was developed to maintain the concept and symbol resources needed.

2.2 Resources

The idea of using concept codes is that it should be possible to represent things or phenomena independent of language or means of communication. The thought that the semantics could be expressed without regard to the syntax is already adopted by web technology, where a semantic web now is developing. The content of a web page should be defined in semantics instead of syntax, allowing information to be searchable and easily processed. The semantic web uses the same tools as the CCF group - RDF to mark up the information and ontologies to describe different concepts - and is an interesting project to learn from in the further work with marking up concepts.

In this section, some research on Blissymbolics language is also provided. The application resulting from this project is supposed to assign Blissymbols to concepts. Knowledge of Blissymbolics is not needed for the actual ranking implementation, as this will not be a matter of analyzing images. However, to know whether or not the category assignment was correct was necessary to adjust the code appropriately.

2.2.1 Semantic Web

The semantic web can be described as an extension of the World Wide Web, where the web content is defined in terms of meaning. For a given domain, concepts and relationships are formally described.

The aim of creating a semantic web is to make information on web pages understandable for machines and not only for humans, as the case is today. In doing this, computers could perform a lot of the tasks today mainly involving humans, like searching and combining information. This could for example facilitate sharing experimental data for scientific purposes.

Markup and use

Today, the World Wide Web is to a large extent based on documents written in HyperText Markup Language (HTML) that annotates the syntax of the text. In such a document, meta tags can be added to categorize the content. The semantic web goes further by publishing data in a Resource Description Framework (RDF), which is a language where the data is categorized in a way understood by computers (Lundälv et al, 2004). RDF expresses data models in terms of objects in relation to each other. In contrast to HTML, RDF can
describe things rather than documents and link together pieces of information
to an item distinct from others.

Using RDF in combination with other description technologies, Web Ontology Language (OWL) and Extensible Markup Language (XML), the semantic web creates descriptions representing and replacing Web documents’ contents. OWL is used to further describe content properties like relations, cardinality and other characteristics. XML has nothing to do with the semantic content, but defines a syntax for the structure of the content (Lundålv et al, 2004). Computers can hence process knowledge in a way similar to human deductive reasoning and inference. This will produce meaningful search listings and ease research processing.

A semantic web aims for a more usable Web and will provide RDF as a standard format, with tools to convert existing data. The document should be marked up (preferably automatically) with semantic information, whether it is metadata or meaningful content. A basic metadata vocabulary would give support to those creating new documents on how to mark up information so it can be used properly.

2.2.2 Blissymbolics - a logical language for an illogical world

The Blissymbolic language consists of a number of basic symbols, which can be combined to form new expressions (like compounds and phrases in natural language). Indicators state the grammatical form of the word or sentence. What makes it easy for people with communication disabilities is the structure and the simplicity of the symbols.

- Graphics
  Unlike natural language, Blissymbolics is a artificial language whose morphemes are made up by graphic symbols rather than sounds. All symbols are made from a number of geometric shapes. Beside the standard geometry, there are some additional shapes used, like international characters, digits, punctuation marks, arrows and pointers. Shapes and location matters greatly in determining a symbol’s meaning.

In forming a Bliss-character, both the placement and the spacing is important. Punctuation spaces can be used both for ending a sequence of characters and for separating characters to specify the meaning (for example in compound symbols). There are some general positions for different kinds of symbols, like pointers and indicators.

- Concept Characters
  The basic Blissymbol set consists of approximately 900 characters (Blissymbolics Communication International, 2004), which are used both as independent symbols and as components in words formed by multiple characters. Additions are sometimes made to this collection, but more often, new expressions are created by combining the existing characters.

There are a number of different types of Bliss-characters. Pictographic Bliss-characters describe concrete objects and the form of the symbol often resembles the physical form. Ideographic Bliss-characters often represent abstract concepts. Function words like the, this and a are called
arbitrary characters. When a symbol consists of two or more characters that have been merged to represent a new meaning, they are called composite. An example is toilet, which is made by the symbols for chair and liquid. Four different Bliss-words can be seen in figure 2.3.

Bliss-characters can have meaning both individually and in combination with other characters. When a unit has a semantic meaning, it represents a unique concept and can be associated with one word (“gloss”) in a natural language.

A Bliss-word may have synonyms to adjust its meaning to be more precise. This means there can be more than one Bliss-word representing the same concept. One reason for this is that Blissymbolics is an international language and there may be cultural differences between countries.

- Functional Characters
  When a Bliss-word consists of more than one character, the components are divided into classifiers, specifiers and modifiers (Blissymbolics Communication International, 2004). Unlike Swedish or English, the word opens with a classifier that decides the semantic or grammatical category. It is followed by a specifier which defines the word’s function or type. (English word houseboat where boat is the classifier would correspond to boat+house in Blissymbolics). Modifiers are sometimes added.

  There are a number of small Bliss-characters used to modify a Bliss-word. These modifiers are often seen as prefixes and they facilitate the possibilities to expand or change the meaning of a word. There are also special combine modifiers that allow the user to express whole new concepts just like a natural language allows creation of new words, compounds or expressions. There are different kinds of modifiers, grammatical, semantic and numeric. For example, a prefixed number signals the number of the item it describes while a suffixed number shows the ordinal number.

  In addition to modifiers, Bliss-words can also be more precisely expressed with help from indicators, semantic as well as grammatical. A character can therefore mean a lot of things, depending only on the indicator centred above. The three main parts of speech all have their own indicators. Verbs have action markers denoting verb tense or voice (active/passive) while a description marker indicates if a word is an adjective. A noun has no particular symbol, but a plural marker can be used to indicate more than one of an item. However, there is the thing indicator, telling us whether a word is an abstract or a concrete noun or if it is a nominalized adjective.
3 Tasks to Address

This chapter describes the problems and tasks that form the different part of the algorithm. There are four different main tasks outlined to be solved in terms for the application to run. The handling of the input is important as the file names need to be adjusted in order to be searched in the WordNet database. The ranking of the matches is a crucial part of the system and needs to be defined. The two last relevant issues are the presentation of the output and the choices for the user, respectively. Each of these matters will be discussed below.

3.1 Adjustments of Data

For this task, a set of symbols is used on the system. These symbols are represented with their corresponding file names when it comes to processing them in the application. The file names are looked up in the large WordNet database and further information is also retrieved from here. The relation between WordNet's structure and the file names' structure is therefore essential to the application. To use WordNet in combination with the application's script in Python, the WordNet for Python module was used. In this section follows a description of both the file names and the WordNet database and its Python module.

3.1.1 WordNet

WordNet is a lexical database, developed by Princeton University. It groups together synonym sets (words sharing the same concept, called synsets) of English words (Fellbaum, 1998). Each synset defines a unique concept and is linked to other synsets depending on lexical and semantic relations, like antonyms or hypo- and hypernyms. The hypo- and hypernym relationship group the words into hierarchies where the words at the same level are hypernyms of the same hypernym. Within a synset, all synonyms representing a concept are collected. If a word has more than one sense, it is to be found in more than one synset. For each sense in a synset, WordNet provides frequency score, which makes it easy to determine which sense is the most common.

Words in WordNet are lemmas, which means there are no morphological information in the search results unless the word has irregular forms. Furthermore, there is no etymological or pronunciation information. Neither does the database contain specialized vocabularies, but instead provides a range of common words.
Over 150,000 words combined in over 117,000 synsets make WordNet a large ontology resource for natural language processing, as an alternative to standard lexicons. The words are categorized by part of speech due to their different grammatical functions and rules. WordNet can be used online and is also free to download.

### 3.1.2 WordNet for Python

In figure 3.1, a search for the word 'dog' in the graphical interface of the WordNet's database on the web can be seen. Figure 3.2 shows an excerpt from the file taken from WordNet's data collection used in this work to collect information.

Each match in the first example is a synset. As can be seen, not only synsets with 'dog' as the first element are found. When using WordNet for Python, there are different notations to access the parts of a synset. The word is first checked to exist in the different part of speech dictionaries of the WordNet database. The search is directed into each dictionary separately, which means that for each symbol, there are four searches if there are no conditions that say otherwise. If there is a match in any of the dictionaries, the frequency count, the definition and the information about sense and offset are retrieved, as more detailed described in section 4.5. As seen in the example above, WordNet stores collocation with an underscore. Hyphens and apostrophes are kept as they are. Information about the part of speech comes with the first number after the percent sign. As in the example, 1 stands for noun, 2 for verb and 5 for adjective.

### 3.1.3 Structure of the File Names

When creating the linking between symbols and concepts, the names of the symbol files are the starting-point. There are 3340 .png files in the set supplied for this task, systematically named and easy to divide into different categories. An overview of these is listed below.

- **Verbs**
  All words that are verbs are marked with the suffix -to, which makes them easy to separate from the rest. The part of speech is thus decided. However, these words could also have other structures to handle for matching the best definition; words with explanations, synonyms or ambiguity.

- **Explanations**
  There are words marked with a hyphen and parentheses. Explanations within paranthesis are given for those words that can be ambiguous. Words within paranthesis may be descriptions of the words or additional concepts to explain.

(3) cold-(opposite_hot)
(4) fish-(food)
Noun

- (42)02084071 S: (n) dog (dog%1:05:00::), domestic dog (domestic_dog%1:05:00::), Canis familiaris (canis_familiaris%1:05:00::) (a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds) “the dog barked all night”

- 10114209 S: (n) frump (frump%1:18:00::), dog (dog%1:18:01::) (a dull unattractive unpleasant girl or woman) “she got a reputation as a frump”, “she’s a real dog”

- 10023039 S: (n) dog (dog%1:18:00::) (informal term for a man) “you lucky dog”

- 09886220 S: (n) cad (cad%1:18:00::), bounder (bounder%1:18:00::), blackguard (blackguard%1:18:00::), dog (dog%1:18:02::), hound (hound%1:18:00::), heel (heel%1:18:00::) (someone who is morally reprehensible) “you dirty dog”

- 07676602 S: (n) frank (frank%1:13:00::), frankfurter (frankfurter%1:13:00::), hotdog (hotdog%1:13:01::), hot dog (hot_dog%1:13:01::), dog (dog%1:13:01::), wiener (wiener%1:13:00::), wienerwurst (wienerwurst%1:13:00::), weenie (weenie%1:13:00::) (a smooth-textured sausage of minced beef or pork usually smoked; often served on a bread roll)

- 03901548 S: (n) pawl (pawl%1:06:00::), detent (detent%1:06:00::), click (click%1:06:00::), dog (dog%1:06:00::) (a hinged catch that fits into a notch of a ratchet to move a wheel forward or prevent it from moving backward)

- 02710044 S: (n) andiron (andiron%1:06:00::), firedog (fire-dog%1:06:00::), dog (dog%1:06:01::), dog-iron (dog-iron%1:06:00::) (metal supports for logs in a fireplace) “the andirons were too hot to touch”

Figure 3.1: Synsets for the word ‘dog’
Figure 3.2: WordNet’s frequency file

- Symbol synonyms
  When there are two symbols referring to the same concept, these are marked with an underscore character and a digit, which refers to the symbol intended. This is the artificial graphical symbol language’s equivalence to a natural language’s synonyms - there can be two pictures connected to the same concept.

  (5) surprise
  (6) surprise_1

- C-words
  Words beginning with a capital letter, from here on referred to as C-words, are considered a separate category. These words may not be a case to handle separately, but it must be confirmed that all these words have a counterpart in WordNet. C-words are distributed in a number of different types seen in table 3.1, among others proper nouns, geographical places, single letters and specific phenomena.

<table>
<thead>
<tr>
<th>Table 3.1: Different kinds of C-words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proper nouns</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>Adam</td>
</tr>
<tr>
<td>Balder</td>
</tr>
</tbody>
</table>

- Collocations
  A lot of the words are collocations, which here are marked with an underscore. In this category, grammatical markers like indicators and inflectional forms may be found as well as common English compounds.

  (7) baking_pan
• Concept Synonyms
  When a symbol corresponds to two or more different words, both repre-
  senting the same concept, these two synonyms are listed with a comma 
  between them.

(8) hot,spicy,peppery

• Others
  The words that do not fall into any of the above categories are without 
  special characters, lowercase and unambiguous. Some words do have a 
  hyphen, but only as a part of the words. Many of these words are also 
  grammatical inflections and markers.

(9) house-boat
(10) circle

As can be seen when the WordNet structure and the file name structure 
are compared, some adjustments are needed for the file names to match 
against WordNet. The words that are concerned are the concept synony-
ms, the verbs, the symbol synonyms and the explanations. How to solve this is 
discussed more in detail in section 3.1.4.

3.1.4 Adjustments

As seen in section 3.1.3, the file names are structured in different ways. The 
differences for WordNet to read this data, must for an application that is sup-
posed to communicate with WordNet result in a subtask of adjusting the file 
names to be searchable.

The verbs in WordNet are not marked with any suffix. The marker -to in 
the file names must thus be removed for the search. However, this grammati-
ical information will be helpful as it indicates a search only in the verb dictionary 
of WordNet, thus excluding all information about the word as a noun, adjective 
or adverb.

For the file names in the set that hold a hyphen and parentheses to be 
searchable, the special characters need to be eliminated. The only word that 
the system will try to match with WordNet is the main word. The words inside 
the parentheses may be used as a help in further search for the right definition.

The underscore and numbers that follow symbol synonyms could simply 
be removed and the search conducted only with the word. The numbers only 
denote that there is more than one symbol for the concept and can not be used 
to determine the concept's part of speech or definition.

The file names that in the symbol set begins with a capital may be lower-
cased when the search is conducted. However, the information could be kept 
to determine the definition in a later stage.

There are symbols representing several concepts and this is denoted in the 
file names as all words separated with a comma. WordNet will, naturally, only 
able to look for one word at a time. These file names will therefore be separated 
and searched for one by one.
As a file name could belong to more than one of these categories, the order of categories to adjust will be important. Furthermore, misspellings, mistypings and inconsistencies, like spaces after tokens, that were found need to be corrected in order to get the best possible input list.

3.2 Ranking

The application’s task is to tell which definition and which part of speech is most likely for a word. Therefore, the ranking was an essential part of the implementation. The algorithm will not be able to decide the correct part of speech for a symbol as a human may do, as the image is not analyzed. There is no point in creating a system performing haphazard guessing but to find the most probable alternative, the application will need some clues. A fair assumption is that a part of speech or a definition with higher frequency in a corpora is more probable than a low-frequent one. The application will therefore rank the alternatives based on their frequency. This information can be provided from WordNet. The application must thus be able to communicate with WordNet in a way that allows retrieving the frequency information for the words. DART wished to collect some other values that WordNet stores as well to use in their database.

3.3 Output

Part of designing the algorithm was to decide what to present to the user on screen, the ambition being to make the interface intuitive and easy to use. An important point is to make it possible to change the system’s suggested definition or part of speech, thus showing all alternatives. There are synonyms among the symbols as well as among the words. The application must be able to present the image together with its word and suggested part of speech or definition, for the user to be able to determine whether to change the suggestion or not.

Aside from the output immediately shown on screen, a file with all information must also be created, holding the information important to store in the database.

3.4 User’s Choices

An interrelated issue for the implementation was how much the user should be allowed to choose and how much should be predefined. A naturally essential function is that the user may choose which directory to read files from. For two reasons it will also be important for the user to specify how many words he or she wants to be shown at a time. Firstly, 3340 files will not be preferable to present with respect to the time this will take to load and process. Secondly, it gives the user an opportunity to adjust the number of files to overview after how much time he or she has.

As a user most likely will not edit too many files in one occasion, it is desirable with a saving feature, letting the user quit whenever a range of files
are edited and continue from that point at a later stage. It should thus be possible to specify a file and a destination to save to. As a means to prevent the user from losing data, the application may save automatically after each finished range of files.

Something else the user should be able to choose, is the number of alternative definitions to show. As the alternatives are ranged by frequency, it is probable that the correct definition is among the top matches. The application will benefit in both loading time and aesthetics from not needing to show all possibilities.

Requested from DART, the application will also feature a search for a particular word or words beginning with a particular letter. The intention is that a search for a token like 'car' will render all words containing this lexeme; 'car', 'car_racing' and 'cable_car', excluding lexemes like 'cartoon' or 'vicar', only holding the string 'car'.
4 Implementation

This section will describe how StoCC - Symbol to Concept Coupler - is constructed. The flow of the system is visualized in figure 4.1.

The user interacts with a Graphical User Interface (GUI) and performs different actions by choosing elements or pressing buttons. The GUI, written in PyGTK (Finlay, 2005), communicates with several methods inside its class as well as the class devoted to do the selection and ranking of the input, written in plain Python.

The input consists of a number of .png files in a folder of the user’s choice. The system converts the file names into strings in a suitable format for the application. The strings, which are the word representations of the symbols, are processed through the system and looked up in a dictionary. Information about part of speech, semantical meaning and frequency is collected and returned to the GUI and the user. The user will now be able to accept the system's rating, or to correct it.

The output is generated in two ways. The screen output is presented in columns with the image symbol in addition to the other information, as a help for the user to decide. The same information is also written to a file chosen by the user. If the user makes any corrections, this is written to the file.

The application can be divided into the following main parts:

- Graphical User Interface
- Preparing methods
- Tagging and ranking
- Methods managing the definitions
- Search methods

These parts are described in the following sections from an implementational point of view, with the exception of the GUI that is described from a user's perspective.

4.1 User’s Perspective

The idea of StoCC is to connect a set of symbols with their corresponding concepts from WordNet. The aim is to facilitate the maintenance of a database holding these concepts, used to produce translations between symbols and text. The system described in this thesis works on a set of symbols of the
user's choice. For each symbol in this set, the system tries to find a match in WordNet. For every match found, all possible parts of speech and all possible definitions are retrieved. This is presented on the screen and provided with functions for the user to correct or edit the information. The idea is that this tool should be a help for updating the database. However, as it for many words may be possible with multiple definitions and several parts of speech, it would be difficult to let the program be totally automatic as the meaning could be totally misleading in relation to the symbol. Thus the user must look at the result, possibly edit the information and save it. The saved data is later entered into the database.

Figure 4.2 shows a screen shot of the application's interface. This is what the user sees when the application is started. As a help for the user, a text is presented to the right to guide him or her through the process of running the system. As no input is loaded by default, the first instruction to the user is to open a folder with images. This is done from the Options menu and will show the user a dialog box from where to choose a folder or a file.

The menus also have options like save, documentation and help, the latter providing information about the application.

As the user has chosen a folder or a file, the next step is to decide if a particular word or letter is to be searched, or if the whole folder should be displayed. The application will also instruct the user to specify how many words to show per page and how many alternative definitions to show per word. To show every alternative and all words at the same time is not advisable as this would be very slow, at least if the folder contains many images. If the user chooses not to, or forgets, to enter those numbers, default values are used.
A range of symbols is then loaded into the page, as seen in figure 4.3. The application displays a progress bar representing the time it takes and information about which page you see and how many pages there are in total. A choice of showing 100 words per page for a folder with 450 images will give 5 pages. The images are now shown in the left column, accompanied by all possible parts of speech in the middle. The most frequent definition of the most probable part of speech is presented below. To the right, all alternative definitions are shown. If the user wants to use another definition, a check in one of the alternatives will change the presented meaning in the middle column. A change of part of speech will replace all definitions with appropriate ones in both columns.

The user can now proceed in three ways; viewing the next range of words if there are any - causing the current page to be saved automatically - quitting the application or doing another search. Each time a file is saved, information about which folder used and the number of words edited is stored in it. When quitting without having edited all files in the folder, this information is used to
4.2 Adjusting the Input

Before implementing any ranking methods, the file names of the supplied set of symbols were adjusted according to those findings in section 3.1 concerning misspellings, mistypings and inconsistencies. Along with this, a simple, unsophisticated tagging was done to provide each word with a part of speech. This list was then manually checked to give each symbol the intended part of speech. The adjustments of the file names were important to make the input the best possible. The tagged list was used as a means to improve the algorithm during the implementation, but more importantly, it was later used as a gold standard when the results were measured.

4.3 Preparing Methods

Preparations for the ranking and presentation are done in different parts of the implementation. All calculations and variables needed for later are done or set; the number of files and pages, the progress bar and the different modes among other things. A dictionary is created, which loads information about words, grammar and frequencies from a WordNet file and stores it. The GUI’s columns are prepared for new information and eventually, each word in the range specified by the user is sent to be tagged and ranked. The returned values are stored and passed along to those methods preparing for output.

4.4 Tagging and Ranking

As specified in section 3.1.3 and as can be seen in table 4.1, the names of the files are on different forms. The different structures could also be combined as in example 11. As explored in section 3.1.2, WordNet does not store words like this and thus an adjustment is necessary. Starting by removing the file extension and lower case all words, the different kinds of structures are handled in different ways. See table 4.1 for a listing of the adjustments. If the words are without any special characters, they are checked for beginning with a capital. Digits are removed from the file names containing them. Both verbs in a file name as in example 12 are marked with the verb marker to keep the grammatical information. If a file name consists of more than one word, either within parenthesis or separated with a comma, those additional words are stored separately from the main word.

(11) flower,plant-(green)

(12) book,reserve-to

The word is now passed along to the ranking. Basically, the different structures are treated the same, with the exception that in those cases there are additional words, these need to be considered in the ranking.
Table 4.1: Adjustment of file names

<table>
<thead>
<tr>
<th>File name</th>
<th>Adjustment</th>
</tr>
</thead>
<tbody>
<tr>
<td>sharp-(taste)</td>
<td>sharp [taste]</td>
</tr>
<tr>
<td>flower,plant</td>
<td>flower [plant]</td>
</tr>
<tr>
<td>accident_1</td>
<td>accident</td>
</tr>
<tr>
<td>read-to</td>
<td>read-to</td>
</tr>
<tr>
<td>Africa</td>
<td>africa</td>
</tr>
</tbody>
</table>

As the WordNet database lists lemmas, all file names need to be lemmatized before processed further. This is done alongside with checking which parts of speech the word belongs to. There are many words with meanings in several parts of speech and all those variations must be examined and presented to the user. For each part of speech the word is found to belong in, all synonym sets are examined. The synonym set consists of those words sharing a concept, which in this thesis is referred to as definition. A word can belong to several synonym sets and each synonym set shares an index number ranked by frequency. As a synonym set - from here on ‘synset’ - is investigated, this index number is used together with the word and the part of speech as an unique key to get the frequency value from the dictionary storing WordNet’s data. See figures 3.2 and 3.1 for illustrations of how the concepts are stored in WordNet. Alongside with these five parameters - word, part of speech, index number, definition and frequency - the offset and the synset number are collected. The offset is a ten-numbered value which uniquely defines the word’s version and the synset number is simply the element number of the word we are looking at in the synonym set. In example 13, looking for the word ‘family’, the synset number is 3.

(13) {class, category, family}

These last two are not presented to the user, but are needed for the DART database at a later stage. They are thus saved in the output file, table 4.2.

Regardless of how many synonym sets have been found, information about them all are stored and later used to get the alternative definitions. However, the highest ranked for each part of speech are stored together and used immediately.

For those file names having additional words, the process also involves considering those. This is done by checking if each synonym set contains any of those extra words. The synonym sets that do, are stored in the ranking list, regardless of their frequency. The difference would therefore be that while a

Table 4.2: An example of a post in the output file

<table>
<thead>
<tr>
<th>Word</th>
<th>Part of speech</th>
<th>Freq</th>
<th>Synnr</th>
<th>Offset</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>beautiful</td>
<td>adjective</td>
<td>25</td>
<td>1</td>
<td>238420</td>
<td>Delighting the senses...</td>
</tr>
</tbody>
</table>
single word only has one definition chosen for each part of speech it belongs to, the other words can have several. If a word starts with a capital, this is checked in the same way, thus making a better guess at words like 'shiva', that could be spelled 'Shiva' as well, having a completely different meaning.

As all top rankings have been collected, they are compared to each other in terms of frequency. The definition having the highest is selected as the most probable and together with its part of speech it is presented to the user. The other parts of speech are together with their first definition stored as alternatives. Both this information and information about all synonym sets that have been examined, regardless of their frequency - are returned to the GUI.

4.5 Methods Managing the Definitions

As the system has now chosen the information to present, this must be displayed to the user. The system starts by loading the image file belonging to the word and places it in the left column. As all images will be shown in this column, they adjust in size to the largest image. This makes all columns and rows the same size, giving a symmetric impression. The gloss of the definition most probable is collected from the dictionary storing all words that have been processed earlier, and added to the middle column. The other element needed in the middle column is a list of the parts of speech connected with the word. This list consists of a radio button for each part of speech, the most probable on top and at the bottom one to suit the possibility that neither of the parts of speech is the correct. The buttons are connected with a method allowing the gloss and the definitions to change if they are clicked. In the right column, a shortening of the glosses is necessary to keep the size of the rows down. On top, a check button representing the most probable definition is placed and below, a button for each alternative definition, retrieved from the large dictionary. All buttons are communicating with a method that will print its definition in the middle column. To facilitate the changing between parts of speech, all buttons are stored.

Changing from a part of speech to another on the user's request, needs consideration of several issues. The gloss in the middle column must be changed to the most probable definition of the new part of speech and all definitions in the right column must be replaced. The system also collects and stores all alternative definitions and their glosses, everything by consulting the large dictionary. If the user has specified a number of alternatives to show that is smaller than the total amount of alternative definitions, only as many as the user has requested are shown. The top button showing the most frequent definition of the part of speech chosen by the system is now replaced with the most frequent definition of the part of speech the user chose, with the correct gloss and the correct parameters. As the rest of the buttons are to be replaced, there are three cases possible. The 'new' part of speech may have fewer definitions than the part of speech to replace. This means for the system to replace as many as there are new and remove the remaining buttons. In the second and easiest case, there is an equal number of definitions for both parts of speech and the replacement is smooth. The third case, of course being that the 'new' part of speech has more definitions, requires adding new buttons after the existing
4.6 Search Methods

It is possible for a user to search for only a particular word or for all words of a particular letter. The searches have common features but differ in several approaches and are thus handled separately. The variables, necessary calculations and the creation of a dictionary are handled just like the preparations made for the default case.

A search for a word starts by creating a regular expression out of the entered word, to find the words that should be included in the search. A search for 'car' should not just find 'car', but also 'car_racing' and 'cable_car'. However, it is not supposed to find words like 'carousel', 'daycare' or 'vicar' even though these words contain the string 'car'. All words in the folder the user has chosen matching the expression are stored. They are passed along to be prepared for tagging and ranking and then processed as the default search.

Searching for a letter simply begins by looking through the folder and store those words beginning with the letter entered by the user. Controlling the range of words that should be shown per page, also these words are passed along to be prepared for tagging and ranking.

Table 4.3 shows the matching of 10 different words searched for in the system. The words 'ride' and 'open' show that all parts of speech for a word are matched, given that the system can not know what the user is looking for. The words 'ice', 'hard', 'old' and 'low' show that all versions of a word - different collocations and compounds, different meanings - are found in a search, which of course is desired of this feature. Furthermore, the word 'last' shows that also all symbol synonyms of a particular meaning of a word are found, which is very useful to get an overview.

<table>
<thead>
<tr>
<th>Search</th>
<th>Found files</th>
</tr>
</thead>
<tbody>
<tr>
<td>car</td>
<td>car, cable_car, car_racing</td>
</tr>
<tr>
<td>apple</td>
<td>apple</td>
</tr>
<tr>
<td>ride</td>
<td>ride-(horse), ride-to</td>
</tr>
<tr>
<td>open</td>
<td>open, open-to</td>
</tr>
<tr>
<td>ice</td>
<td>ice_cream, ice_cream-(cone), ice_cream-(bar), ice_hockey, ice_skates</td>
</tr>
<tr>
<td>hard</td>
<td>hard-(opposite_soft), hard-(opposite_new), hard_cheese</td>
</tr>
<tr>
<td>old</td>
<td>old-(opposite_new), old-(opposite_young), old</td>
</tr>
<tr>
<td>low</td>
<td>low, low_water, low_tide,ebb</td>
</tr>
<tr>
<td>last</td>
<td>last, last_1</td>
</tr>
<tr>
<td>honey</td>
<td>honey-(food), honey-(spread)</td>
</tr>
</tbody>
</table>
5 Results

To estimate the application's result, they can be compared to the gold standard that was made in the beginning of this work. In the gold standard, as described in section 5.1 below, each file name is assigned to the preferred part of speech for the intended symbol. This standard was used as a reference during the implementation as a means to prevent mistakes in the ranking. The aim is that the system should tag words to their most frequent part of speech. Even though this is accomplished, the results will never be 100% correct in comparison to the gold standard. There are two reasons for this. Firstly, all words will not be found in WordNet. Unusual or specific terminology will thus be tagged as unknown. Secondly, the preferred part of speech for a specific symbol is not always the most frequent part of speech for that word.

The results can thus not be expected to be perfect matches to the gold standard. They only show if the ranking method was effective - that is, if most of the words are in line with the gold standard. Below is demonstrated to which extent the application improved from its early stages and which factors that affected this. How to get the application to provide another part of speech in favour for the most frequent in those cases where it is desired, would probably require more information. Any solution to this may be an important key to further improvements.

The application also aimed at choosing the correct and most probable definition of a word. It is harder to evaluate if the method did well here, as several answers could be correct. As the symbols do have broader meaning than a written word, more than one definition could be considered correct as the first-hand choice. The file name corresponding to an English word may have many different meanings, but if the chosen definition is the correct one depends on the symbol image. The system may choose the most frequent definition, but still, it would have to be changed to suit the symbol.

Important to note is that the gold standard contains the same files that the application is trained on and the results can not be seen as an objective evaluation of the application. A smaller set of data that was not used in the training is therefore used to validate the results.

5.1 Gold Standard

To measure the results, a gold standard with the correct or preferred part of speech for each word was needed. This standard was also used as mentioned in section 4.1, as a help to improve the implementation. It did not just establish the parts of speech but also the consistency and spelling of the words. After
the words were normalized in the same way mentioned in section 4.5, a search in WordNet was performed to do a rough tagging. The search does not try to match all parts of speech the word belongs to. Instead it assumes that the first part of speech found is the correct one, looking in order of nouns, adjectives, adverbs and verbs. Each word is thus designated a part of speech. These pairings were later gone through manually to correct the parts of speech or the words where it was necessary.

5.2 The Improvements

The results were compared to the gold standard in different phases and it is interesting to see what improvements different solutions gave. It is, however, not possible to compare the results directly as the conditions changed between these phases. Thus, some of the comparisons made below can only be seen as an indicator of the improvements. There were three main factors that influenced the results; the development from a naïve to a more sophisticated approach, the editing of the file names and the consideration of verb markers. In all cases below where not stated otherwise, the verb markers are considered and the edited file names are used.

The implementation has gradually increased in complexity, but there is an apparent difference between the earliest approach and the latest. The early approach works similarly to how the gold standard was created. It stops looking when a match for a word is found and assigns it the first possible part of speech. In the later version, there are more sophisticated methods for matching a word against a part of speech. All words are tested against all parts of speech and all matches are ranked to provide the most frequent one. As seen in table 5.1, the naïve approach has better precision but a lower recall with the exception for nouns. This is not unexpected. Many words whose most frequent part of speech is something else will be found as nouns. However, the adjectives and adverbs that are found have a higher chance of being correct since they have proven not to be a noun. The sophisticated approach has a higher recall overall. The recall for nouns are slightly lower than for the naïve approach, but the precision is very high. Almost all nouns that are tagged are tagged correctly but not all nouns are found. The sophisticated approach finds a high measure of both adjective and adverbs compared to the naïve one, but not all of them are correct.

The set used for the system contains 3340 files. However, a file can hold more than one word, as in example 8. The files that primarily were used were the ones that were edited to better match against WordNet or to correct errors. The original file names were kept and is here used to show what improvement the editing gave rise to. Due to the editing, the numbers of words differ depending on which folder the system runs on as the words have been divided somewhat differently depending on synonyms and explanations. It is not possible to calculate the precision and recall for those words. The original folder is thus not that interesting in itself, and cannot be exactly compared to the gold standard, where the file names are corresponding to the edited folder. It is here shown only to reflect the improvements. In table 5.2, output from the different folders is shown. These numbers do not measure whether or not the
words for each part of speech is correctly tagged. It can only give an indicator of what the editing of the filenames might improve. Excluding verbs, that are predefined, more words are tagged for all parts of speech in the edited folder than in the original. Even if the differences may seem small, the total number of unknown words has decreased significantly. These statistics apply regardless of which approach used.

The verbs in the set are all marked with ‘-to’ to indicate their part of speech. These markers are not considered in the first phase of the implementation which means that all words are treated the same. The early naïve approach that stops looking when a match for the words is found does not tag even a third of the verbs correctly. The sophisticated approach that compares all matches to provide the most frequent one tags considerably more words than it should to verbs. In a later stage, the verb markers are taken into consideration. Those words are automatically tagged as verbs by both approaches and not searched for in any other part of speech. No other words are considered to be a verb. The naïve approach then tags all verbs correctly. The sophisticated approach tags 411 words as verbs out of the 418 in the edited folder. The increased number of matches for the sophisticated approach is due to the look-up. As opposed to the simplistic tagging, words predefined as verbs are not excluded from the matches to WordNet’s database and if the word is not found there, it is not tagged. In table 5.3, output with and without verb consideration can be seen. These results are not measured with regard to precision and recall, but they indicate that the naïve approach fails to find many of the verbs and that the sophisticated approach tags too many words as verbs.

<table>
<thead>
<tr>
<th>Table 5.1: Edited folder with verb markers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Naïve</td>
</tr>
<tr>
<td>Precision</td>
</tr>
<tr>
<td>Nouns</td>
</tr>
<tr>
<td>Adjectives</td>
</tr>
<tr>
<td>Adverbs</td>
</tr>
<tr>
<td>Verbs</td>
</tr>
<tr>
<td>Untagged</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5.2: Verb markers considered</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Unedited</td>
</tr>
<tr>
<td>Naïve</td>
</tr>
<tr>
<td>Total</td>
</tr>
<tr>
<td>Nouns</td>
</tr>
<tr>
<td>Adjectives</td>
</tr>
<tr>
<td>Adverbs</td>
</tr>
<tr>
<td>Verbs</td>
</tr>
<tr>
<td>Untagged</td>
</tr>
</tbody>
</table>
The results from the sophisticated approach in table 5.1 show that the system tags approximately 90% for each part of speech, verbs excluded. Given that the set holds several quite rare and domain specific nouns, those numbers are expected for this part of speech. It seems that several words are inaccurately tagged as adverbs and adjectives. This might be due to that a word often belongs to both of these parts of speech and WordNet's classification does not correspond to the symbols'. Furthermore, an adverb and an adjective may in Blissymbolics be represented by the same symbol. Another reason may be that many common prepositions are homonyms to adverbs and/or adjectives, which means that the spelling is the same but not the meaning and maybe not the part of speech. WordNet only stores words from the four largest parts of speech. This could case these words to be tagged as adverb or adjectives and by that increasing their total number.

### 5.3 Validation

To validate the results shown in the previous section, the application is run on a smaller set of data. The set consists of 384 words, which is around a tenth of the training set. However, the distribution of the parts of speech does not correspond to the gold standard in the larger set. The number of adjectives and adverbs are significantly smaller and expectantly, the nouns stand for the larger part. The test data consists of recently created Blissymbols that are meant to be incorporated in the total vocabulary. None of those symbols are functional words and therefore, the number of 'unknown' words are almost nonexistent. The size of the set and the distribution should give some differences in the results compared to the training data. A difference of only a few words may increase or decrease the measurements significantly. As seen in table 5.4, both approaches gave perfect scores for both precision and recall for adverbs, which should be due to the small number of adverb and the lack of function words in the set. The verbs show the same pattern as in the training set; the naïve approach tags all verbs regardless of a match in WordNet and the sophisticated approach has a lower recall due to the look-up. Both approaches have approximately the same recall score for nouns, but the sophisticated scores a higher precision. It also scores a significantly higher precision for adjectives even if the recall is not perfect. The words in this data set that have been tagged unknown

---

<table>
<thead>
<tr>
<th></th>
<th>Verb markers</th>
<th>No verb markers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Naïve</td>
<td>Sophisticated</td>
</tr>
<tr>
<td>Total</td>
<td>3453</td>
<td>3453</td>
</tr>
<tr>
<td>Nouns</td>
<td>2513</td>
<td>2309</td>
</tr>
<tr>
<td>Adjectives</td>
<td>169</td>
<td>342</td>
</tr>
<tr>
<td>Adverbs</td>
<td>32</td>
<td>134</td>
</tr>
<tr>
<td>Verbs</td>
<td>418</td>
<td>613</td>
</tr>
<tr>
<td>Unagged</td>
<td>321</td>
<td>330</td>
</tr>
</tbody>
</table>
are almost only nouns that are not found in WordNet. Many of these words that are about to be added to a already existing vocabulary are specific to certain domains. Many of them are not found and thus tagged unknown, which explains the low recall for nouns. The absence of functional words thus makes the precision score for the unknown words a little misleading.

Table 5.4: Test data

<table>
<thead>
<tr>
<th></th>
<th>Naïve Precision</th>
<th>Naïve Recall</th>
<th>Sophisticated Precision</th>
<th>Sophisticated Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nouns</td>
<td>0.97</td>
<td>0.74</td>
<td>0.99</td>
<td>0.73</td>
</tr>
<tr>
<td>Adjectives</td>
<td>1.0</td>
<td>0.5</td>
<td>0.76</td>
<td>0.81</td>
</tr>
<tr>
<td>Adverbs</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Verbs</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0.86</td>
</tr>
<tr>
<td>Untagged</td>
<td>0.01</td>
<td>1.0</td>
<td>0.01</td>
<td>1.0</td>
</tr>
</tbody>
</table>

5.4 Test Case

Table 5.5: A test case of the system’s ranking results

<table>
<thead>
<tr>
<th>File name</th>
<th>Word</th>
<th>Part of speech</th>
<th>Alternatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>anywhere,somewhere.png</td>
<td>somewhere</td>
<td>adverb</td>
<td>noun</td>
</tr>
<tr>
<td>bitter.png</td>
<td>bitter</td>
<td>adjective</td>
<td>noun, adverb</td>
</tr>
<tr>
<td>christian.png</td>
<td>christian</td>
<td>adjective</td>
<td>noun</td>
</tr>
<tr>
<td>cold.png</td>
<td>cold</td>
<td>adjective</td>
<td>noun</td>
</tr>
<tr>
<td>feeling.png</td>
<td>feeling</td>
<td>noun</td>
<td></td>
</tr>
<tr>
<td>flat.png</td>
<td>flat</td>
<td>adjective</td>
<td>noun, adverb</td>
</tr>
<tr>
<td>good.png</td>
<td>good</td>
<td>adjective</td>
<td>noun, adverb</td>
</tr>
<tr>
<td>green-(bci).png</td>
<td>green</td>
<td>adjective</td>
<td>noun</td>
</tr>
<tr>
<td>kite.png</td>
<td>kite</td>
<td>noun</td>
<td></td>
</tr>
<tr>
<td>Muslim.png</td>
<td>muslim</td>
<td>noun</td>
<td>adjective</td>
</tr>
<tr>
<td>north.png</td>
<td>north</td>
<td>noun</td>
<td>adjective, adverb</td>
</tr>
<tr>
<td>piano.png</td>
<td>piano</td>
<td>noun</td>
<td>adjective, adverb</td>
</tr>
<tr>
<td>poetry.png</td>
<td>poetry</td>
<td>noun</td>
<td></td>
</tr>
<tr>
<td>sharp-(taste),acid,sour.png</td>
<td>sharp</td>
<td>adjective</td>
<td>noun, adverb</td>
</tr>
<tr>
<td>sick.png</td>
<td>sick</td>
<td>adjective</td>
<td>noun</td>
</tr>
<tr>
<td>six.png</td>
<td>six</td>
<td>adjective</td>
<td>noun</td>
</tr>
<tr>
<td>square-(shape).png</td>
<td>square</td>
<td>noun</td>
<td>adverb</td>
</tr>
<tr>
<td>Swedish-(language).png</td>
<td>swedish</td>
<td>adjective</td>
<td>noun</td>
</tr>
</tbody>
</table>

The test case shows in table 5.5 how the system tags and ranks words. Several words can be found in more than one part of speech. For example, all digits (see ‘six’) and colours (see ‘green’) are both an adjective and a noun, as
are the nationalities (see 'Swedish') and religions (see 'Muslim'). The system ranks the adjective 'Swedish' higher than the noun. In cases like these, the word 'language' inside the file name is supposed to help the system choose the correct part of speech, which in this case is noun, over a possible more frequent one. Here, it fails as the WordNet gloss does not include the word 'language'.

Discussion

There were several factors that improved the application. The consideration of verb markers had the greatest impact. This separated the verbs from the rest of the words. If they were matched in WordNet, they immediately got tagged and did not have to compare frequency to another possible part of speech. File names with no verb markers did not need to be searched as verbs and fewer alternatives made it more probable to find the correct part of speech. The verbs were marked in the supplied material. The reason for keeping them was to get as good a result as possible. In another set of files, however, verbs might not be marked and this will presumably decrease the ratings, as table 5.3 indicates. It will also force the implementation to be somewhat edited to handle verbs in the same way as the other parts of speech. The current implementation could have lead to incorrect ranking, had there been verbs in the supplied set that by mistake were not marked. However, the verbs are not treated any differently in the search for a particular word or a letter. The regular expression is formed in a way that the search catches all words regardless of these kinds of markers. This means that verbs show up in all searches as well as the other parts of speech.

Another reason that the application got better results than originally, was the editing of the file names. The adjustment of these to WordNet decreased the number of words not found and made the system find more words in almost every part of speech. In some cases, the precision was improved within a part of speech, but the largest improvement was the increased recall.

That the algorithm was developed from its earliest stage to a solution that ranks the frequencies achieved a fair outcome instead of a haphazard one. This resulted in a higher recall for adjectives and adverbs, but also in a lower precision score. This is partly due to the fact that the frequencies are not always in line with the desired meaning for a particular symbol. However, it has more to do with that many symbols with the meaning of common prepositions were given the tags WordNet could provide, thus increasing the number of tagged adjectives and adverbs. Without further information and without possibility to compare with frequencies for part of speech that are not included in WordNet, the system has no ground for excluding those words. A possible solution would be to incorporate a stop word list with common prepositions. Even if the results for each set of files depends on the words in the set, the ranking provides more reliable results. This compensates the higher speed of the early algorithm.

The application’s subparts evaluated, a discussion of its total usability is important to achieve an integration of it as a tool for DART. The system is supposed to be of use for the developers of the symbol database that DART
maintains. It should facilitate the updating when each new symbol does not have to be looked up and written in manually. The use of the application itself requires only a basic knowledge of computers since there are step-by-step instructions (seen in Appendix B) to guide the user through the process. Two different types of users may here be identified. The maintainers of the database only need to handle the saved output and use it for updating their own data. The user of the interface, however, must have knowledge of the symbol system to correct the chosen definitions.

Even though the application is at first hand developed for DART, it will be considered open source and available for other users as well. It may hopefully be useful for maintenance of different ontologies and other kinds of databases where concepts and symbols are stored. The details of the implementation might have to be adjusted to different environments. The code could therefore be more wieldy, for example further divided into modules. This is something that could be achieved in the course of time. The application could be considered vulnerable since the file with the frequencies is saved locally. This might mean that possible updates of WordNet’s vocabulary will not be caught. However, this is partly an advantage since structural changes that may occur in WordNet will not affect the application. The interface is based on the Python GUI and if the current version is upgraded, the system might need maintenance regularly.

The application is in some aspect limited, given that it assumes a particular kind of input. The system adjusts the input based on the structure on the training material. If another set of files is used, these files must be of the same file type and the same structure to work. The size of the files and the symbol forms may vary as these matters are not specified in the implementation. A set of Pictogram symbols, PCS or plain photos may be used as long as the structure of the file names can be adjusted the way the application works. This is the main limitation since the structure of the file names is the key to the implementation and decides the rest of the application’s performance.

Time also set some limitations on implementing further features for the interface, such as a possibility to see an already edited page without losing information. This should be of value to a user that wants to go back and forth in a set of files. Further options for the presentation would also be important for the experience of the interface, for example a search more adjusted by the user.

An interesting issue is whether the data is representative. This can be viewed in two aspects. The structure of the file names is particular for this set. Even if some of the structures may be the intuitive way of representing words - for example a collocation with an underscore - the verb markers are not representative for any set. However, several word types are covered given that both synonyms, collocations, homonyms and the different parts of speech all are present. Though the structure may be unrepresentative, the 3340 files correspond to the double number of words used in an everyday vocabulary in English. The included words are mainly common words; out of these files, barely 10 percent are not found at all and given that the set also contains domain specific words, mainly centered around disabilities, this is not surprising. The data must be considered a quite good set to train the application on. Of course, one or several other sets would have been a fine addition to conduct
further tests and improve the application. As mentioned in section 5.3, the
test data that is used to validate the results has not an ideal distribution be-
tween the parts of speech. The words are to a large extent nouns and there are
almost no functional words. However, this application will be used to update
sets. Data that is added to an existing set is probably mostly nouns and prob-
ably within specific domains that WordNet might not be able to match. The
validation is therefore a good indicator of how an update may look like when
the application is used.

Something else to consider in terms of usability, is that even though this
application was developed on a Linux platform, a transformation to an exe-
cutable Windows file would be desired if it is possible. This could increase the
possibilities of a concrete use. Even if the supposed user has access to a Linux
operative system, the option to choose platform depending on for example lo-
ocation and not be limited to one computer is important. A Windows versions
will also make the application available for more users, both on DART and for
other possible users.

6.1 Conclusions

The purpose of this thesis was to describe the development of a system that
linked AAC symbols to their corresponding concepts. The aim of the system is
to facilitate maintenance of the ontology resources in a concept coding project
conducted by DART. The system was implemented in Python, with the help
of the python GTK library to construct a GUI for the application.

The application matches the file names of the input symbols to WordNets lexical database and presents the most frequent part of speech and definition
for each word to the user.

A gold standard with the parts of speech intended to each symbol was used
to measure the application’s performance. Three factors were important to get
a good result. The consideration of verb markers stood for a significant increase
in both precision and recall compared to when this information was not used.
The editing of the original file names to forms that the WordNet could accept
was also important. At last, the ranking the frequencies gave better results than
a naïve approach. The final version of the system showed that out of the 3340
files, barely 10 percent of those are not found at all. Most of those words are
unusual and domain specific and not in WordNet’s vocabulary.

From the results, conclusions may be drawn that a ranking method together
with the factors mentioned above improved the initial measures and the reliability. In most cases, the most frequent part of speech is corresponding to the
one intended for the symbol. A challenge would be to find additional algo-
rithms that can improve even further. It was harder to measure the method’s
efficiency in ranking the definitions since a gold standard for the definition
would be difficult to create given that many definitions could match a symbol.
In lack of any statistics that could indicate otherwise, the decision to provide
the most frequent definition seems natural. Both the user interface and the
algorithm could be more developed and improved by different means, but in
regard to the application’s present task, it succeeds in finding the most frequent
alternative. The application could improve its performance by optimizing the
algorithm, for example by dividing it into modules. Already now, it could be of help in maintaining databases. If it is improved further by for example the creation of direct connection to the databases, the possibility to change an edited word and more flexibility in the handling of different files could make it an even more useful tool.

6.2 Further Work

There are several things that can be done to improve the application further. To start with, it would be desirable that all kinds of files can be accepted. Accepting other extensions than .png, which were the files supplied here, is an easily made adjustment, either by specifying the extension in a way regardless of type or to let the user type in which extension is used. The latter suggestion will of course only be advisable for homogeneous set of files - if there are many different files in a set, this will not work. As for the structure of the files, an excellent addition to the present system would be an optional module, developed to look at the files, analyze them and suggest changes to the user, simultaneously explaining which structure the system requires. An edited version of this might be that the part of the application that adjusts the files is made an own module that easily can be modified depending on environment. This would probably lead to an optimized algorithm and increase the speed.

Something else that could be a valuable improvement would be to let the user see all or more of the definitions directly in the interface just by pressing a button, if the number of alternatives he or she has chosen does not prove satisfying. It would be interesting to evaluate the present choice of recommending 4 definitions per word. A user-centered evaluation could show if this number often is enough to find the right definition or if it should be changed. The interface should also be supplied with a previous button that lets the user see an already edited page and edit again. An additional feature to the search, both for total sets and for particular words or letters, could be options to see only synonyms, collocations or compounds or a specific part of speech to narrow the search.

It would be interesting to see if the frequency rating could be complemented with another method to increase the ratings, for example pattern recognition. To possibly improve the application further would be to get it to learn from the user’s corrections and thus produce more accurate results.

Finally, to further discover weaknesses and possible errors in the coding, different sets of words could be run and evaluated.
A Terminology

AAC  Augmentative and alternative communication, a term used for all kinds of speech substitutes.
BRO  Base Reference Ontology, a concept relation database based on WordNet, a part of CCF
CCF  Concept Coding Framework, a structure for handling concept based language
Concept coding The annotation of words with codes that define their meaning/concept
CRO  Complementary Reference Ontology, a concept relation database complementing BRO with function words
HTML HyperText Markup Language, a standard annotation and syntax for web pages
Ontology A formal representation of a set of concepts within a domain and their relationships
OWL  Web Ontology Language, a knowledge representation language and a technology used for the semantic web
PCS  A high-transparent symbol system consisting of drawings easy to understand, used for AAC
Pictogram A graphic symbol set in black and white used for AAC, where each symbol represents a concept
RDF  Resource Description Framework, a metadata model structuring information syntactically for computers
Semantic Web An extension of the World Wide Web, where the web content is defined in terms of meaning
SYMBERED A Nordic project that has developed an editing tool to create Web pages based on concept coding
Synonym set A group of terms that share a concept/have a similar meaning
WordNet A lexical database that groups English synonym sets and links them to each other
WWAAC The World-Wide AAC project, develops accessibility to the Web for people with communication impairment
XML  A specification for creating markup languages, to help information systems share structured data
## B Instructions

**Table B.1: Instructions**

<table>
<thead>
<tr>
<th>Point</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting the program</td>
<td>“Welcome to the concept coupler! Begin by choosing which image folder you want to open in the Options menu. If you have previously saved your work, you can open this file to continue where you finished.”</td>
</tr>
<tr>
<td>Folder open</td>
<td>“You have now opened [folder]/[file]. Next, enter the number of words you want to show per page (recommended size 100 or below) and number of definition alternatives per word (3-5 recommended). Press then the OK button.”</td>
</tr>
<tr>
<td>Symbol shown</td>
<td>“To change a part of speech or to see more of a definition, press its button. To see the next [number of] files, press the ‘Next’ button. Save your work or quit in the Options menu. You can find more information and help in the About menu.”</td>
</tr>
<tr>
<td>At last page</td>
<td>“No more files will be processed. Save your work or quit in the Options menu. You can find more information and help in the About menu.”</td>
</tr>
</tbody>
</table>

**Table B.2: Error handling**

<table>
<thead>
<tr>
<th>Point</th>
<th>Error handling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word search</td>
<td>The word [word] was not found in this set of symbols</td>
</tr>
<tr>
<td>Letter search</td>
<td>No words of the letter [letter] in this set of symbols</td>
</tr>
<tr>
<td>Too few boxes entered</td>
<td>You have to choose a word or a letter to search before pressing the search button</td>
</tr>
<tr>
<td>Too many boxes entered</td>
<td>Fill only out one of the search boxes - a word or a letter, not both</td>
</tr>
</tbody>
</table>