OBJECT PREDICTION AND SEMANTIC FEATURE LEARNING THROUGH NATURAL LANGUAGE INTERACTION

HONOURS DISSERTATION

by

Khaleeq-U-Zaman Ahmad

for the degree of BSc Computer Science

Supervisor: Prof. Oliver Lemon

Second reader: Dr. Lilia Georgieva

27 April 2015

Department of Computer Science

School of Mathematical and Computer Sciences

Heriot-Watt University

Declaration

I, Khaleeq-U-Zaman Ahmad confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed:
Date:

Abstract

Computers may be approaching some level of artificial intelligence, but they still have a long way to go in terms of appreciating or even understanding the sensory experiences and physical interactions that we can enjoy as human beings. There have recently been great strives in interactive technologies which emulate the human senses. Machines may not yet experience the world in the same way we do, but machines can now 'see' and machines can now 'speak'. But what do they think? How much do they really understand of the everyday 'things' around us? This project aims to develop an object prediction system which has semantic understanding of real world 'things', building on top of a set of 'feature norms' which describe different noun concepts. A system has been developed which is able to generate appropriate questions for guessing and attempts have been made for it to learn from its interactions of limited natural language dialogue in order to continuously expand its semantic knowledge of the world.

Acknowledgements

I'd like to thank my supervisor Oliver Lemon for his continued support and invaluable guidance. I always enjoyed our discussions, which helped to spark a real interest into the natural language topic. I'd also like to thank Verena Rieser for her initial advice and encouragement.

To my friends for putting up with me even in those last few days! We survived four years... just about!

And finally to my dad for always pushing me.

Contents

List of Tables

List of Figures

1. Introduction

This document has been prepared as the final submission for a fourth year honours project, and will first of all outline the aims and objectives of creating an object prediction system which learns from limited natural dialogue interaction. This document will review appropriate technical literature in the field of multi-modal detection systems, including both natural language and computer vision systems. The document will then address the actual design and implementation of the system in Python. The document will then go on to talk through a research evaluation which tests a basic and an advanced version of the system on human participants in order to assess how well the system learns, summarising its findings as I do so. The paper will finish with a discussion into limitations and potential future-work and improvements that could be made to the system, before rounding off with a conclusion.

1.0.1 Project aims

This project aims to design, build and evaluate an object learning system with some limited natural language understanding of question answer responses. Currently there are many systems for visual recognition of objects, however this project aims at learning solely through natural languages interactions.

A major component of the system will be a text-based dialogue system which will ask questions based on specific functional features [23] in order to make the most 'educated' guess at the given object. It will have some basic level of natural language understanding of user responses. Ideally this system should then be able to learn from the user's answers, which would help it make better predictions in future.

A potential end goal is to expand these semantic 'feature norms' into a more extensive living system which will eventually have some real 'understanding' of what the object semantically is. Such a system could then be used to train robots and machines about the real world.

There is some background reading on computer vision techniques for visual object detection as this was an early goal of the project. Silberer and Lapata's project [34] takes an interesting approach of combining two modalities, visual and textual. It utilised semantic textual data from Wikipedia and trained images on nouns from McRae et al.'s set of 'feature norms' [23]. Their research found that their bimodal model was more effective than using just one modality. A multi-modal system such as this is beyond the current scope of this project but it is an interesting challenge to consider in future. Combining visual image recognition on top of a working system with semantic object prediction would be quite a novel approach.

To truly understand what an object is on a semantic level we must not look just at its properties and functions, but also to its relationship to other objects and the context of its current use. We make concrete steps towards ideas such as the 'semantic world' and the 'web of things'[13, 10], however they are beyond the scope and limited time-frame of the project.

1.0.2 Objectives

In order to accomplish the aims set out above, a number of objectives have been identified. These are are outlined below.

To offer the best possible prediction of a noun concept from a given set of features
This objective requires the system to be able to make hypotheses of the best candidate concepts most fitting the given features.

To interpret and evaluate some dialogue through language understanding
This objective requires the system to be able to interpret the users responses through an interactive dialogue system and to have some understanding of how to process this meaningfully.

To test predictions by asking the user questions
This objective requires the creation of a dialogue system for interaction with the user. This dialogue system needs to select the best question to evaluate its predictions.

To learn from the user to improve future accuracy
This objective requires the system to learn from its interactions with the user. It should be able to correct incorrect assumptions in order to improve the accuracy of future predictions.

2. Background

A number of different papers were researched as part of a literature review. This was in order to firstly, gain a general understanding of the background of important fields of artificial intelligence such as natural language processing and computer vision; but also to gain knowledge of technical concepts necessary to reach our project objectives, as outlined earlier in this document.

2.1 Feature Norms

McRae et al. [23] have collected and produced a large dataset of semantic 'feature norms' for 512 noun concepts – consisting of both living and non-living things. They were collected over a number of years by asking participants to name feature descriptors for lists of different noun concepts.

As the dataset includes a largely mapped set of semantic features with nouns, it would serve as an ideal basis for generating object-specific questions. However, it does not come without its own drawbacks and limitations.

McRae et al. note that these features are far from being a complete representation due to the complexities of communicating some complex ideas. Features which are difficult to verbalise or those relating to more specialised knowledge were harder to gather. For this reason many features common to a variety of objects may also be missing, as distinguishing features are noted first. This also leads the feature norms to have a positive bias, the dataset is concerned with recording features known to apply to a certain concept (eg *can fly*, but does this imply that all concepts missing the feature cannot fly? Or is it meant to imply an ambiguity, an answer is not known either way? Or, thirdly, simply an oversight – the information just happens to be missing. Complicating matters is the fact that the dataset actually contains the complementary feature *can fly*, but this approach is not consist for every function. For these reasons it would be an interesting approach for us to expand upon the feature norms to create an 'evolving' datatset which is in constant learning to improve its knowledge.

2.2 Natural Language Processing

Natural language processing (NLP) is an area of artificial intelligence, with links to computer science, cognitive science and linguistics. It forms part of human-computer interaction (HCI), through looking at how natural language affects and interacts with computers. Its aim is to emulate the way in which humans communicate through natural languages such as English.

Communication is a two-way street. Naturally as human beings we initiate, continue and conclude conversations through both talking and listening; admittedly some are more prone to one than the other; however both acts are equally important. Mapped to a computer model, we could see these two senses simply serving as 'input' and 'output' just as a computer system might have a microphone and speaker. Beyond this however, is the human intelligence we use to make sense of meaning and in order to convey our own. It is thus wrong to think of speaking and listening –and indeed reading and writing– as simply external sensory actions, they are in-fact far more complex and nuanced activities requiring due brain processing.

The two main areas of NLP involve interpreting unambiguous meaning and context from language (Natural Language Understanding) and the creation of natural sounding responses (Natural Language Generation). This project will be looking somewhat into both areas. The former to understand how humans answer questions using typical natural word responses, as well as considering the latter in making natural sounding questions out of 'feature' names. These, however, are by no means the only important areas of NLP, which is a wide-ranging discipline with many research goals and applications.

2.2.1 History and Background

Although seventeenth century philosophers René Descartes and Gottfried Leibniz discussed metaphysical concepts, which would later relate to the NLP topic of machine translation, these were purely theoretical. In 1950, it was the ideas proposed by Alan Turing, that would later come to be known as his famous 'Turing test', that laid the foundations for Artificial Intelligence, including NLP. Turing asked whether machines could think or at least imitate humans[37]. ELIZA was an example of an early chat-bot which aimed to pass the test. It used Rogerian psychotherapy to repeat the user's answers back to them. It should be obvious to the reader that such an approach would, of course, not pass the Turing test, although many further attempts have been made over the years.

In 1954 the Georgetown experiment was conducted, it was able to automatically translate over sixty sentences from Russian to English. However it took until the 1980s for there to be much in the progress in machine translation.

The late 1980s introduced machine learning algorithms, up until this point NLP systems used, sometimes complex and long, lists of if-statements and rules. Decision trees like this are still used in modern day NLP. However, machine learning has introduced probabilistic decision making by ranking outcomes with weighted values and using this information to predict the best of many different outcomes. From this, we see that probabilistic decision making is much closer to human reasoning and the 'thinking' that Turing intended in his thesis. As human beings, we have the capacity to analyse, judge and predict different courses of action. Our thinking is a lot more complex than the 'rote learning' of a purely mechanical flowchart or long list of instructions, as each thought or idea has a different weight, whether emotional, logical or based on previous experiences. Some form of this kind of probabilistic approach would be necessary for a system order to have a system which truly understands the semantic properties of an object.

2.3 Dialogue Systems

A 'dialogue system' is type of human-computer-interaction using natural language techniques to have a conversation with a user. Rieser and Lemon describe such a system as a computer agent which "interacts with humans by understanding and producing language in a coherent way"[30]. They are often used for information retrieval and for completing tasks such as shopping. Dialogue systems are now being used in smartphone applications such as Siri for iOS.

.

2.4 Object Recognition and Definition

Understanding how humans use the semantics of language to identify objects is the true aim of this study, however in order to identify objects we must first see how they are defined. Grounding computer systems in our own real world examples allows us to shape systems in a real world context, thereby allowing these systems to 'step into our shoes'[31]. As children we all learn the words for objects in our native language, by looking at how we make these first connections we can begin to understand the meaning behind lexical definitions.

2.4.1 Language Acquisition

Developmental psychology has a great deal of literature on how children develop their early linguistic skills and learn their first language. Children learn to recognise patterns very early on, before they can understand the structures of linguistic communication [32]. Understanding these patterns, children specifically look at the world with a 'shape bias', rather than other patterns such as colour, size or texture, in order to group similarly shaped objects together[20].

However, there is also research to support the 'function bias'. Upon learning a name, a two-year-old child can understand the object's function and apply the same name to other objects with similar functions[17]. Other research shows that children may also have the capacity for 'basic categorisation by semantic meaning'[11].

2.4.2 Shape vs. Functional Bias

Eguchi's approach takes these ideas from child developmental psychology [9]. His research argues that this 'functional bias' is as important as an object's shape and colour. An object recognition model with a purely shape bias would struggle to associate functionally similar objects which happen to have different shapes, for example two different brands of car, or two different types of chair[15]. How would a shape-based object classifier distinguish these?

Another good example of this is if we look at different breeds of dog. One shape does not fit all, dogs come in many different shapes and sizes, and therefore a 'shape bias' does not provide a general identifier fit for the whole group. We could of course think of the 'prototypical' dog and derive identifiers from a so-called 'standard shape', however this would not be true to all cases in a real-world environment, and such 'prototypes' would need at least some level of grounding to be useful between cultures or even individuals.

These examples tell us that there is more to an object than just its shape. We define a chair as somewhere to sit, and a car as something we drive. Obviously these functional categories hold more weight than a generalised shape, as they tell us what we can actually use the object for and influence how we interact with the said object. Of course we can generalise even further, a car is a kind of vehicle and a chair a piece of furniture.

We can look at this hierarchy in a similar way to class inheritance in Object-oriented programming. Each object has different properties (its shape, colour and size), but also a number of functions. Some of these functions are inherited, so for example vehicles such as cars inherit a drive function. This does open the issue of how we differentiate between different levels of generalisation, can our system differentiate between a vehicle and a car? To solve this we must use some sort of ontological model.

If we were to look at a purely functional bias, there may be confusion between say a car and van which are functionally similar, but have some differences. In the same vein, what if we test between two objects that look similar but are functionally very different? In his study, when using a purely shape-biased system Eguchi found it difficult to differentiate between an antiperspirant and an insecticide, both came in aerosol cans, however had very different uses[9]. Although there may be a level of lexical ambiguity again here, as technically both could be defined as aerosol cans, for the purposes of his experiment Eguchi treated them as separate objects. These findings show that a combination of both shape-bias and function-bias lead to the best results.

Diesendruck and Bloom argue that children believe similar looking objects belong to the same categories based because of their own perceptions of a shape bias, rather than from learning any deeply linked association in their minds. [8].

2.4.3 Grounded Systems and Ambiguity

Words can at times be very ambiguous, for this reason context is very important. For example, let us look at the example of dog breeds again. There are a number of properties which all dogs share –such as a long snout, the ability to bark and a strong sense of smell– a number of such attributes together help us to identify the entity we know as 'dog'. Bi-

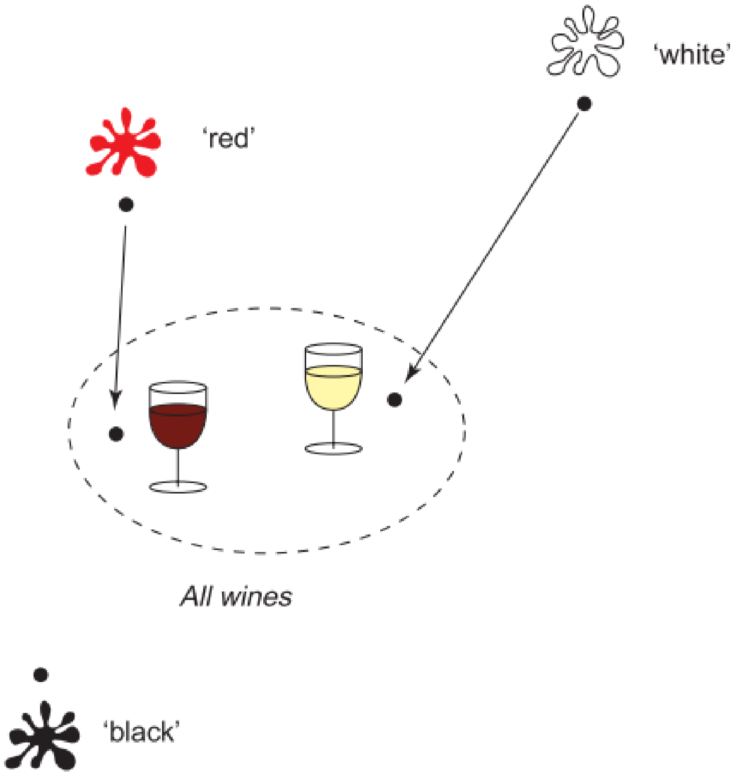


Figure 2.1: Despite 'red wine' being quite a different hue from the 'red' prototype, this model show how such an arbitrary language convention could have born[31].

ologists use an extensive taxonomic classification system to define living creatures under different ranks, defining dogs as members of the family Canidae[12]. However, there is ambiguity even here. When we say the word 'dog', do we include wild species such as wolves, foxes and jackals? Do we include other disparate species with 'dog-like' characteristics such as hyenas? Or are we more likely referring to particular domesticated breeds, or even have a specific breed in mind? Perhaps you have the idea of the 'prototypical' dog in your mind, but as discussed earlier what does that actually mean and is it really relevant to real world situations?

Word meaning can change and have different meanings in different contexts. As specific meanings are not always fixed, we cannot tightly map our word representations as in 'purely symbolic' models [31]. When creating language models of the real world, we must look at 'grounded systems' which can 'step into our shoes'[31].

Colour naming models, such as Mojsilovć's (2005), are one example of how words can be grounded by perceptual categories. Mojsilović's model [25] associates colour names with prototype colours, with the assumption going that these values are fixed. However, in real life situations we can use colours in a variety of ways not easily caught in such a model. For example 'red hair', 'red wine' and 'red sportscar' can all refer to completely different hues which may be closer to orange or purple in other contexts.

An alternative, more context-sensitive colour model is Gärdenfors' model [31, 14]. This model more closely shows placement of different colours, as shown in the example of red and white wine in Figure 2.1. Arbitrary language conventions can lead to differences such as the darker 'red wine' being known as 'black wine' in Catalan. However, this model shows that it would be impossible to swap the red and white names due to their relative distances from dark and light shades.

Silberer and Lapata (2014) present a multi-modal model aiming to learn 'grounded meaning representations' [34] using stacked auto-encoders. An auto-encoder (also known as a Diablo network) is a neural network designed to learn codings.

Silberer and Lapata's model [34] combines both the visual and textual modalities, encoding each as vectors of natural language attributes generated automatically using text and visual data. They extracted attribute pairs using the program Strudel [2] and a dump from the English Wikipedia. Meaning representations came from the McRae feature norms [23] , a set of 541 animate and inanimate objects, each with a list of 'feature' properties. Images were taken from ImageNet [7] and trained on nouns from the McRae feature norms. Their research found that their bimodal model had improved accuracy over unimodal alternatives.

Kollar et al. [18] developed a system for 'grounded language acquisition', that would learn from natural language expressions when referring to the real world.

2.5 Computer Vision and Image Processing

Computer vision (CV) is a field in artificial intelligence for using systems to acquire, process and understand visual inputs, modelled after the human eye.

Understanding CV will ground us in having a greater understanding of varying multi-modal approaches. If we were to consider a digital image file, most semantic information contained in an image is meta-information about the file itself rather than the concepts and objects it represents. Computer vision techniques go some way in addressing how you go from this starting image file right through to its final identification, through various stages of processing and extraction of features such as lines, shapes and colours.

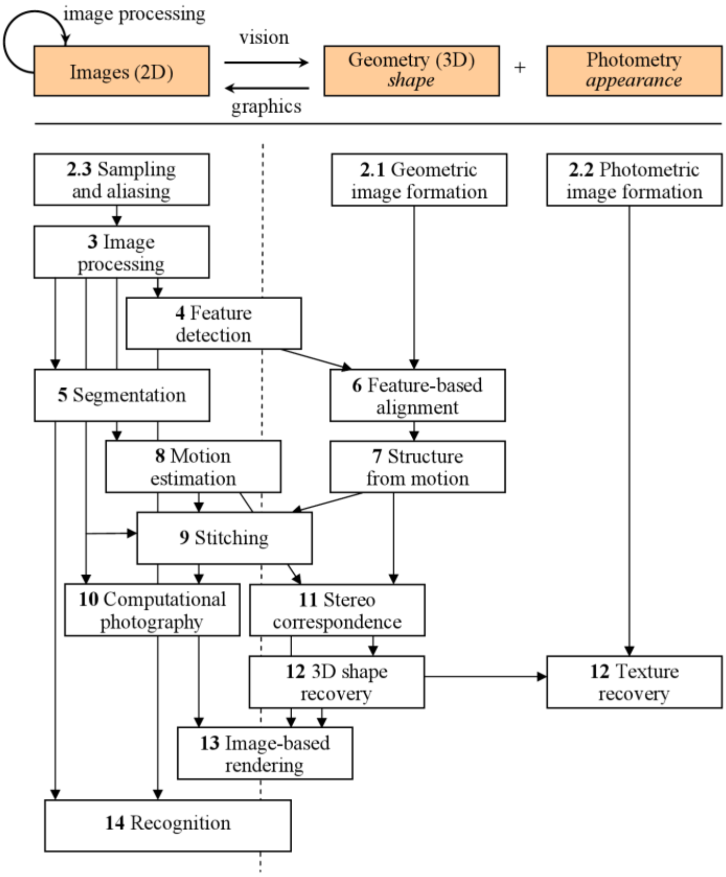


Figure 2.2: Different aspects of Computer Vision, relating to images, geometry and photometry [35].

Information can be extracted from a variety of sources such as multiple camera angles of the same subject, video recorded motion and output from 3-dimensional medical scanners. After it is extracted, an image may be 'pre-processed' for consistency, with specific features – such as lines, edges or particular points of interest– being detected for extraction and segmented for further processing, through methods such as Bag of Visual Words (BoVW)[19]. This data may then be further processed through image recognition, detection or identification algorithms.

2.5.1 Applications for Computer Vision

The applications for computer vision are far-ranging and varied. These include optical character recognition (OCR) and also fingerprint and facial recognition for bio-metrics. In these cases images are used to identify a certain source, whether that be a single character, a car numberplate or an individual's identity.

Image detection is where we look at the input image for certain criteria. Diagnostic imagery from ultrasounds, x-rays and tomography machines can be processed to detect for physiological malignities such as tumours[1]. Medical researchers also use these tools to continually learn about organic structures such as neural pathways in the brain.

Military applications for CV include missile guidance and enemy surveillance. Surveillance is not just limited to the battlefield, crowd counting and event detection software is also used in conjunction with security cameras[33]. Computer vision is also being increasingly used as input for human-computer interaction as opposed to or in addition to natural language in so-called 'multi-modal' systems.

'Machine vision' is a related field which crosses over with CV, it is of particular use in industrial robotics, where it is used to develop navigational and control systems for autonomous robots[16]. In computer-aided manufacturing, these autonomous robots can also be used for tasks such automatic quality checking.

2.5.2 Visual Bag of Words

The Visual Bag of Words (VBoW) model is a novel method for image classification based on similar Bag of Words (BoW) techniques already used for text classification in documents. BoW is a "histogram representation based on independent features" [19]; essentially it is based around the general idea of removing elements from their context, in order to create a vector with the number of times each element re-occurs. For document analysis this is represented as a multiset or "bag" containing multiple instances of words. As the most common words tend to be the most important, both ordering and grammar can be ignored however multiplicity must be kept.

This Bag of Words Model is now being increasingly used as a method to solve problems in computer vision, by treating images like documents. As per the text-based model, the goal is to represent each image with a vector of iconic features known as 'visual words', ignoring its position or context. These 'words' still need to be found through a feature detection algorithm.

Following detection is the process of feature representation. A feature descriptor, such as SIFT (Scale-invariant feature transform) is used for dimensionality reduction, as there is a large amount of unused data. SIFT can extract the most important and relevant

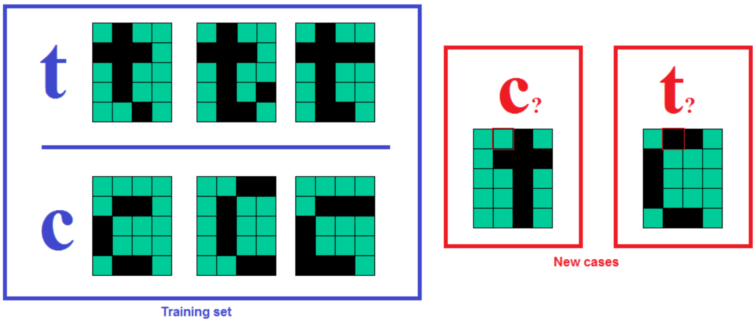


Figure 2.3: This is an example of bad generalisation from a training set causing an overfitting error. In this case the system has erroneously 'learnt' to distinguish between '***t***' and '***c***' based on only the second pixel. *Example adapted from* [5]

interest points into a feature vector [38], dealing with variations in scale, rotation and intensity.

The third important step is to generate a visual codebook (or dictionary). Codebooks are useful for grouping visual words together. Clustering methods,such as k-means, are popularly used for this purpose. K-means clustering [36, 22] is an iterative process, from a given cluster it computes the centre as the mean distance between all members. Each element is then reassigned to the cluster centre it is nearest to. This process repeats until no members need to be changed. These clusters very often refer to different objects, or parts of objects. Each cluster then forms a codeword. As each image feature is matched to a codeword, this ultimately allows creation of histograms.

Although the BoVW method is very successful in classifying images by the objects they contain, being unaffected by an object's position or orientation in doing so, it is a relatively new technique which requires further testing for changes in scale and camera view. It also has no way of dealing with positioning, as this information has been removed. As images often represent a three-dimensional space, the position of objects in relation to each other is arguably more important for an image than word order may be in text.

2.5.3 Classification Methods

Of course, despite increasing research in this field, a computer may never be able to 'see' and 'process' visual stimuli as well as the eye and brain of a human child. For example, a computer will have greater difficulty finding corresponding points between two images when when there are differences in illumination or position, a task which the human brain should find trivial[24]. Classification errors such as these can result due to interference from randomness and 'noise', leading to the system making poor generalisations. This is known as 'overfitting'[5].

Overfitting, as shown in Figure 2.3, can be minimised by using a greater set of training data as this should train for a more diverse set of results and potentially cut down on any erroneous stereotyping. Jittering is another way to overcome overfitting, by introducing random noise into the dataset.

Support vector machines (SVMs) are a type of non-probabilistic supervised learning model used for classification and regression analysis. We can plot a number of points on an SVM, in order to best distinguish between two categories, the points should be divided by a large boundary. New points are then added to represent the new unclassified input. Its category is then predicted based on its fall on either side of the gap. [5, 4]

3. Software Design and Implementation

This section will outline goals of the software application. It will discuss the application's requirements, how it was designed and go on to look more in-depth into its actual implementation.

3.1 Requirements

3.1.1 Functional Requirements

1. The system will be linked to a relational database to store and retrieve knowledge.
2. The system will be able to generate an initial question to split the dataset.
3. The system will include a dialogue system which the user can interact with.
4. The system will be able to select the best of a number of questions to ask the user.
5. The system will be able to make further decisions based the user's answers.
6. The system will decide from amongst a group of candidate objects.
7. The system will be able to improve its answers based on what it has learnt from the user.
8. The system will ask the user about discrepancies in a measure to prevent learning mistaken information.

3.2 Prerequisites

3.2.1 Software Environment

The system was developed in Python [1] using the Natural Language Toolkit in order to find words and definitions on the WordNet Corpus[2].

[1] Available at: http://www.python.org

[2] Available at: https://wordnet.princeton.edu

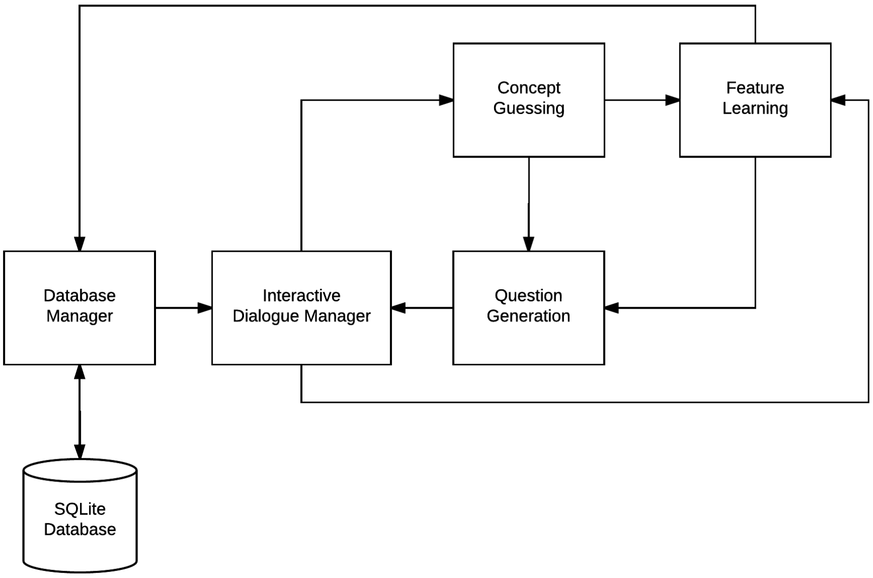


Figure 3.1: Design of the system's main functional components.

3.3 Functional Design

This section overviews the high-level design of the system.

The main components of the system are data storage and retrieval, question generation, concept guessing and feature learning as shown in Figure 3.1. The interaction manager sends and receives database information. It passes this on to facilitate guessing and learning with the user's input. Both guessing and learning rely on question generation, which is sent through the interaction manager. Further elaboration on the design of each component follows.

3.3.1 Database Representation of Concepts and Features

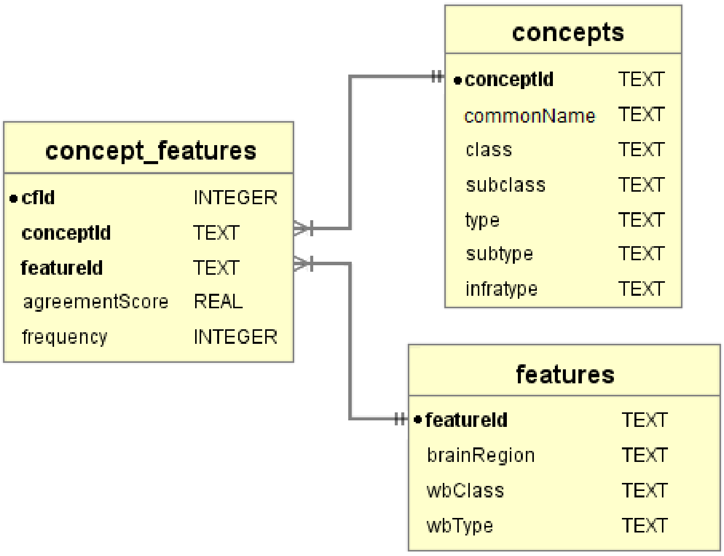


Figure 3.2: Entity-Relationship (ER) diagram for the concept/feature database.

The system's knowledge, which is adapted from the McRae feature norms [23] is stored in a relational database. As shown in Figure 3.2, the database has separate tables for both `concepts` and `features` with a third table `concept_features` joining these together.

3.3.1.1 Concepts

The system contains information on 512 unique noun concepts, and each concept has a unique ID tied to a WordNet synset [29]. This is less than the original 541 from McRae's dataset [23], as some concepts (such as *cottage* and *bungalow*) had to be manually combined to be represented under a single synset.

I also created a rudimentary taxonomic system to classify each concept under a hierarchy of categories, as shown in Table 3.1. Categories are organised in a hypernym-hyponym relationship, where each child category has an 'is-a' relationship to its parent category. For example, the concept *dog* was categorised under *natural*→*animal*→*vertebrate*→*mammal*→*canine*. The taxonomy was adapted somewhat from a combination of hypernyms listed on WordNet [29] and the Suggested Upper Merged Ontology (SUMO) [28, 27], which itself has concepts mapped to WordNet. Ultimately the first level, 'class' (*artificial* and *natural*) is deep enough for the finished system, however the skeleton for a more diversified taxonomy is still there for future expansion.

3.3.1.2 Features

The features again originate from the McRae dataset [23], as over 2500 different traits associated with each noun concept. Ranging from the general to the specific, these include aspects such as functional properties, membership of a group or possession of a certain physical characteristic. Features were renamed to assist with later question generation, this was mainly based through search and replacement of the existing phraseology, as the original feature names were themselves heavily standardised. The Wu and Barsalou [21] and Brain Region [6] classifications were also considered when deciding on alternative names. They have since been kept for organisational purposes, however are not currently used by the system.

3.3.1.3 Concept-Feature Relationship

As every concept is linked to a number of different features, which will only continue to grow as learning takes place, it is important to consider the best organisation for this. It will be necessary for the system to keep a measure of 'relatedness' for each feature. For the existing relationships, the production frequency will be adapted into an `agreementScore`, with an additional `frequency` field to keep track of the number of times a question about this concept's feature has been answered (defaulting to 30 for pre-existing features, as this is the number of people questioned by McRae et al). The *dialogue manager* will then be able to calculate this 'relatedness' score by division of the `agreementScore` by the `frequency`. The McRae feature norms [23] defaults to having a positive bias as it only contains features which participants considered to be present in each noun concept, this system will take this into account by accumulating negative values to the `agreementScore` for 'No' answers. Scores approaching zero will be considered undecided, or otherwise unknown if not present in the table at all. Border-line scores which are not yet high enough to be 'yes' or 'no' may also be considered 'likely' or 'unlikely' instead.

3.3.2 Database Manager

The system will need a database management module in order to connect the code with the external database. This database manager would be responsible for retrieving new information from the database as well as updating it to reflect new changes to the system when adding or updating new or existing concepts or features.

3.3.3 Interaction Manager

The interaction manager contains the main control flow of the application as it is where the user interaction takes place. It will load data from the database manager which it passes through its flow. The interaction manager will control both dialogue from question generation and user input, as well as managing how the user interacts with concept guessing and feature learning. Both learning and guessing subsystems have the aim of 'understanding' the user's input.

3.3.4 Question Generator

The system requires a question generation module to produces output for the dialogue manager which is required for both the concept guessing and feature learning processes. A question must first be selected, this is done by the Concept Guessing module which will aim to split the top candidates in the middle following a traditional divide and conquer algorithm, after finding this middle value it selects a question which will apply to the closest number of candidates. Following the current design, the system will use a simple control statement to decide how to rewrite a feature name into a question, as defined by a number of standard formats in the database. See Table 3.2 for the list of rewrite rules.

3.3.5 Concept Guessing

The concept guessing module looks for the best candidates matching the current feature-set. It splits down the number of possibilities in order to determine the best candidates for the question generator to ask about. And from interpreting a question it has functions to alter the scoring for the remaining candidates. The system will eventually decide on the best guess before continuing on with the feature learning.

3.3.6 Feature Learning

The feature learner module will evaluate the user's question answers, to append new agreement scores for features. It understands a range of different natural language answers, which can be converted to number ranges. Upon learning about new concepts it looks up word synsets on the WordNet corpus. A synset is WordNet's name for a grouping of synonyms (or 'lemmas') tied to the same lexical meaning. It will also provide discrepancy checking, comparing major differences between the system's existing knowledge and the user's answers as reassurance before updating the database. Eventually, feature learning may have more advanced language understanding wherein it could then interpret the user's dialogue in learning about open-ended features.

3.4 Implementation

3.4.1 Database and Database Manager

The database was initially implemented in MySQL [3] running over an Apache server [4] , however I found this set-up to be far too slow for loading. Instead I found SQLite[5] to be a somewhat faster alternative. The database is implemented as three tables as shown in Figure 3.2.

Two copies of the database are stored under `objpred/data`, a basic static version which used for the basic non-learning system, and another which has been updated with feature learning.

The `DbManager` class (`objpred/main/nl/db/DbManager.py`) connects to SQLite in order to setup and fetch concepts and features which will be passed on in dictionaries to the main dialogue managers. When these dictionaries have been updated upon feature learning the `DbManager` calls a function, to update `concept_features` with the new information. When the learnt object is a brand new concept, it runs a function to insert it into the `concepts` table.

[3] Available at: http://www.mysql.com

[4] Available at: http://httpd.apache.org

[5] Available at: http://www.sqlite.org

3.4.2 Basic vs. Learnable Versions of the System

The main control and dialogue management for both systems have been written in separate files, although they are functionally similar and share most of the same functions from the remainder of the system. Both systems first of all call a Database Manager to prepare dictionaries containing all features and concepts. Both then go on to set up an ordered dictionary containing guess scores for each concept, however there is a major difference in how the two systems will process this dictionary due to the different types of answers each system can process.

Fundamentally both systems will go through a process of breaking down a set of answers to find the best candidates, using the question generator to find a question that splits this in the middle. The specific number is found by taking half of the difference between the maximum and minimum and then finding the closest absolute number. Initially this was attempted with a median score, however a median score was found to skew away from the absolute middle after a lot of mid-range numbers were missing.

The fundamental differences between the two systems are:

3.4.2.1 More forgiving priority system

The Learnable system will use a priority queuing system which is more forgiving, instead of objects being removed almost immediately after getting a ”no” response everything is kept. Instead the system will look at the top 25th quartile of scored objects and deem them the best candidates from the entire set. When the system next asks for a question it will adjust each score and again look at the 25th quartile until it reaches a termination point.

3.4.2.2 Natural Language Responses

The Learnable system will be able to understand a limited number of natural language responses and understand their equivalence on a 5-point scale between -1 and 1. Each word is interpreted as a score and scores from the database can be retrieved back. See Table 3.2 for of the currently understood responses in each category.

Answer name	Score	Possible answers
No	-1	”definitely not”,”false”,”negative”,”never”,”no,”no”,”nope”
Unlikely	-0.5	”doubt,”doubtful”,”hardly” ”maybe not”,”i don’t think so”, ”negative”,”not confident””not likely”,”not really”, ”unconfident”,”occasionally”,”perhaps not”,”presumably not”, ”probably not”,”rarely”,”seemingly not”,”seemingly not”, ”seldom”,”unlikely”
Neutral	0	”can’t tell,”cannot tell”,”do not know”,”does not apply”, ”doesn’t apply”,”don’t know”,”hard to tell”,”hard to tell”, ”indefinite”,”irrelevant”,”n/a”,”no idea”,”not”, applicable”, ”skip”,”uncertain”,”undecided”,”undetermined”,”unknown”, ”unsure”
Likely	0.5	”believably,”confident”,”i think so”,”likely”,”maybe”, ”partially”,”perhaps”,”plausibly”,”positive”,”presumably”, ”probable”,”probably”,”seemingly”,”seemingly”,”sometimes”, ”somewhat”
Yes	1	”affirmative,”always”,”definitely”,”positive”,”true”,”yeah”, ”yep”,”yes”

Table 3.3: Listing of natural language mappings the system can understand.

3.4.2.3 Learning

The Learnable system adds to the database. It will input new features inorder to learn both through guessing and the through the `FeatureLearning` module. The `FeatureLearning` module asks the user further questions to compare the user’s object concept to the closest runner up when correctly guessed, or otherwise with that incorrectly guessed concept. The aim is to learn more specifically how to differentiate between those two objects in order to avoid repeating the same mistake. The system will also check the user’s answers for any which are the opposite polarity to what is in its database (eg ”yes” or ”likely” for something it thinks is ”unlikely”, as a method of consistency checking).

The Learnable system can also learn about entirely new concepts, this is done through use of the Natural Language Toolkit (NLTK)[3] [6], to query the WordNet lexical corpus for different definitions. WordNet is organised into groups (known as synsets) of different synonyms (or lemmas), so the system can still find a concept if the name they give is a different lemma from the the synset's head name.

[6] Available at: http://www.nltk.org/

class	subclass	type	subtype	infratype
artificial	clothing	footwear		
		headwear		
		jewelry		
		nightwear		
		undergarment		
	construction	artwork		
		building	home	
			place_of_worship	
			room	
		material		
		monument		
	device	machine	kitchen_appliance	
		musical_instrument		
		text		
		toy		
		vehicle	aircraft	
			land_vehicle	
			watercraft	
			wheeled_vehicle	
		weapon		
	food			
	furniture	plumbing_fixture		
	implement	container		
		covering		
		sporting_equipment		
		tool	kitchen_utensils	
		toy		
		weapon		
natural	animal	invertebrate	arthropod	arachnid
				crustacean
				insect
			mollusc	
		vertebrate	amphibian	
			bird	
			fish	
			mammal	canine
				feline
				marine_mammal
				primate
				rodent
				ungulate
			reptile	
	mineral			
	plant	plant_produce	fruit	
			nut/seed	
			vegetable	
		vascular_plant	tree	

Table 3.1: Shows the constructed taxonomic structure which concepts were classified by.

Prefix	Rule
”it ”	Replace ”it” with ”This object ”
”is ”	Replace ”is” with ”Is it ”
”has ”	Replace ”has” with ”Does it have ”
”was ”	Replace ”was” with ”Was it ”
”it’s ”	Replace ”it’s” with ”Is it ”
”for example ”	Replace ”for example” with ”Is ” and append ”an example or version of this object”
(all)	Append ”?”

Table 3.2: Table of question generation rewrite rules. Each ’feature’ name

4. Evaluation and Analysis

A research study was carried out to evaluate the two systems, in order to decide whether improvements to the Learnable system yielded more successful results. This section will outline the aims of the investigation as well as how the experimental procedure was carried out before analysing the findings.

4.1 Research Hypotheses

The study will be testing the following two hypotheses over the two systems.

Hypotheses 1

System B (the Learnable system) is more successful at guessing concepts than the basic system (A).

Hypotheses 2

System B (the Learnable system) will become more successful at guessing concepts over time.

4.2 Testing Procedure and Design

The study involved testing a sample of ten participants who were each given a list of ten word concepts to test on each of the two systems (see Appendix A for the full task scenario). In order to directly compare between both systems, the experiment makes use of within-subject design, meaning each participant was required to test both systems. To minimise order bias the orders of systems tested was randomised using matched pairs. The word orders were also randomised, with each subject given a random ordering for each system they were testing.

Participants were asked to answer with either yes or no responses to System A, meanwhile in order to best account for the natural language features of System B they were asked to respond openly. A list of possibilities were provided as a guide, however they were strongly encouraged to answer with the most natural sounding response. Unrecognised

wordlist
*arrow
bullet
butterfly
elephant
grape
jet
*parrot
penguin
sword
tomato

Table 4.1: The list of concepts provided to participants, *arrow* and *parrot* are notable for being new concepts, not already found in the database.

commands would of course not pass through input validation, the system would allow the them the opportunity try those again.

Some special considerations had to be taken into account when testing Hypothesis 2. To ensure that each every occurrence of a word concept can be tested against itself fairly and impartially, the database update procedure for system B only takes place on launching the system for a new participant. On loading from the refreshed database it will retrieve all the database changes from the last participant's round.

Each system automatically logged to its own CSV file, a number of values from running the tests such as the question count, the user's answers and the system's previous guesses. [1].

Additionally, each task was followed with a questionnaire to assess user preferences and opinions of each system. Participants were asked evaluation questions on a four-point Likert scale. Four points were chosen to force the user to make a judgement. These were followed by some short answer qualitative questions.

Finally, participants were asked to complete a final post-questionnaire which asked demographic questions on age, gender and whether the participant was a native English speaker.

4.2.1 Participants

The study took a sample of ten participants, who were all students at Heriot-Watt University. All participants were aged between 18 and 25, and were native English speakers.

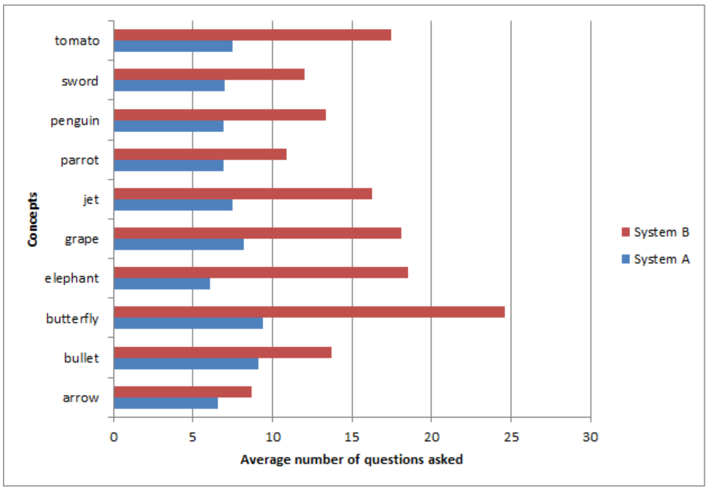
[1] Can be found within the online code listing

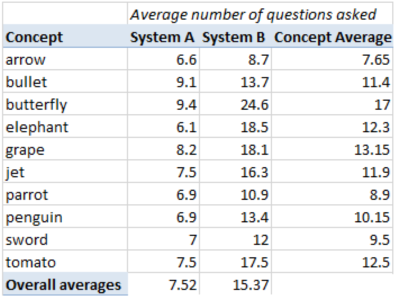
Among the sample, six were male and four were female.

4.3 Analysis and Findings

This section will analyse our results.

4.3.1 Number of Questions Asked





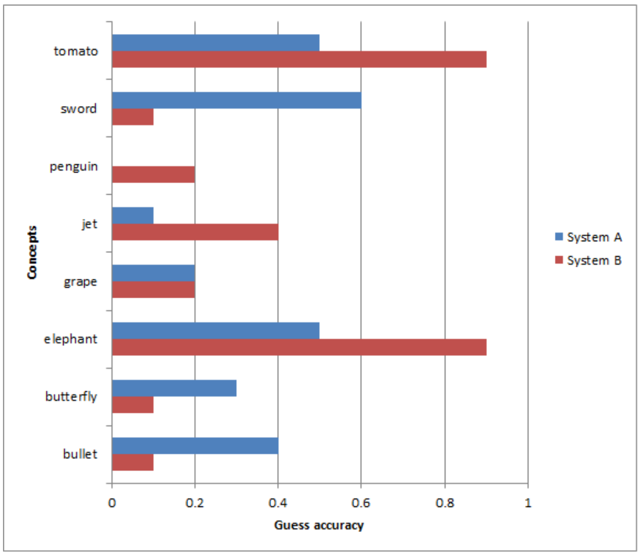
	Average number of questions asked		
Concept	**System A**	**System B**	**Concept Average**
arrow	6.6	8.7	7.65
bullet	9.1	13.7	11.4
butterfly	9.4	24.6	17
elephant	6.1	18.5	12.3
grape	8.2	18.1	13.15
jet	7.5	16.3	11.9
parrot	6.9	10.9	8.9
penguin	6.9	13.4	10.15
sword	7	12	9.5
tomato	7.5	17.5	12.5
Overall averages	7.52	15.37	

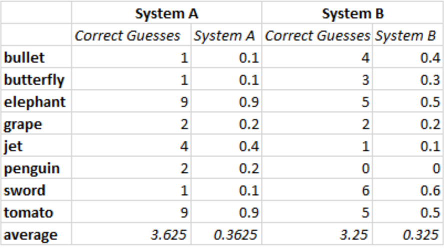
Figure 4.1: Average number of questions asked on both systems

As was expected system A outperforms system B universally. From these findings it looks like System A (averaged at 7.52 questions) is in-fact twice as fast as System B (15.37). Interestingly new concept *arrow* is one of System B's fastest. This might suggest that a higher number of features in the database lead to a longer guess.

4.3.2 Guess Accuracy

System B may slightly edge out System A in terms of which did not predict penguin correctly at all, however as the averages are so close show that guess accuracy is actually similar for both systems. System B seems to be far better at predicting *tomato*, *elephant* and *jet*; however it has much more trouble with *sword*, *bullet* and *butterfly* These results seem to suggest that neither system is currently more accurate but that there are other factors in play here.



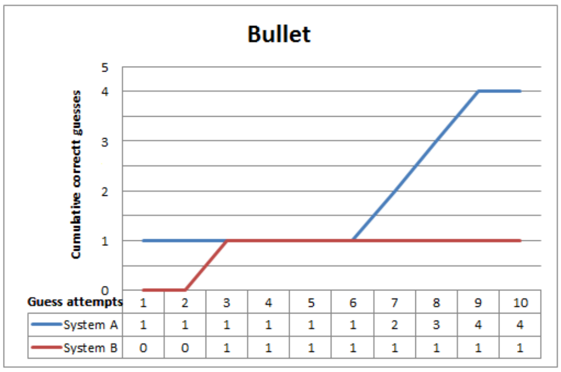


	System A		System B	
	Correct Guesses	System A	Correct Guesses	System B
bullet	1	0.1	4	0.4
butterfly	1	0.1	3	0.3
elephant	9	0.9	5	0.5
grape	2	0.2	2	0.2
jet	4	0.4	1	0.1
penguin	2	0.2	0	0
sword	1	0.1	6	0.6
tomato	9	0.9	5	0.5
average	3.625	0.3625	3.25	0.325

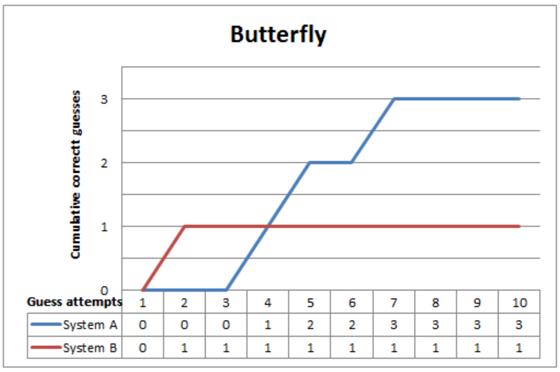
Figure 4.2: Average correct guesses over time in both systems.

4.3.3 Cumulative Correct Guesses over Time

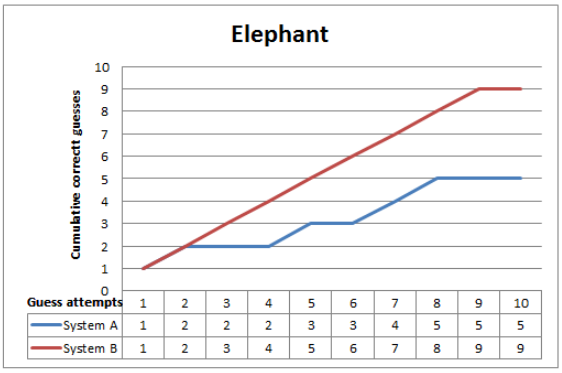
In order to try to prove Hypotheses 2, the cumulative frequencies of correct guesses were calculated for each concept and compared by system. *Arrow* and *parrot* have again been excluded. *Penguin* has ofcourse been kept as it was still in the original dataset for System A, even if it could not be predicted.



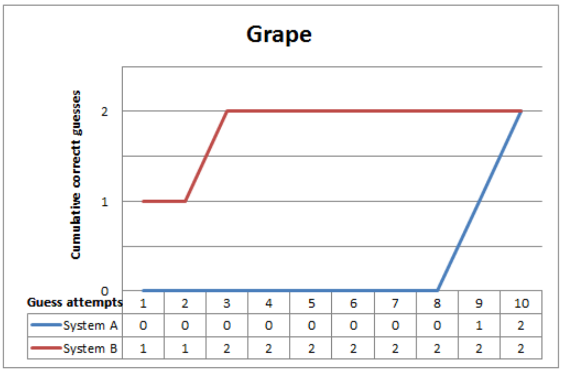
Guess attempts	1	2	3	4	5	6	7	8	9	10
System A	1	1	1	1	1	1	2	3	4	4
System B	0	0	1	1	1	1	1	1	1	1



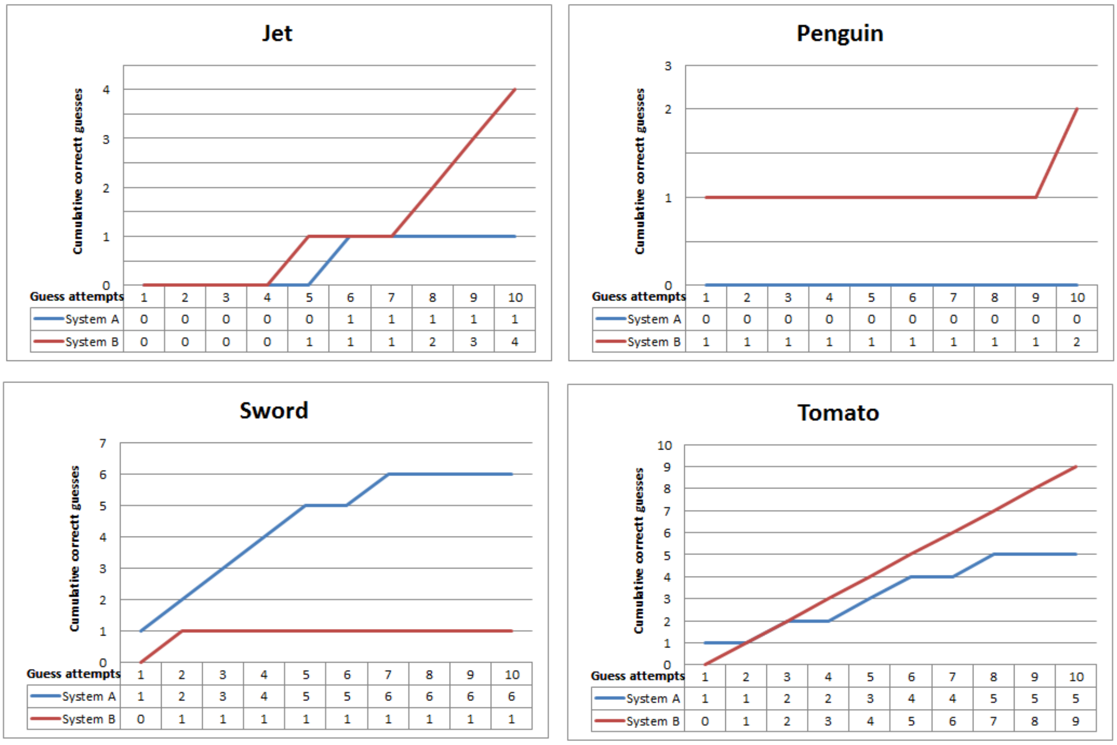
Guess attempts	1	2	3	4	5	6	7	8	9	10
System A	0	0	0	1	2	2	3	3	3	3
System B	0	1	1	1	1	1	1	1	1	1



Guess attempts	1	2	3	4	5	6	7	8	9	10
System A	1	2	2	2	3	3	4	5	5	5
System B	1	2	3	4	5	6	7	8	9	9



Guess attempts	1	2	3	4	5	6	7	8	9	10
System A	0	0	0	0	0	0	0	0	1	2
System B	1	1	2	2	2	2	2	2	2	2



Jet

Guess attempts	1	2	3	4	5	6	7	8	9	10
System A	0	0	0	0	0	1	1	1	1	1
System B	0	0	0	0	1	1	1	2	3	4

Penguin

Guess attempts	1	2	3	4	5	6	7	8	9	10
System A	0	0	0	0	0	0	0	0	0	0
System B	1	1	1	1	1	1	1	1	1	2

Sword

Guess attempts	1	2	3	4	5	6	7	8	9	10
System A	1	2	3	4	5	5	6	6	6	6
System B	0	1	1	1	1	1	1	1	1	1

Tomato

Guess attempts	1	2	3	4	5	6	7	8	9	10
System A	1	1	2	2	3	4	4	5	5	5
System B	0	1	2	3	4	5	6	7	8	9

Figure 4.3: Cumulative frequencies for correct guesses from both systems. Concepts *arrow* and *parrot* are not included as they did not exist in the original dataset for System A.

Results do vary for each concept, but again we do see *elephant*, *jet* and *tomato* as the best contenders, seemingly gaining the most out of system B's learning.

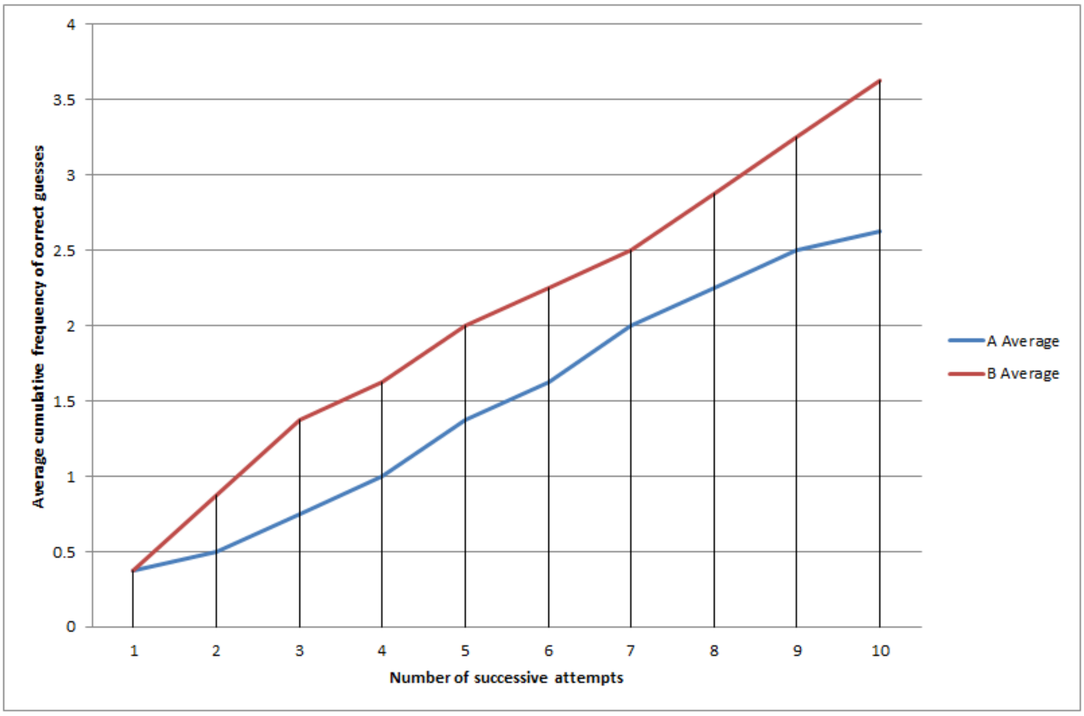
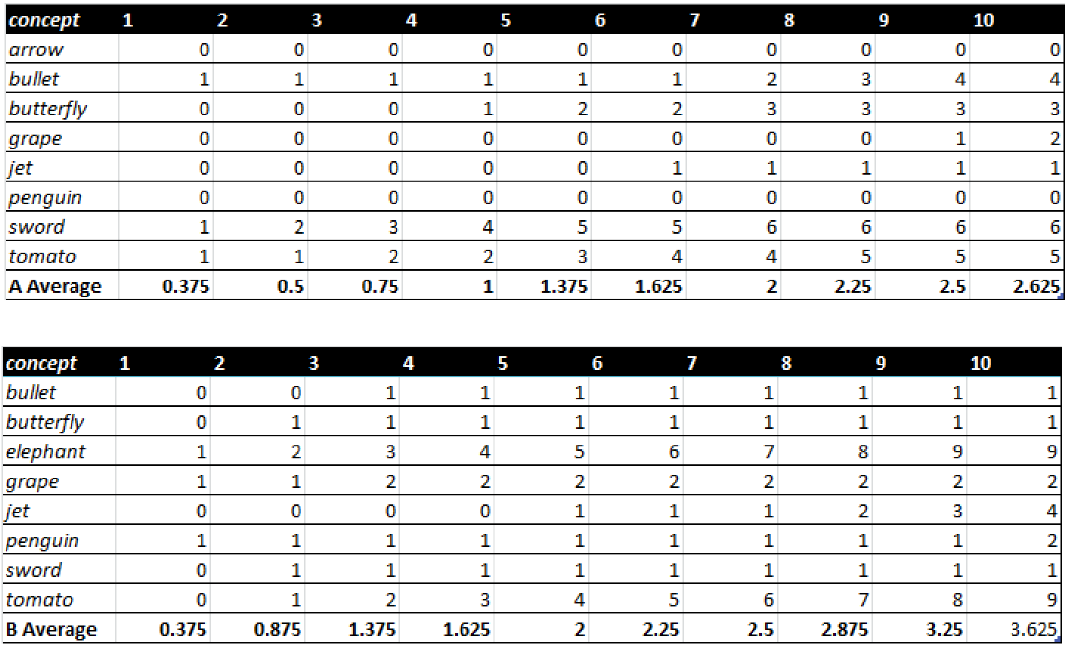


Figure 4.4: Average cumulative frequency of correct guesses for both systems

We do see that overall, System B has a steeper curve and this does seem to suggest that it is improving a little more steadily than System A. However more testing may need to be carried out and repeated further on a bigger sample to ensure these results remain consist.

4.3.4 Frequency of Natural Language Phrases in System B

We can see from Figure 4.5, that despite the aim for natural language, most questions were still heavily answered with only yes or no responses. Due to 'yes' and 'no' (represented as 1 and -1 respectively) being the most substantial multipliers in either direction, their disproportionate use may leave greater ramifications than expected. This seems similar to the issue with the Basic system throwing too many things away at once, too many instances of a negative one value can cause potential answers to move out of range.



concept	1	2	3	4	5	6	7	8	9	10
arrow	0	0	0	0	0	0	0	0	0	0
bullet	1	1	1	1	1	1	2	3	4	4
butterfly	0	0	0	1	2	2	3	3	3	3
grape	0	0	0	0	0	0	0	0	1	2
jet	0	0	0	0	0	1	1	1	1	1
penguin	0	0	0	0	0	0	0	0	0	0
sword	1	2	3	4	5	5	6	6	6	6
tomato	1	1	2	2	3	4	4	5	5	5
A Average	**0.375**	**0.5**	**0.75**	**1**	**1.375**	**1.625**	**2**	**2.25**	**2.5**	**2.625**

concept	1	2	3	4	5	6	7	8	9	10
bullet	0	0	1	1	1	1	1	1	1	1
butterfly	0	1	1	1	1	1	1	1	1	1
elephant	1	2	3	4	5	6	7	8	9	9
grape	1	1	2	2	2	2	2	2	2	2
jet	0	0	0	0	1	1	1	2	3	4
penguin	1	1	1	1	1	1	1	1	1	2
sword	0	1	1	1	1	1	1	1	1	1
tomato	0	1	2	3	4	5	6	7	8	9
B Average	**0.375**	**0.875**	**1.375**	**1.625**	**2**	**2.25**	**2.5**	**2.875**	**3.25**	**3.625**

Table 4.2: Full tabular view of cumulative correct guesses.

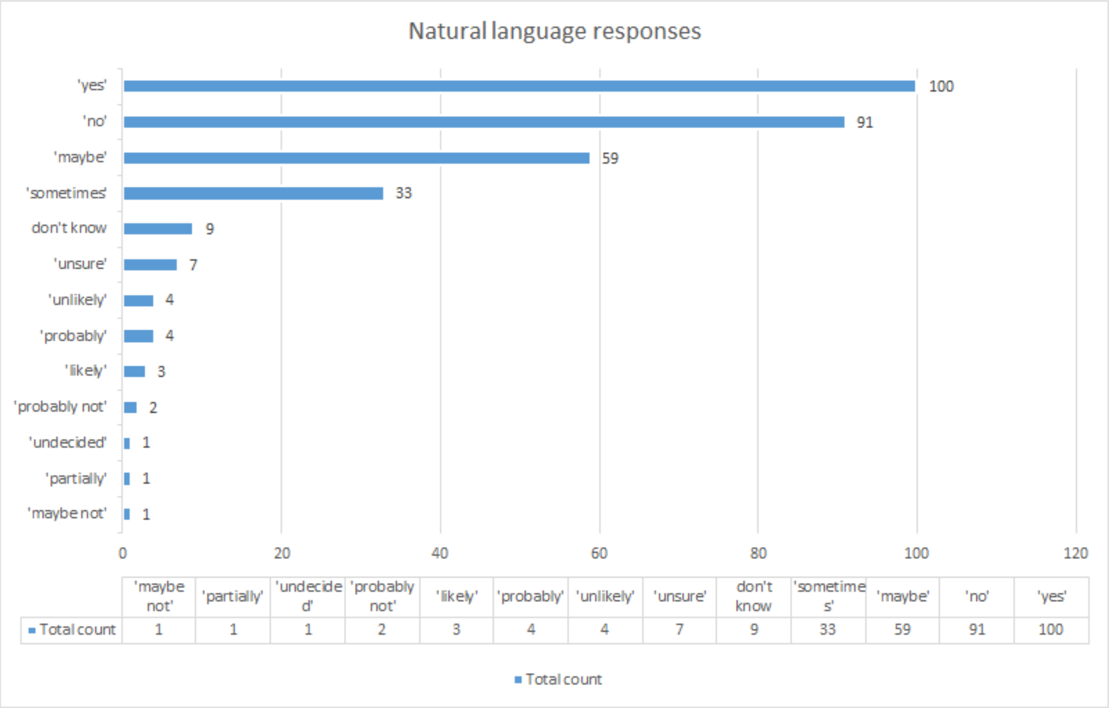


Figure 4.5: Frequency count of Natural Language answers in System B

4.3.5 Conclusion of Analysis

In order to conclude the investigation we must either accept or reject our hypotheses. Although system B is somewhat more accurate on average than system A, there is not as significant as we had expected so we must reject Hypotheses 1, instead we will accept the null hypothesis that "Learnable System B is not more accurate at guessing than Basic System A." If the experiment was repeated we may then accept the hypotheses.

Secondly we found that System B's graph measuring cumulative frequency of correct answers is somewhat steeper than System A's. We may accept the hypotheses that System B learns to be more accurate over time. Again though as to our small sample and the relative closeness of these results it may be worth rerunning the results.

5. Discussion

This section will cover a final discussion of the project, accounting for any limitations in the approach and look toward future work in the area.

5.1 Limitations

As we were unable to prove our initial hypotheses about System B is more accurate, we need to consider the implications of why. Highlighting the current limitations also allows us to raise awareness into a number of potential areas for improvement.

5.1.1 Natural language mapping limitation

This may be for a number of reasons such as the above stated findings about a Yes-No bias in lieu of natural language terms which can push the value to such an extreme range that can no longer be retrieved. This may be fixed by changing the assigned values for 'yes' and 'no' so they are no longer as polarised as '1' and '-1', instead putting them onto a more granular scale with more specific middle points. This may also have the added benefit of allowing modifiers such as "definitely" to adjust the scores further, "definitely" is justified in increasing polarity as that would be the intention. There are additional limitations with the current set of words, by tying everything to a five point scale you lose a lot of potential expression. Also there may be an ambiguity problem in words such as "maybe", without further context "maybe" may refer positively indicative of words such as "possibly" or "likely", or it could be more neutral in order to convey uncertainty. Of course an obvious next step would be to move on to even more natural sounding language, which would inherently add context in decided on meanings such as these. Another obvious limitation of the current system is that it cannot learn any new features, a natural language interpreter would be a step toward improving this.

5.1.2 Positive bias in the 'feature norms'

As the dataset is based on the 'feature norms' which come with a positive bias this may also be having an impact on results. The feature norms were collected to [23] show features present in a number of objects, there was no negative representation. The project system would take time to adapt to this in order to change the whole dataset to have an equal mix of neutral and negative relationships.

5.1.3 Termination problem

Within System B there is a question of when to Terminate as nothing is ever actually being removed as a possibility. For the current system the termination limit was set to having a difference of 150 over everything else. Of course there are issues with this when too many or too little questions have been asked and the dataset may never be cleared. More usually though this seems to manifest itself in guessing a wrong answer early when other suitable answers may still be high up in the candidate list. A potential solution for this may be to find an arbitrary limit to end the guessing after a certain number of turns. If the program has good enough source of knowledge then all it would be doing is building up evidence until reaching a point where it is confident. Finding an arbitrary number like this would also avoid having as long-winded a game, provided of course the database and/or program were good enough to know at guess at a reasonable enough point.

5.2 Future work

The is section will address potential areas for future work, beyond improvements based on the aforementioned limitations. The system may not have met all of its goals however there are still areas of expansion should the system improve.

5.2.1 Testing semantic vs visual features

I feel that more work could be done following this study in comparing visual and semantic features separately, rather tha using them all together in this system. The feature norms were all nominally semantic, having come from facts but it would be interesting to see a difference in testing those features referring exclusively to shape and visual appearance against those of functional or taxonomic properties, or further yet testing by different sensory properties as defined by Cree and McRae's 'brain region' categoriesCree2003. This shape vs. function bias was previously tested by [9] compared the

Studies such as [9, 20, 8, 26] have shown shape to be an important factor in object identification.

5.2.2 Multi-modal potential involving Vision system and "Spoken" dialogue.

There is potential for the system to be expanded into a multi-modal system able to process and interact through a number of different 'senses'. The system could be converted to enable spoken interaction, by combining it with speech recognition and text to speech modules.

It could also include a vision system to form a smarter multi-modal system much like which can understand both semantic and visual features. There is ongoing research in this area in computer vision, however there is yet to exist a large scale system which recognises major visual attributes in terms of real world features (eg the presence of wings) for a large dataset. A potential novel system would try to automatically predict an object's visual characteristics using an external vision system on say a photograph, before going on to ask feature questions to learn the semantic properties seperately. Over time the system may be able to start automatically predicting some semantic functions as well, eg if it sees wings it can predict that the object can fly without having to be explicitly told this by the user.

5.2.3 Browser-based learning

There is also potential in hosting the system online, so that it may be accessed by different people over the web, in this way it could build up a much more extensive dataset faster and more reflective of real world opinions. The system could also be adapted in to a Question Answering system based on the the way it handles discrepancies. With an improved learning algorithm, an online version like this would serve as a dynamic, adaptable and constantly evolving representation for feature knowledge, which could arguably be more accurate or relevant.

5.2.4 Expansion into other areas of Knowledge and Machine Learning

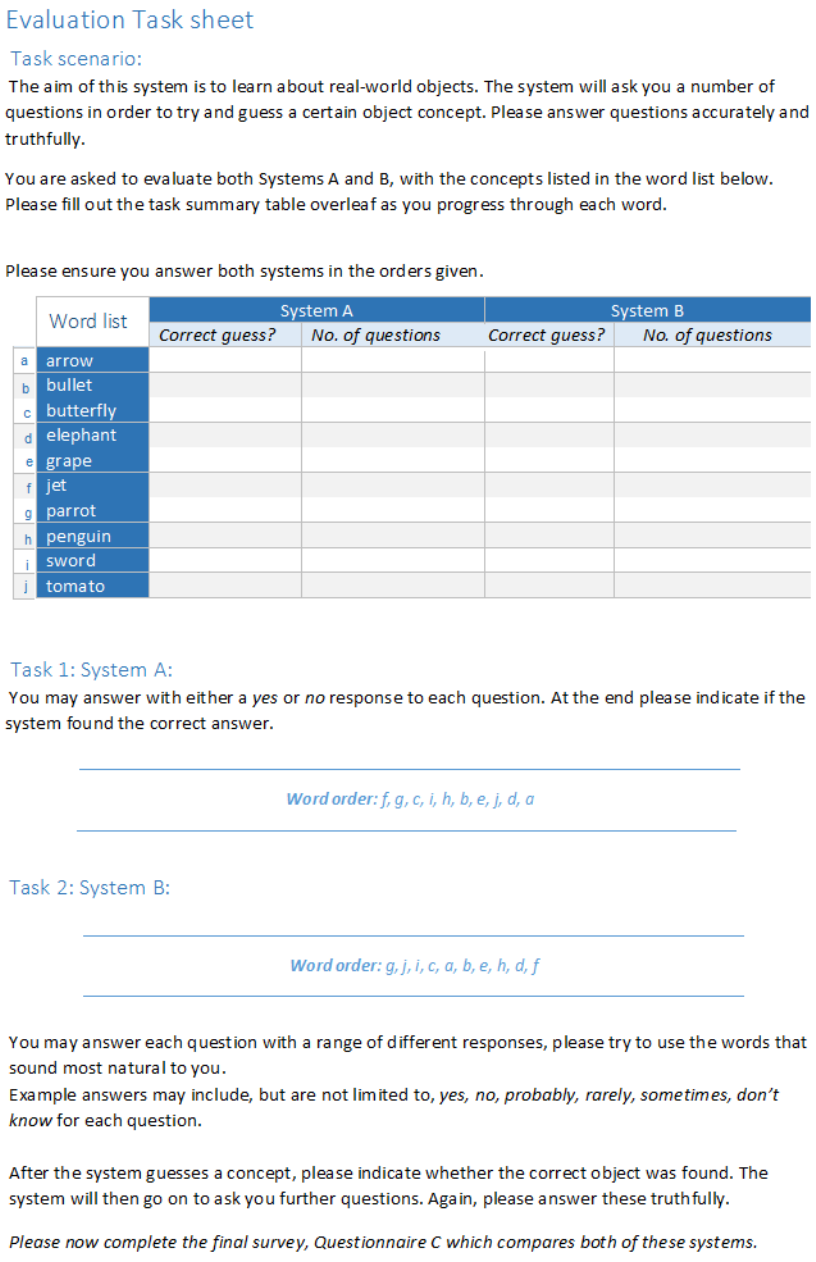
Decision making by the system could further be improved by making use of data-driven machine learning techniques. These would improve maintainability of the system as it would no longer be a long flow of decisions. Decision making would be drastically improved by a better more adaptable model of human behaviour.

Also the database could be expanded into a semantic ontology which could store inference relationships which could infer properties by being member of a certain taxonomic group (e.g. all mammals are warm-blooded) without being explicitly told.

5.3 Conclusion

To conclude the project looked at developing an object prediction system which has had some success at correctly identifying concepts. The system met it's implementation objectives in being able to guess, learn and generate questions. Although results were not conclusive in proving that guesses were more accurate after repeated learning over time, there does seem to be a benefit in repeating concepts on the system over time, as well as a number of suggestions for further improvements and future work.

A. Evaluation Task – Scenario Task sheet



Evaluation Task sheet

Task scenario:

The aim of this system is to learn about real-world objects. The system will ask you a number of questions in order to try and guess a certain object concept. Please answer questions accurately and truthfully.

You are asked to evaluate both Systems A and B, with the concepts listed in the word list below. Please fill out the task summary table overleaf as you progress through each word.

Please ensure you answer both systems in the orders given.

	Word list	System A		System B	
		Correct guess?	*No. of questions*	*Correct guess?*	*No. of questions*
a	arrow				
b	bullet				
c	butterfly				
d	elephant				
e	grape				
f	jet				
g	parrot				
h	penguin				
i	sword				
j	tomato				

Task 1: System A:

You may answer with either a *yes* or *no* response to each question. At the end please indicate if the system found the correct answer.

***Word order:** f, g, c, i, h, b, e, j, d, a*

Task 2: System B:

***Word order:** g, j, i, c, a, b, e, h, d, f*

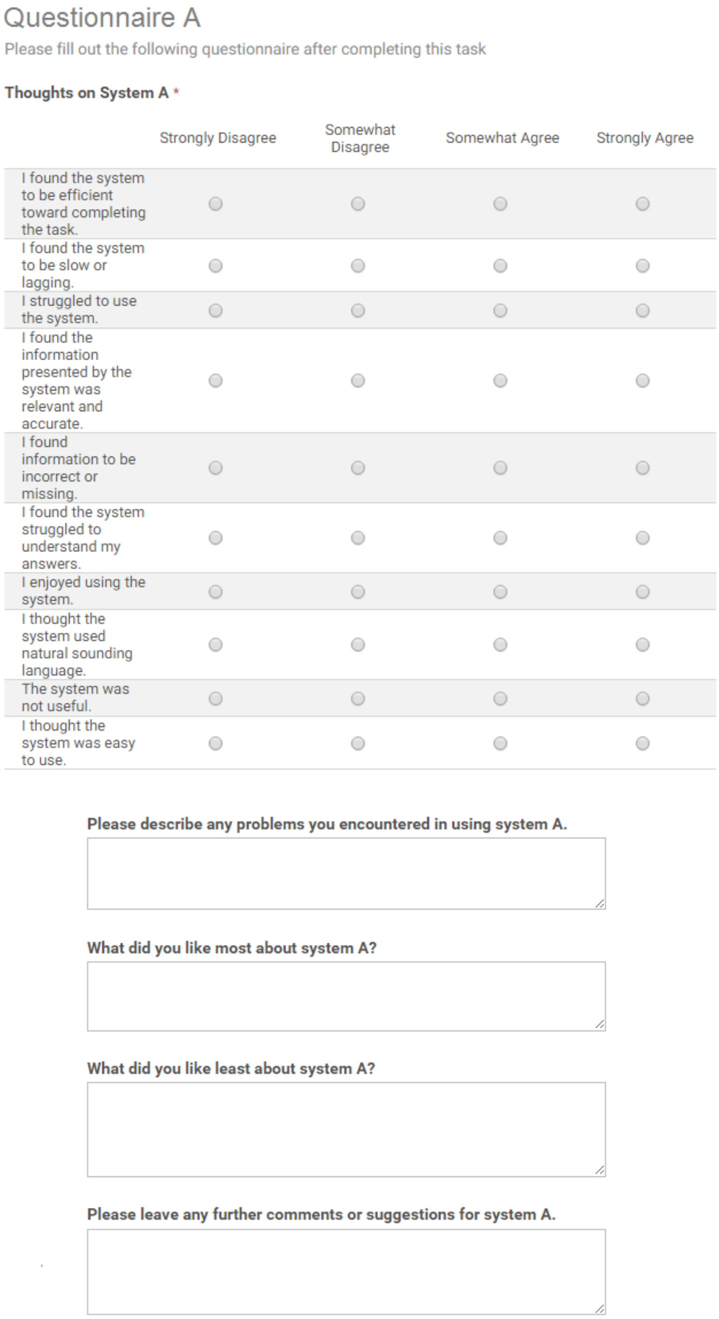
You may answer each question with a range of different responses, please try to use the words that sound most natural to you.
Example answers may include, but are not limited to, *yes, no, probably, rarely, sometimes, don't know* for each question.

After the system guesses a concept, please indicate whether the correct object was found. The system will then go on to ask you further questions. Again, please answer these truthfully.

Please now complete the final survey, Questionnaire C which compares both of these systems.

Evaluation task sheet

B. Evaluation Task – Questionnaires A and B



Questionnaire A

Please fill out the following questionnaire after completing this task

Thoughts on System A *

	Strongly Disagree	Somewhat Disagree	Somewhat Agree	Strongly Agree
I found the system to be efficient toward completing the task.	○	○	○	○
I found the system to be slow or lagging.	○	○	○	○
I struggled to use the system.	○	○	○	○
I found the information presented by the system was relevant and accurate.	○	○	○	○
I found information to be incorrect or missing.	○	○	○	○
I found the system struggled to understand my answers.	○	○	○	○
I enjoyed using the system.	○	○	○	○
I thought the system used natural sounding language.	○	○	○	○
The system was not useful.	○	○	○	○
I thought the system was easy to use.	○	○	○	○

Please describe any problems you encountered in using system A.

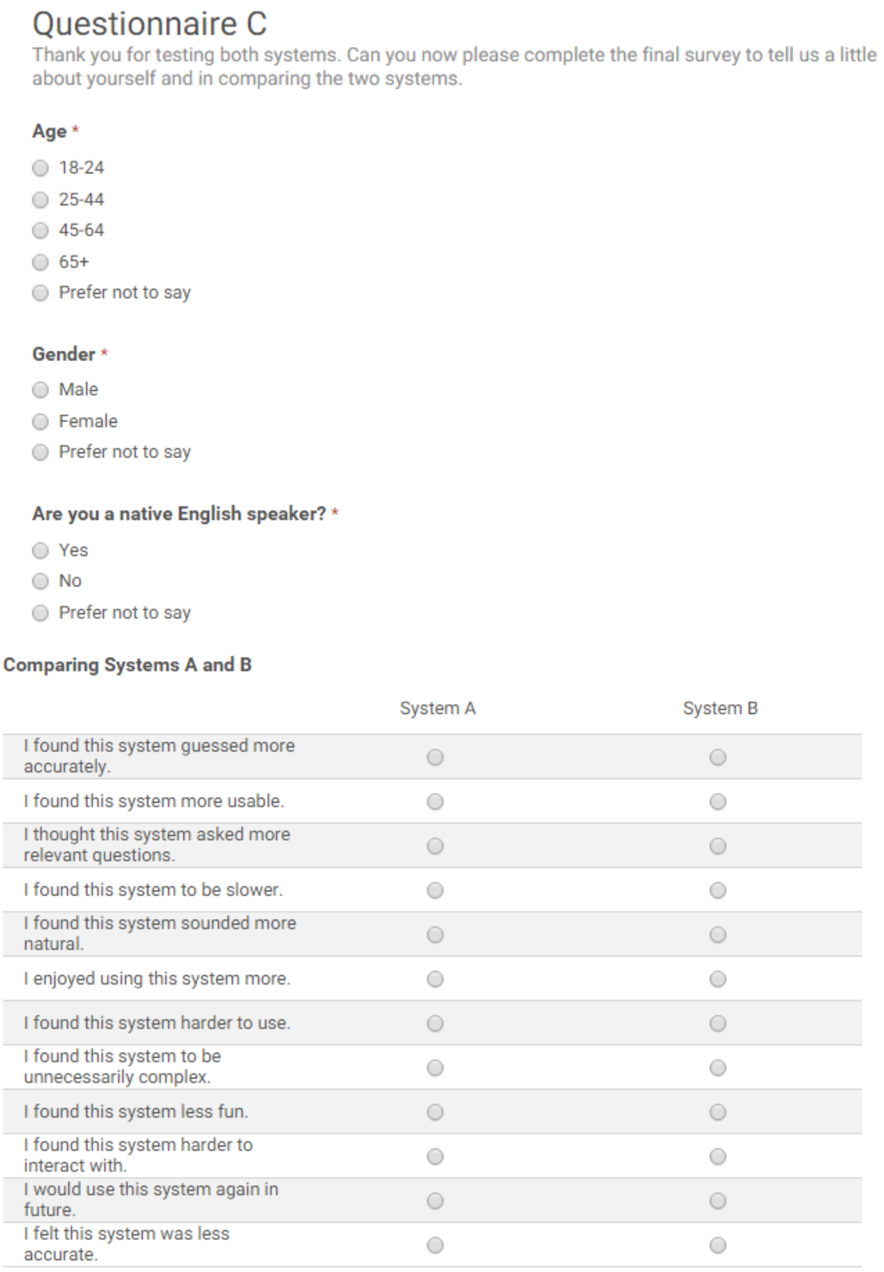
What did you like most about system A?

What did you like least about system A?

Please leave any further comments or suggestions for system A.

Questionnaires A and B. Both questionnaires asked the same questions.

C. Evaluation Task – Questionnaire C



Questionnaire C

Thank you for testing both systems. Can you now please complete the final survey to tell us a little about yourself and in comparing the two systems.

Age *

○ 18-24
○ 25-44
○ 45-64
○ 65+
○ Prefer not to say

Gender *

○ Male
○ Female
○ Prefer not to say

Are you a native English speaker? *

○ Yes
○ No
○ Prefer not to say

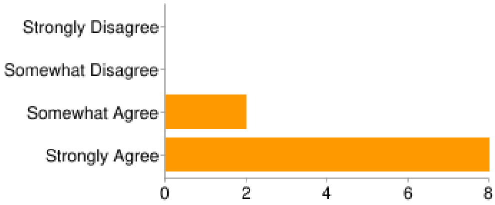
Comparing Systems A and B

	System A	System B
I found this system guessed more accurately.	○	○
I found this system more usable.	○	○
I thought this system asked more relevant questions.	○	○
I found this system to be slower.	○	○
I found this system sounded more natural.	○	○
I enjoyed using this system more.	○	○
I found this system harder to use.	○	○
I found this system to be unnecessarily complex.	○	○
I found this system less fun.	○	○
I found this system harder to interact with.	○	○
I would use this system again in future.	○	○
I felt this system was less accurate.	○	○

Questionnaire C, which compared the two systems directly.

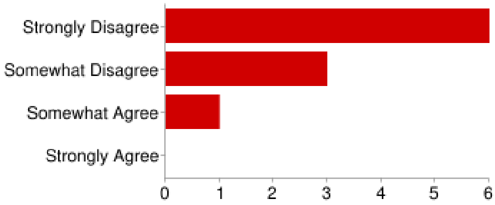
D. Evaluation Results Summary – Questionnaire A

I thought the system was easy to use. [Thoughts on System A]



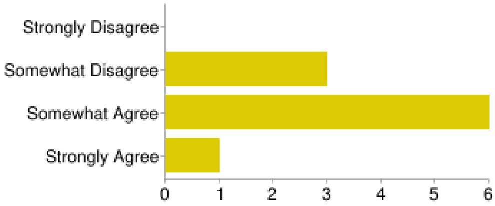
Strongly Disagree	**0**	0%
Somewhat Disagree	**0**	0%
Somewhat Agree	**2**	20%
Strongly Agree	**8**	80%

I found the system to be slow or lagging. [Thoughts on System A]



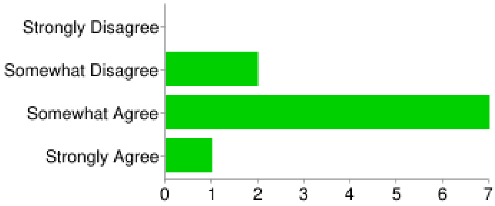
Strongly Disagree	**6**	60%
Somewhat Disagree	**3**	30%
Somewhat Agree	**1**	10%
Strongly Agree	**0**	0%

I found information to be incorrect or missing. [Thoughts on System A]



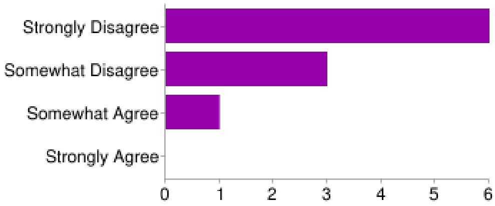
Strongly Disagree	**0**	0%
Somewhat Disagree	**3**	30%
Somewhat Agree	**6**	60%
Strongly Agree	**1**	10%

I enjoyed using the system. [Thoughts on System A]



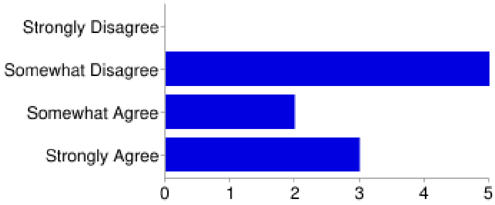
Strongly Disagree	**0**	0%
Somewhat Disagree	**2**	20%
Somewhat Agree	**7**	70%
Strongly Agree	**1**	10%

I struggled to use the system. [Thoughts on System A]



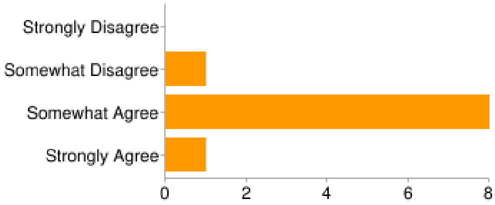
Strongly Disagree	**6**	60%
Somewhat Disagree	**3**	30%
Somewhat Agree	**1**	10%
Strongly Agree	**0**	0%

I found the system to be efficient toward completing the task. [Thoughts on System A]



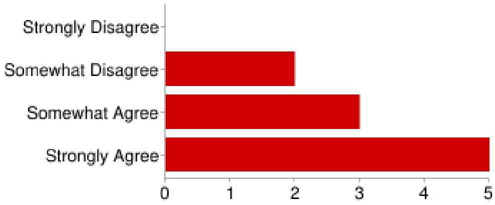
Strongly Disagree	**0**	0%
Somewhat Disagree	**5**	50%
Somewhat Agree	**2**	20%
Strongly Agree	**3**	30%

I found the information presented by the system was relevant and accurate. [Thoughts on System A]



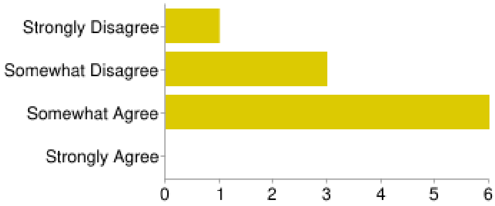
Strongly Disagree	**0**	0%
Somewhat Disagree	**1**	10%
Somewhat Agree	**8**	80%
Strongly Agree	**1**	10%

I thought the system used natural sounding language. [Thoughts on System A]



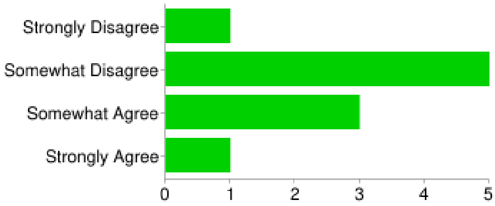
Strongly Disagree	**0**	0%
Somewhat Disagree	**2**	20%
Somewhat Agree	**3**	30%
Strongly Agree	**5**	50%

I found the system struggled to understand my answers. [Thoughts on System A]



Strongly Disagree	**1**	10%
Somewhat Disagree	**3**	30%
Somewhat Agree	**6**	60%
Strongly Agree	**0**	0%

The system was not useful. [Thoughts on System A]



Strongly Disagree	**1**	10%
Somewhat Disagree	**5**	50%
Somewhat Agree	**3**	30%
Strongly Agree	**1**	10%

What did you like most about system A?

- Questions are clear and easy to understand
- not too many questions
- can get the result fast
- I liked how easy the system was to use, using one word answer.
- it was easy to understand, language was relevant to getting task completed in an easy and simple manner
- Easy to use functionality. Questions are relevant, although sometimes answer might seem unclear.
- It's fun to use and sometimes asks the most unexpected questions.
- Was faster than system B

Please describe any problems you encountered in using system A.

- many of the questions could have had a possible yes/no answer, although only one could be entered
- some question can not answer by yes or no
- When System A didn't get the right answer the answer was either not in the database or had previously marked the correct answer out. At times, I was also unsure as to the correct answer to a questions and uncertain if this had a profound effect on the outcome of the program.
- Instead of having to keep typing yes or no, it should have an option to let us click yes or no. much better for interactivity
- no probelms
- It did not guess the objects I was thinking about.
- Questions that are relative hard to answer eg:Is it small?
- limited range of possible answers
- One problem I have encountered is that at certain times the question illogically repeat itself. Also having to type out the long answers.

What did you like least about system A?

- that i need to keep typing yes or no
- typing
- kept asking similar questions. For examples is it metal? then asked is it plastic. Even tho it was a metal.
- Database was incomplete and unsure if one incorrect answer to a series of questions lead to program eliminating correct answer
- typing out the answers
- many of the objects I had thought of whlst entering yes/no did not appear to be what the program found
- the result is not always correct
- Some questions are a bit hard to answer to because I was not sure about the right answer.
- wasn't very good at guessing

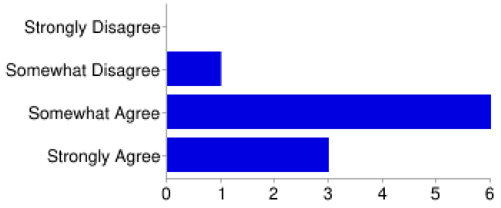
Please leave any further comments or suggestions for system A.

- It would be great if it could guess the answer with less questions. If for example it finds a hit, it could display its guess. Then when its not correct, it keeps asking questions.
- Split questions into groups such as material, colour and so on so question aren't repeated.
- some information is incorrect!
- Be more forgiving on possible mistakes made from the user. Keep asking questions after the wrong guess
- have a middle way - 'either' option
- if you could add yes or no button and may be make it visually more apealing it would be great.

An interactive interface would be nice

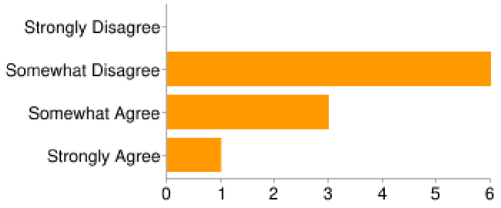
E. Evaluation Results Summary – Questionnaire B

I thought the system was easy to use. [Thoughts on System B]



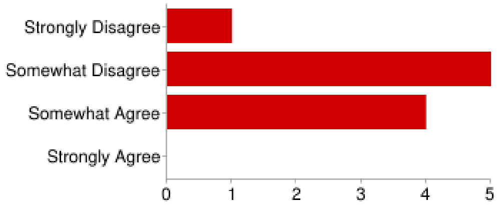
Strongly Disagree	**0**	0%
Somewhat Disagree	**1**	10%
Somewhat Agree	**6**	60%
Strongly Agree	**3**	30%

I found the system to be slow or lagging. [Thoughts on System B]



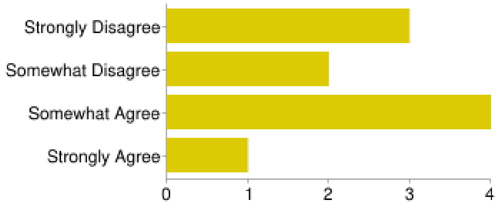
Strongly Disagree	**0**	0%
Somewhat Disagree	**6**	60%
Somewhat Agree	**3**	30%
Strongly Agree	**1**	10%

I found information to be incorrect or missing. [Thoughts on System B]



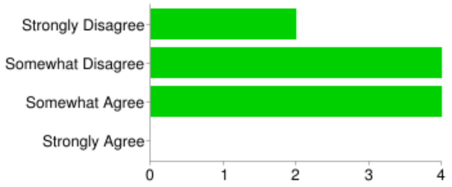
Strongly Disagree	**1**	10%
Somewhat Disagree	**5**	50%
Somewhat Agree	**4**	40%
Strongly Agree	**0**	0%

I enjoyed using the system. [Thoughts on System B]



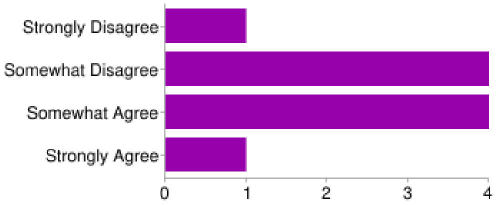
Strongly Disagree	**3**	30%
Somewhat Disagree	**2**	20%
Somewhat Agree	**4**	40%
Strongly Agree	**1**	10%

I struggled to use the system. [Thoughts on System B]



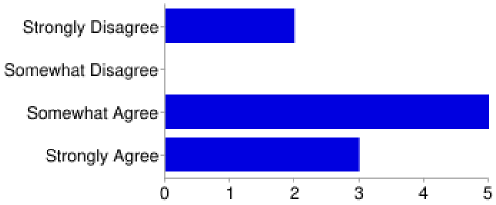
Strongly Disagree	**2**	20%
Somewhat Disagree	**4**	40%
Somewhat Agree	**4**	40%
Strongly Agree	**0**	0%

I found the system to be efficient toward completing the task. [Thoughts on System B]



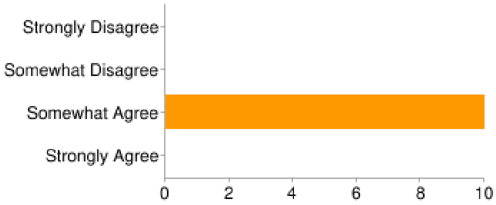
Strongly Disagree	**1**	10%
Somewhat Disagree	**4**	40%
Somewhat Agree	**4**	40%
Strongly Agree	**1**	10%

I found the information presented by the system was relevant and accurate. [Thoughts on System B]



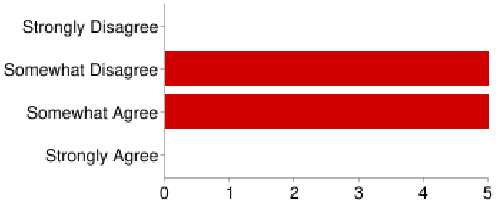
Strongly Disagree	**2**	20%
Somewhat Disagree	**0**	0%
Somewhat Agree	**5**	50%
Strongly Agree	**3**	30%

I thought the system used natural sounding language. [Thoughts on System B]



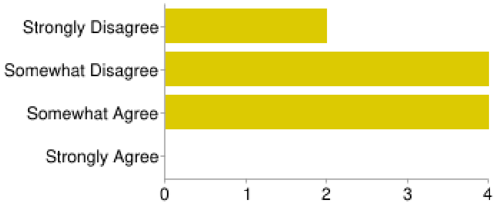
Strongly Disagree	**0**	0%
Somewhat Disagree	**0**	0%
Somewhat Agree	**10**	100%
Strongly Agree	**0**	0%

I found the system struggled to understand my answers. [Thoughts on System B]

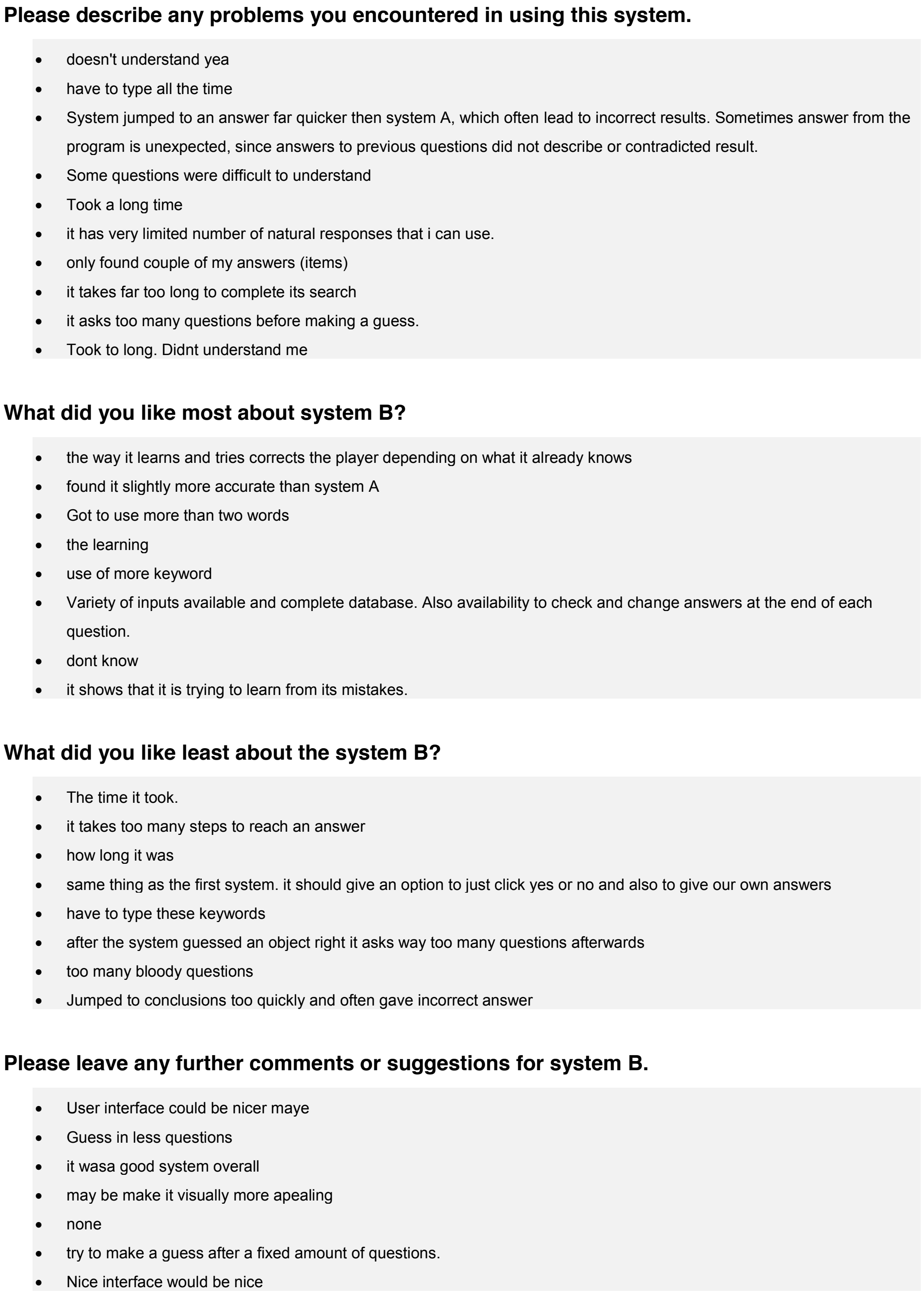


Strongly Disagree	**0**	0%
Somewhat Disagree	**5**	50%
Somewhat Agree	**5**	50%
Strongly Agree	**0**	0%

The system was not useful. [Thoughts on System B]



Strongly Disagree	**2**	20%
Somewhat Disagree	**4**	40%
Somewhat Agree	**4**	40%
Strongly Agree	**0**	0%



Please describe any problems you encountered in using this system.

- doesn't understand yea
- have to type all the time
- System jumped to an answer far quicker then system A, which often lead to incorrect results. Sometimes answer from the program is unexpected, since answers to previous questions did not describe or contradicted result.
- Some questions were difficult to understand
- Took a long time
- it has very limited number of natural responses that i can use.
- only found couple of my answers (items)
- it takes far too long to complete its search
- it asks too many questions before making a guess.
- Took to long. Didnt understand me

What did you like most about system B?

- the way it learns and tries corrects the player depending on what it already knows
- found it slightly more accurate than system A
- Got to use more than two words
- the learning
- use of more keyword
- Variety of inputs available and complete database. Also availability to check and change answers at the end of each question.
- dont know
- it shows that it is trying to learn from its mistakes.

What did you like least about the system B?

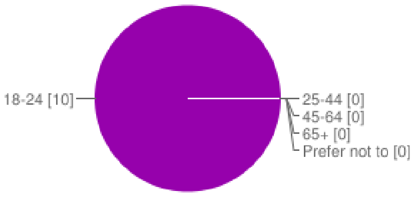
- The time it took.
- it takes too many steps to reach an answer
- how long it was
- same thing as the first system. it should give an option to just click yes or no and also to give our own answers
- have to type these keywords
- after the system guessed an object right it asks way too many questions afterwards
- too many bloody questions
- Jumped to conclusions too quickly and often gave incorrect answer

Please leave any further comments or suggestions for system B.

- User interface could be nicer maye
- Guess in less questions
- it wasa good system overall
- may be make it visually more apealing
- none
- try to make a guess after a fixed amount of questions.
- Nice interface would be nice

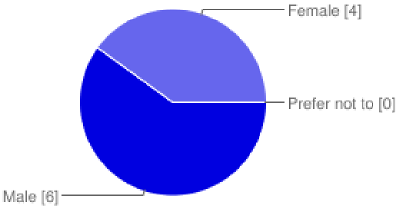
F. Evaluation Results Summary – Questionnaire C

Age



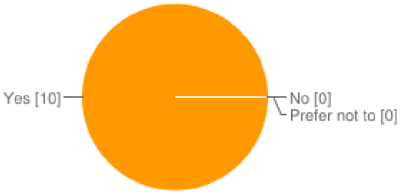
18-24	**10**	100%
25-44	**0**	0%
45-64	**0**	0%
65+	**0**	0%
Prefer not to say	**0**	0%

Gender



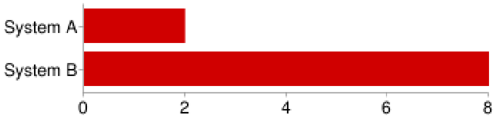
Male	**6**	60%
Female	**4**	40%
Prefer not to say	**0**	0%

Are you a native English speaker?



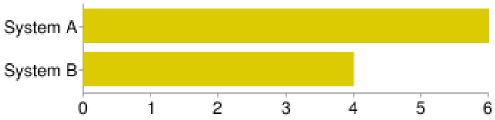
Yes	**10**	100%
No	**0**	0%
Prefer not to say	**0**	0%

I found this system guessed more accurately. [Comparing Systems A and B]



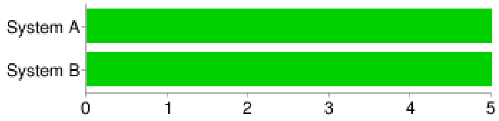
System A	**2**	20%
System B	**8**	80%

I found this system more usable. [Comparing Systems A and B]



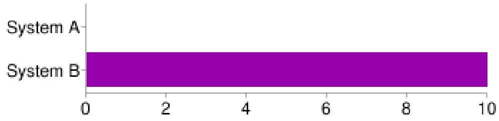
System A	**6**	60%
System B	**4**	40%

I thought this system asked more relevant questions. [Comparing Systems A and B]



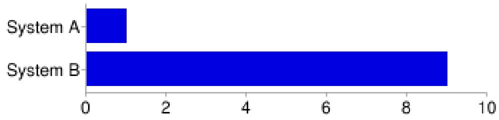
System A	**5**	50%
System B	**5**	50%

I found this system to be slower. [Comparing Systems A and B]



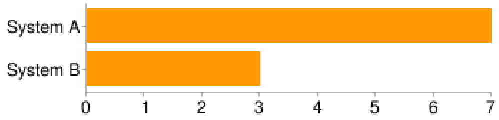
System A	**0**	0%
System B	**10**	100%

I found this system sounded more natural. [Comparing Systems A and B]



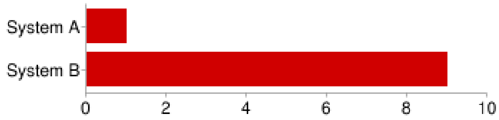
System A	**1**	10%
System B	**9**	90%

I enjoyed using this system more. [Comparing Systems A and B]



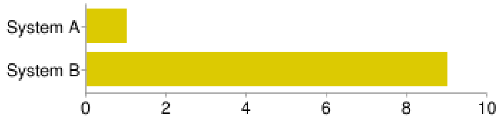
System A	**7**	70%
System B	**3**	30%

I found this system harder to use. [Comparing Systems A and B]



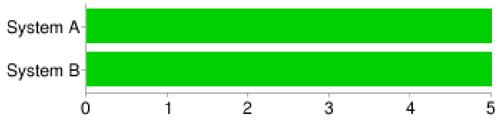
System A	**1**	10%
System B	**9**	90%

I found this system to be unnecessarily complex. [Comparing Systems A and B]



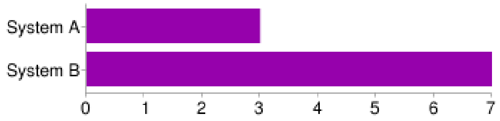
System A	**1**	10%
System B	**9**	90%

I found this system less fun. [Comparing Systems A and B]



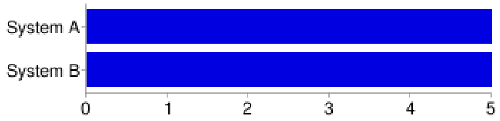
System A	**5**	50%
System B	**5**	50%

I found this system harder to interact with. [Comparing Systems A and B]



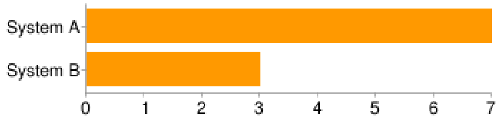
System A	**3**	30%
System B	**7**	70%

I would use this system again in future. [Comparing Systems A and B]



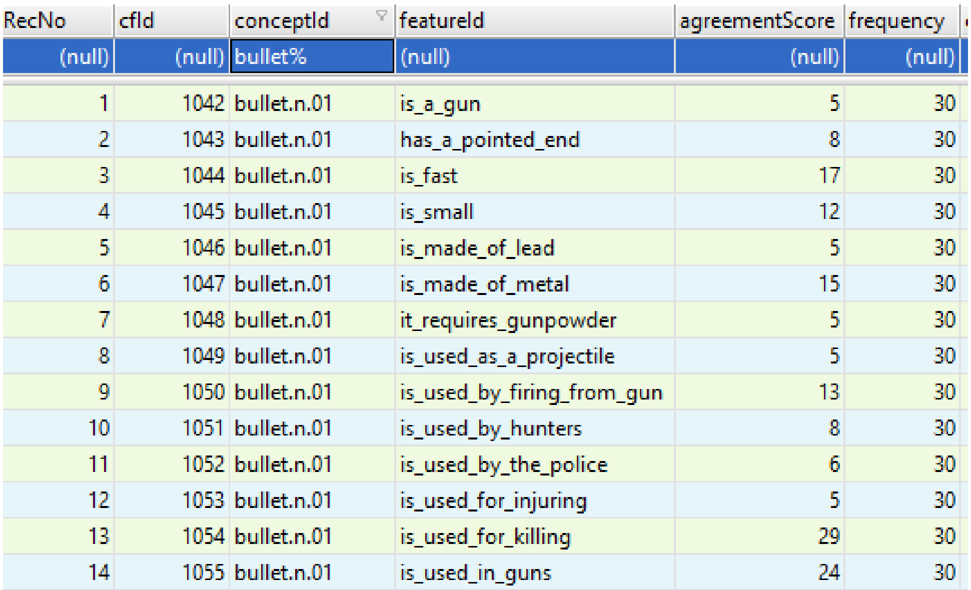
System A	**5**	50%
System B	**5**	50%

I felt this system was less accurate. [Comparing Systems A and B]

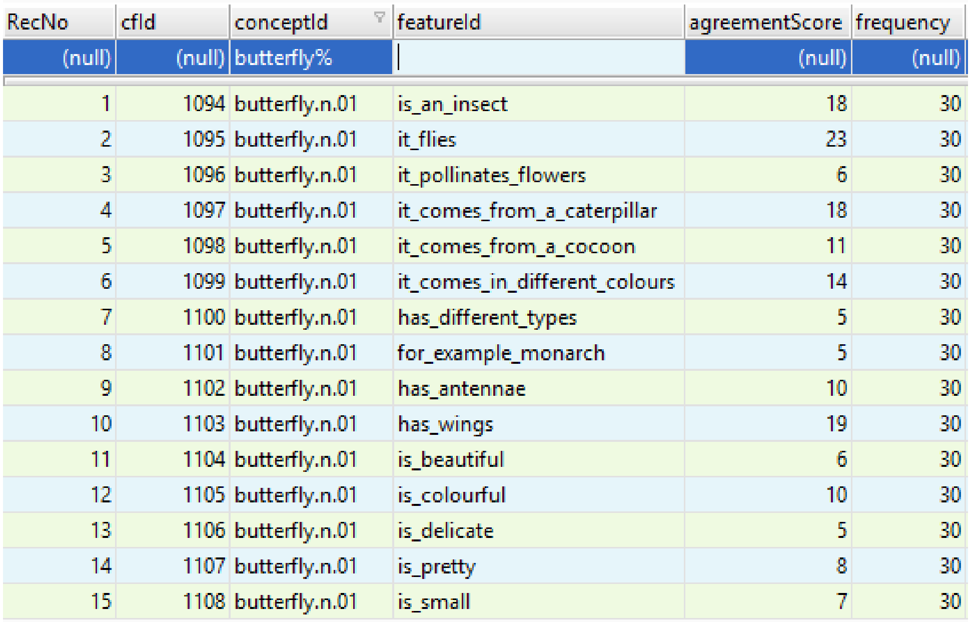


System A	**7**	70%
System B	**3**	30%

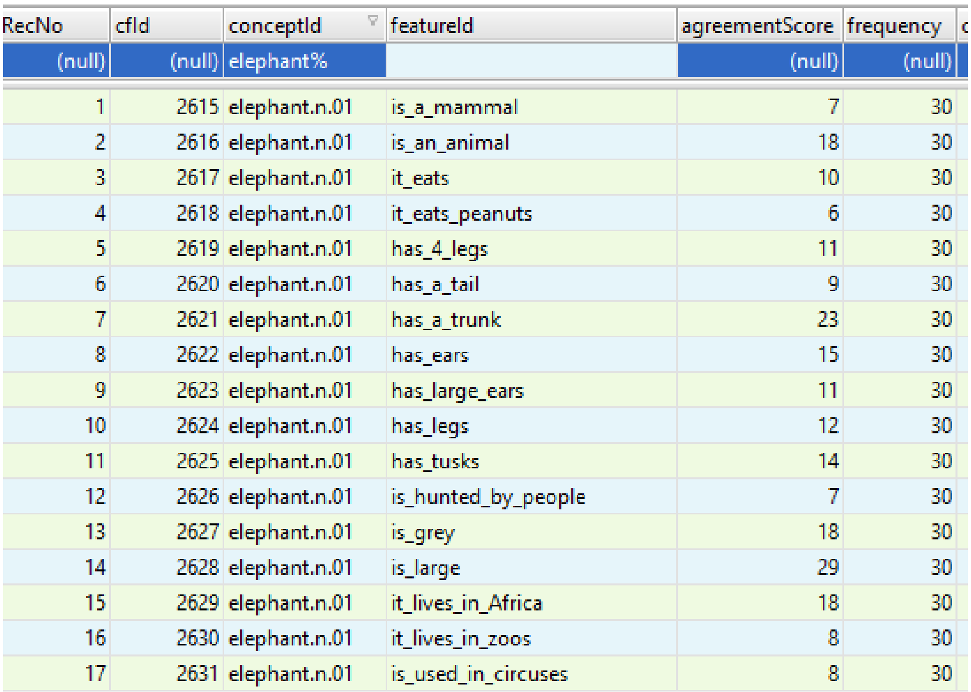
G. Original Features for Chosen Concepts



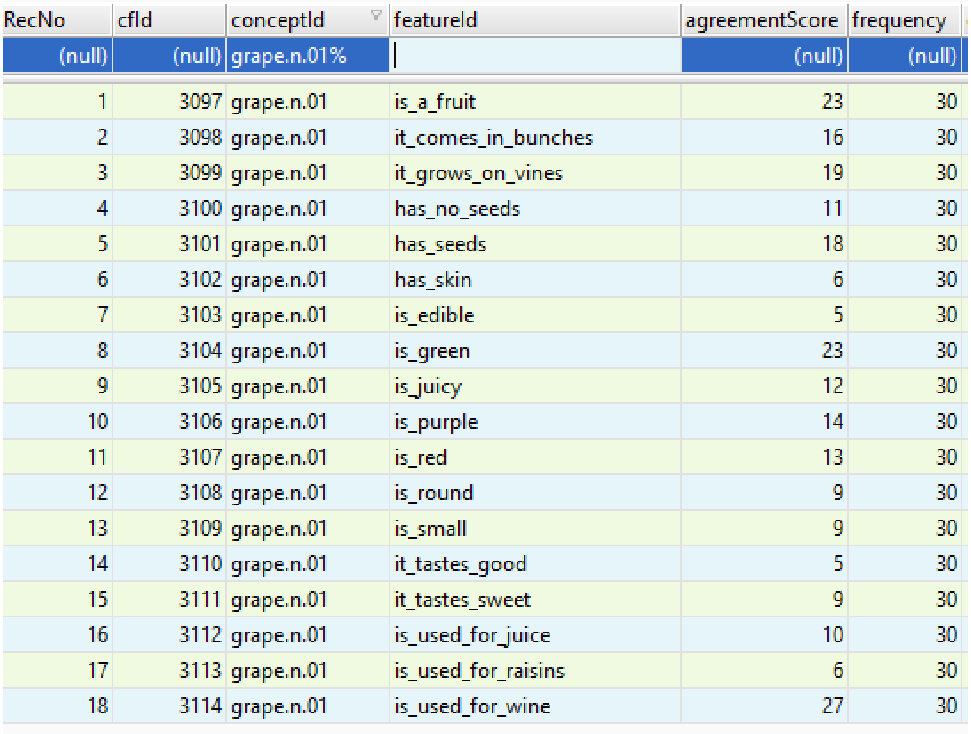
RecNo	cfId	conceptId	featureId	agreementScore	frequency
(null)	(null)	bullet%	(null)	(null)	(null)
1	1042	bullet.n.01	is_a_gun	5	30
2	1043	bullet.n.01	has_a_pointed_end	8	30
3	1044	bullet.n.01	is_fast	17	30
4	1045	bullet.n.01	is_small	12	30
5	1046	bullet.n.01	is_made_of_lead	5	30
6	1047	bullet.n.01	is_made_of_metal	15	30
7	1048	bullet.n.01	it_requires_gunpowder	5	30
8	1049	bullet.n.01	is_used_as_a_projectile	5	30
9	1050	bullet.n.01	is_used_by_firing_from_gun	13	30
10	1051	bullet.n.01	is_used_by_hunters	8	30
11	1052	bullet.n.01	is_used_by_the_police	6	30
12	1053	bullet.n.01	is_used_for_injuring	5	30
13	1054	bullet.n.01	is_used_for_killing	29	30
14	1055	bullet.n.01	is_used_in_guns	24	30



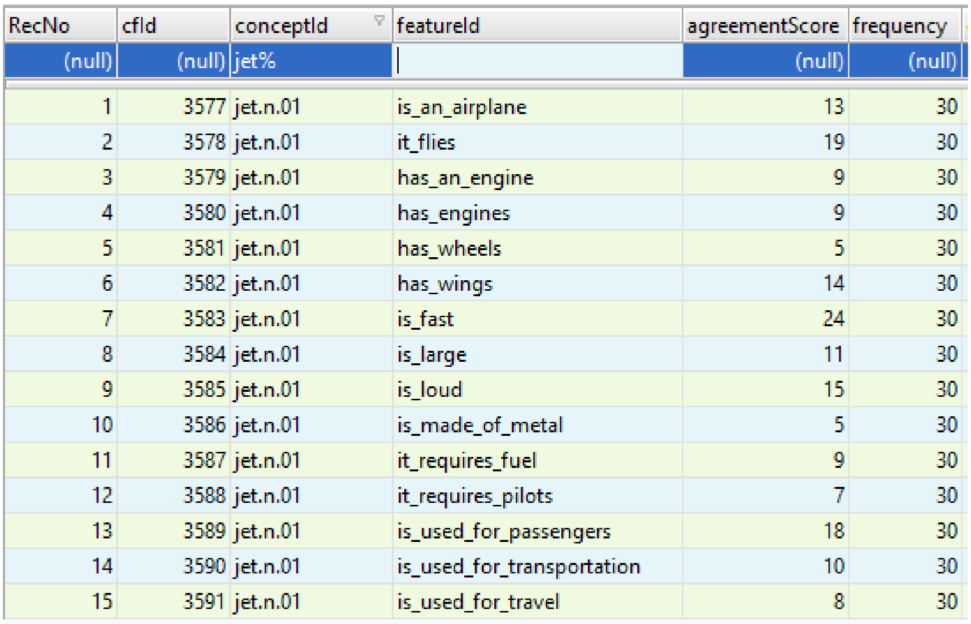
RecNo	cfId	conceptId	featureId	agreementScore	frequency
(null)	(null)	butterfly%		(null)	(null)
1	1094	butterfly.n.01	is_an_insect	18	30
2	1095	butterfly.n.01	it_flies	23	30
3	1096	butterfly.n.01	it_pollinates_flowers	6	30
4	1097	butterfly.n.01	it_comes_from_a_caterpillar	18	30
5	1098	butterfly.n.01	it_comes_from_a_cocoon	11	30
6	1099	butterfly.n.01	it_comes_in_different_colours	14	30
7	1100	butterfly.n.01	has_different_types	5	30
8	1101	butterfly.n.01	for_example_monarch	5	30
9	1102	butterfly.n.01	has_antennae	10	30
10	1103	butterfly.n.01	has_wings	19	30
11	1104	butterfly.n.01	is_beautiful	6	30
12	1105	butterfly.n.01	is_colourful	10	30
13	1106	butterfly.n.01	is_delicate	5	30
14	1107	butterfly.n.01	is_pretty	8	30
15	1108	butterfly.n.01	is_small	7	30



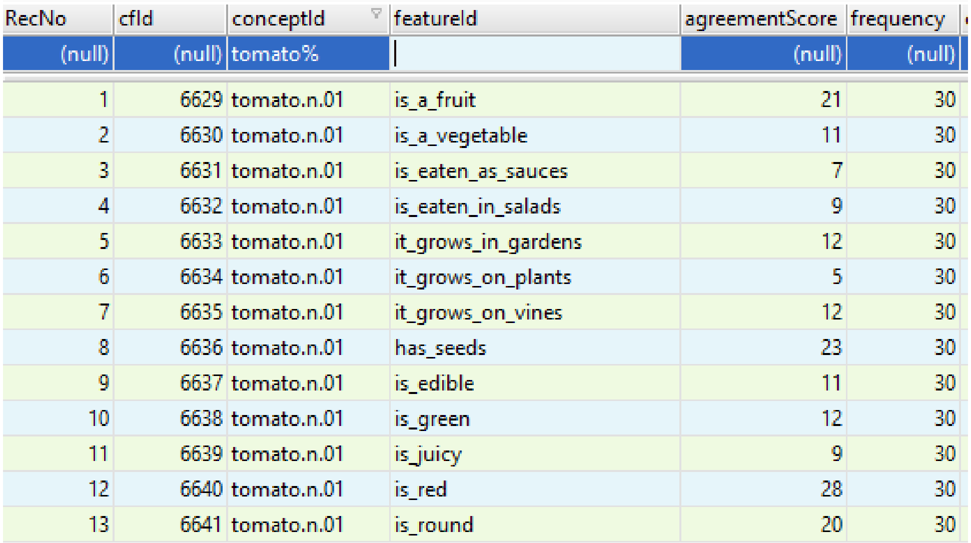
RecNo	cfId	conceptId	featureId	agreementScore	frequency
(null)	(null)	elephant%		(null)	(null)
1	2615	elephant.n.01	is_a_mammal	7	30
2	2616	elephant.n.01	is_an_animal	18	30
3	2617	elephant.n.01	it_eats	10	30
4	2618	elephant.n.01	it_eats_peanuts	6	30
5	2619	elephant.n.01	has_4_legs	11	30
6	2620	elephant.n.01	has_a_tail	9	30
7	2621	elephant.n.01	has_a_trunk	23	30
8	2622	elephant.n.01	has_ears	15	30
9	2623	elephant.n.01	has_large_ears	11	30
10	2624	elephant.n.01	has_legs	12	30
11	2625	elephant.n.01	has_tusks	14	30
12	2626	elephant.n.01	is_hunted_by_people	7	30
13	2627	elephant.n.01	is_grey	18	30
14	2628	elephant.n.01	is_large	29	30
15	2629	elephant.n.01	it_lives_in_Africa	18	30
16	2630	elephant.n.01	it_lives_in_zoos	8	30
17	2631	elephant.n.01	is_used_in_circuses	8	30



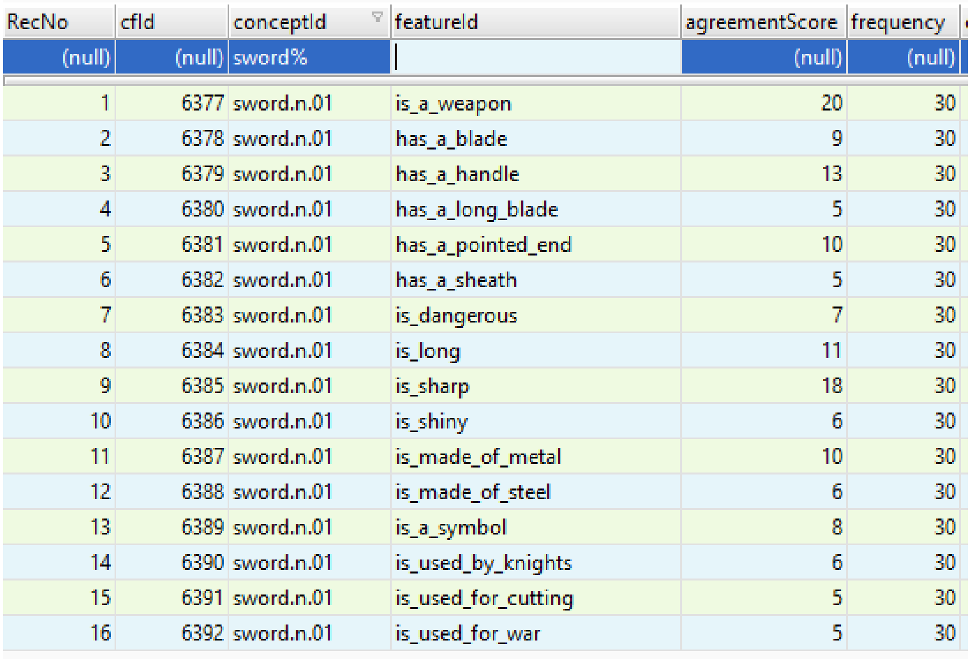
RecNo	cfId	conceptId	featureId	agreementScore	frequency
(null)	(null)	grape.n.01%	|	(null)	(null)
1	3097	grape.n.01	is_a_fruit	23	30
2	3098	grape.n.01	it_comes_in_bunches	16	30
3	3099	grape.n.01	it_grows_on_vines	19	30
4	3100	grape.n.01	has_no_seeds	11	30
5	3101	grape.n.01	has_seeds	18	30
6	3102	grape.n.01	has_skin	6	30
7	3103	grape.n.01	is_edible	5	30
8	3104	grape.n.01	is_green	23	30
9	3105	grape.n.01	is_juicy	12	30
10	3106	grape.n.01	is_purple	14	30
11	3107	grape.n.01	is_red	13	30
12	3108	grape.n.01	is_round	9	30
13	3109	grape.n.01	is_small	9	30
14	3110	grape.n.01	it_tastes_good	5	30
15	3111	grape.n.01	it_tastes_sweet	9	30
16	3112	grape.n.01	is_used_for_juice	10	30
17	3113	grape.n.01	is_used_for_raisins	6	30
18	3114	grape.n.01	is_used_for_wine	27	30



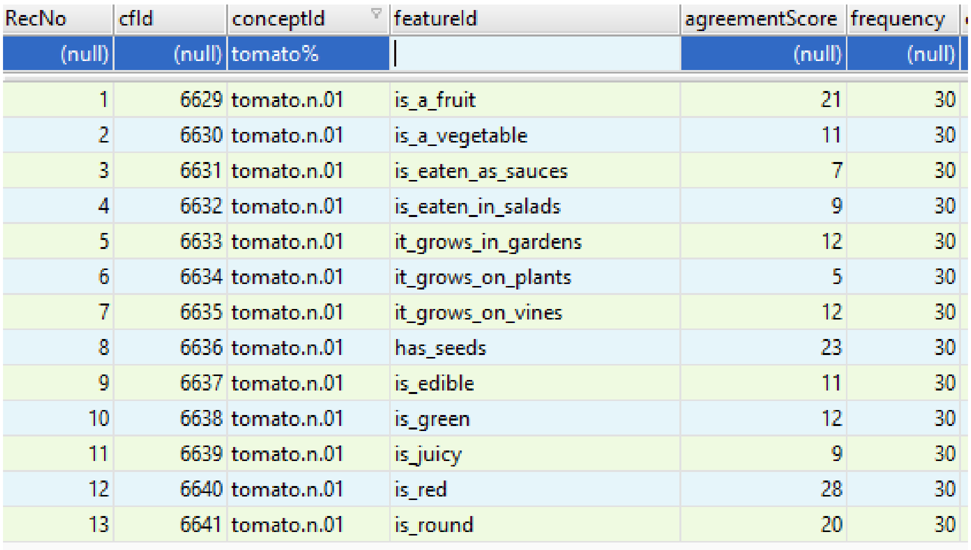
RecNo	cfId	conceptId	featureId	agreementScore	frequency
(null)	(null)	jet%		(null)	(null)
1	3577	jet.n.01	is_an_airplane	13	30
2	3578	jet.n.01	it_flies	19	30
3	3579	jet.n.01	has_an_engine	9	30
4	3580	jet.n.01	has_engines	9	30
5	3581	jet.n.01	has_wheels	5	30
6	3582	jet.n.01	has_wings	14	30
7	3583	jet.n.01	is_fast	24	30
8	3584	jet.n.01	is_large	11	30
9	3585	jet.n.01	is_loud	15	30
10	3586	jet.n.01	is_made_of_metal	5	30
11	3587	jet.n.01	it_requires_fuel	9	30
12	3588	jet.n.01	it_requires_pilots	7	30
13	3589	jet.n.01	is_used_for_passengers	18	30
14	3590	jet.n.01	is_used_for_transportation	10	30
15	3591	jet.n.01	is_used_for_travel	8	30



RecNo	cfId	conceptId	featureId	agreementScore	frequency
(null)	(null)	tomato%		(null)	(null)
1	6629	tomato.n.01	is_a_fruit	21	30
2	6630	tomato.n.01	is_a_vegetable	11	30
3	6631	tomato.n.01	is_eaten_as_sauces	7	30
4	6632	tomato.n.01	is_eaten_in_salads	9	30
5	6633	tomato.n.01	it_grows_in_gardens	12	30
6	6634	tomato.n.01	it_grows_on_plants	5	30
7	6635	tomato.n.01	it_grows_on_vines	12	30
8	6636	tomato.n.01	has_seeds	23	30
9	6637	tomato.n.01	is_edible	11	30
10	6638	tomato.n.01	is_green	12	30
11	6639	tomato.n.01	is_juicy	9	30
12	6640	tomato.n.01	is_red	28	30
13	6641	tomato.n.01	is_round	20	30



RecNo	cfId	conceptId	featureId	agreementScore	frequency
(null)	(null)	sword%		(null)	(null)
1	6377	sword.n.01	is_a_weapon	20	30
2	6378	sword.n.01	has_a_blade	9	30
3	6379	sword.n.01	has_a_handle	13	30
4	6380	sword.n.01	has_a_long_blade	5	30
5	6381	sword.n.01	has_a_pointed_end	10	30
6	6382	sword.n.01	has_a_sheath	5	30
7	6383	sword.n.01	is_dangerous	7	30
8	6384	sword.n.01	is_long	11	30
9	6385	sword.n.01	is_sharp	18	30
10	6386	sword.n.01	is_shiny	6	30
11	6387	sword.n.01	is_made_of_metal	10	30
12	6388	sword.n.01	is_made_of_steel	6	30
13	6389	sword.n.01	is_a_symbol	8	30
14	6390	sword.n.01	is_used_by_knights	6	30
15	6391	sword.n.01	is_used_for_cutting	5	30
16	6392	sword.n.01	is_used_for_war	5	30



RecNo	cfId	conceptId	featureId	agreementScore	frequency
(null)	(null)	tomato%		(null)	(null)
1	6629	tomato.n.01	is_a_fruit	21	30
2	6630	tomato.n.01	is_a_vegetable	11	30
3	6631	tomato.n.01	is_eaten_as_sauces	7	30
4	6632	tomato.n.01	is_eaten_in_salads	9	30
5	6633	tomato.n.01	it_grows_in_gardens	12	30
6	6634	tomato.n.01	it_grows_on_plants	5	30
7	6635	tomato.n.01	it_grows_on_vines	12	30
8	6636	tomato.n.01	has_seeds	23	30
9	6637	tomato.n.01	is_edible	11	30
10	6638	tomato.n.01	is_green	12	30
11	6639	tomato.n.01	is_juicy	9	30
12	6640	tomato.n.01	is_red	28	30
13	6641	tomato.n.01	is_round	20	30

Initial feature values stored for concepts *bullet*, *butterfly*, *elephant*, *grape*, *jet*, *penguin*, *sword*, *tomato*. Concepts *arrow* and *parrot* were not yet in the database.

Bibliography

[1] N. Ayache. Medical computer vision, virtual reality and robotics, 1995. ISSN 02628856.

[2] M. Baroni, B. Murphy, E. Barbu, and M. Poesio. Strudel: a corpus-based semantic model based on properties and types. *Cognitive science*, 34(2):222–54, Mar. 2010. ISSN 1551-6709. doi: 10.1111/j.1551-6709.2009.01068.x. URL http://www.ncbi.nlm.nih.gov/pubmed/21564211.

[3] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. "O'Reilly Media, Inc.", 2009. ISBN 0596555717. URL https://books.google.com/books?id=KGIbfiiP1i4C&pgis=1.

[4] Chih-Wei Hsu, Chih-Chung Chang and C.-J. Lin. A Practical Guide to Support Vector Classification. *BJU international*, 101(1):1396–400, 2008. ISSN 1464-410X. URL http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

[5] D. W. Corne. Lecture 9 - Top ten data mining methods, Neural networks, Overfitting and SVMs. In *Data Mining and Machine Learning*. Heriot-Watt University. URL http://www.macs.hw.ac.uk/~dwcorne/Teaching/DMML/DMMLT10_OF_NN_SVM.ppt.

[6] G. S. Cree and K. McRae. Analyzing the factors underlying the structure and computation of the meaning of chipmunk, cherry, chisel, cheese, and cello (and many other such concrete nouns). *Journal of experimental psychology. General*, 132(2): 163–201, June 2003. ISSN 0096-3445. URL http://www.ncbi.nlm.nih.gov/pubmed/12825636.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-fei. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5206848.

[8] G. Diesendruck and P. Bloom. How Specific is the Shape Bias? *Child Development*, 74(1):168–178, Feb. 2003. ISSN 0009-3920. doi: 10.1111/1467-8624.00528. URL http://doi.wiley.com/10.1111/1467-8624.00528.

[9] A. Eguchi. Object recognition based on shape and function: inspired by children's word acquisition. *Inquiry Journal of Undergraduate Research*, 13: 38–49, 2013. URL http://dora.uark.edu/fedora/repository/uark%3A2144/OBJ/Inquiry201209.pdf#page=38.

[10] A. Eguchi, H. Nguyen, and C. W. Thompson. Everything is alive: towards the future wisdom Web of things. *World Wide Web*, 16(4):357–378, Aug. 2012. ISSN 1386-145X. doi: 10.1007/s11280-012-0182-4. URL http://link.springer.com/10.1007/s11280-012-0182-4.

[11] P. D. Eimas and P. C. Quinn. Studies on the Formation of Perceptually Based Basic-Level Categories in Young Infants. *Child Development*, 65(3):903–917, June 1994. ISSN 0009-3920. doi: 10.1111/j.1467-8624.1994.tb00792.x. URL http://doi.wiley.com/10.1111/j.1467-8624.1994.tb00792.x.

[12] Encyclopedia Brittanica. Taxonomy. URL http://www.britannica.com/EBchecked/topic/584695/taxonomy/48704/A-classification-of-living-organisms.

[13] J. D. Eno and C. Thompson. Virtual and Real-World Ontology Services. *IEEE Internet Computing*, 15(5):46–52, Sept. 2011. ISSN 1089-7801. doi: 10.1109/MIC.2011.75. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5871569.

[14] P. Gärdenfors. *Conceptual Spaces: The Geometry of Thought.* MIT Press, 2004. ISBN 0262572192. URL http://books.google.co.uk/books/about/Conceptual_Spaces.html?id=FSLFjw1EcBwC&pgis=1.

[15] H. Grabner, J. Gall, and L. Van Gool. What makes a chair a chair? In *Computer Vision and Pattern Recognition*, pages 1529–1536. IEEE, 2011. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5995327.

[16] A. Johnson, Y. C. Y. Cheng, and L. Matthies. Machine vision for autonomous small body navigation. *2000 IEEE Aerospace Conference. Proceedings (Cat. No.00TH8484)*, 7, 2000. ISSN 1095-323X.

[17] D. G. Kemler Nelson, R. Russell, N. Duke, and K. Jones. Two-Year-Olds Will Name Artifacts by Their Functions. *Child Development*, 71(5):1271–1288, Sept. 2000. ISSN

0009-3920. doi: 10.1111/1467-8624.00228. URL `http://doi.wiley.com/10.1111/1467-8624.00228`.

[18] T. Kollar, J. Krishnamurthy, and G. Strimel. Toward Interactive Grounded Language Acquisition. *Robotics: Science and Systems*, 2013.

[19] L. Fei-Fei, R. Fergus, and A. Torralba and R. F. L. Fei-Fei. *Recognizing and Learning Object Categories, CVPR 2007 short course.* URL `http://people.csail.mit.edu/torralba/shortCourseRLOC/index.html`.

[20] B. Landau, L. B. Smith, and S. S. Jones. The importance of shape in early lexical learning. *Cognitive Development*, 3(3):299–321, July 1988. ISSN 08852014. doi: 10.1016/0885-2014(88)90014-7. URL `http://linkinghub.elsevier.com/retrieve/pii/0885201488900147`.

[21] L.-l. ling Wu and L. W. Barsalou. Perceptual simulation in conceptual combination: Evidence from property generation. *Acta psychologica*, 132(2):173–189, Oct. 2009. ISSN 00016918. doi: 10.1016/j.actpsy.2009.02.002. URL `http://www.ncbi.nlm.nih.gov/pubmed/19298949`.

[22] M. Matteucci. A tutorial on clustering algorithms: k-means clustering. URL `http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html`.

[23] K. McRae, G. S. Cree, M. S. Seidenberg, and C. Mcnorgan. Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods*, 37(4):547–559, Nov. 2005. ISSN 1554-351X. doi: 10.3758/BF03192726. URL `http://www.springerlink.com/index/10.3758/BF03192726`.

[24] P. Meer. Are we making real progress in computer vision today? *Image and Vision Computing*, 30(8):472–473, Aug. 2012. ISSN 02628856. doi: 10.1016/j.imavis.2011.10.004. URL `http://linkinghub.elsevier.com/retrieve/pii/S0262885612000662`.

[25] A. Mojsilović. A computational model for color naming and describing color composition of images. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 14(5):690–9, May 2005. ISSN 1057-7149. URL `http://www.ncbi.nlm.nih.gov/pubmed/15887562`.

[26] J. C. Niebles. A Hierarchical Model of Shape and Appearance for Human Action Classification. In *2007 IEEE Conference on Computer Vision and Pattern Recog-*

nition, pages 1–8. IEEE, June 2007. ISBN 1-4244-1179-3. doi: 10.1109/CVPR.2007.383132. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4270157.

[27] I. Niles and A. Pease. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems - FOIS '01*, volume 2001, pages 2–9, New York, New York, USA, Oct. 2001. ACM Press. ISBN 1581133774. doi: 10.1145/505168.505170. URL http://dl.acm.org/citation.cfm?id=505168.505170.

[28] A. Pease. The Suggested Upper Merged Ontology (SUMO), 2000. URL http://www.adampease.org/OP/.

[29] Princeton University. WordNet. URL https://wordnet.princeton.edu/.

[30] V. Rieser and O. Lemon. *Reinforcement learning for adaptive dialogue systems.* Springer, Berlin, Heidelberg, 2011. ISBN 978-3-642-24941-9. doi: 10.1007/978-3-642-24942-6. URL http://link.springer.com/10.1007/978-3-642-24942-6.

[31] D. Roy. Grounding words in perception and action: computational insights. *Trends in cognitive sciences*, 9(8):389–396, Aug. 2005. ISSN 1364-6613. doi: 10.1016/j.tics.2005.06.013. URL http://www.ncbi.nlm.nih.gov/pubmed/16006171http://www.sciencedirect.com/science/article/pii/S1364661305001853.

[32] J. Saffran, R. Aslin, and E. L. Newport. Statistical learning by 8-month-old infants. *Science*, 1996. URL http://www.sciencemag.org/content/274/5294/1926.short.

[33] K. Sage and S. Young. Security applications of computer vision. *IEEE Aerospace and Electronic Systems Magazine*, 14(4):19–24, 1999.

[34] C. Silberer and M. Lapata. Learning Grounded Meaning Representations with Autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 721–732. Association for Computational Linguistics, 2014. URL https://www.cs.utexas.edu/users/ml/clamp/silberer_lapata_acl2014.pdf.

[35] R. Szeliski. Computer Vision : Algorithms and Applications. *Computer*, 5:832, 2010. ISSN 10636919. doi: 10.1007/978-1-84882-935-0. URL

http://research.microsoft.com/en-us/um/people/szeliski/book/drafts/szelski_20080330am_draft.pdf.

[36] T. Leung and J. Malik, T. L. Malik, and J. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29 – 44, 2001. doi: 10.1023/A:1011126920638.

[37] A. M. Turing. Computing Machinery and Intelligence. *Mind*, LIX(236):433–460, 1950. ISSN 0026-4423. doi: 10.1093/mind/LIX.236.433. URL http://mind.oxfordjournals.org/cgi/doi/10.1093/mind/LIX.236.433.

[38] Vidal-Naquet and Ullman. Object recognition with informative features and linear classification. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 281–288 vol.1. IEEE, 2003. ISBN 0-7695-1950-4. doi: 10.1109/ICCV.2003.1238356. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1238356.