# Abstract

Estimating the size of an object is a task that humans can do but struggle to explain how, it is common sense to be able to estimate the size of an object. This dissertation project aims to create a database of objects with their commonly found sizes to assist software in being able to make these same estimates. The project will achieve this by using information extraction techniques on the internet to find written examples of objects sizes.

To date the I have built basic models for extracting this information and have created tools to help with labelling training data to be used with semi-supervised learning to improve the accuracy of the models.

# Introduction

Understanding an objects size could help machine learning models in image and video recognition by allowing estimation of an unknown objects size by comparing it with other objects in the image. For example, if you had to classify an object in an image as either a dog or a horse knowing that it was stood next to a person, and that on average a person is larger than a dog but smaller than a horse, this could inform your classification.

This dissertation takes on the problem of determining the general sizes of different objects. Humans are talented at estimating sizes of objects based on common sense or memory. As mentioned above this can help us make estimations about new objects, can help us to judge the distance of an object, or can -------------

Named entity recognition techniques have been shown to produce accurate results, as have relationship extraction models. Combining this with gazetteers for objects or units of size is likely to give good results. The limitation to these models will be the quality and quantity of the training data. This is a trade-off, increasing the quality of our data takes more time and therefore reduces the quantity we can gather. Therefore, this project will go with a semi-supervised learning technique that tries to strike a good balance between the two.

## Aims and Objectives

The aim of this dissertation is to make progress towards creating a database containing information about objects and their usual sizes. This can be broken down into three stages. Stage one is fulfilling our requirement of training data containing various objects and sizes. The aim is to train machine learning models to be able to identify objects and sizes within a sentence and determine if they are related. To collect enough data to adequately train these models we will use semi-supervised learning, which means that this stage will run simultaneously alongside stage two.

Stage two is building the named entity recognition and relationship extraction models for both identifying objects and sizes in text and determining if they are related. To help with the semi-supervised learning we can build some very basic models to start collecting data. Using regular expressions to determine sizes, and part-of-speech tagging to determine nouns, we can build models with poor accuracy but that will help in collecting data that can be refined into the final training set.

The final stage of the project will be to collect all the results into a database. The accuracy of the collected data can be improved if objects have been found multiple times. We can look at previously found sizes of the object to determine if this new measurement is accurate. If we introduce an object hierarchy to determine if objects are related, then we can also use similar objects to estimate realistic sizes.

## Overview of the Report

This report will begin with a literature survey.

This will be followed by an in-depth investigation into the requirements of the project and an analysis of the problem and how the project will be tackling it.

This will be followed by an update on progress made so far.

Finally, there will be a conclusion and detailed project plan.

# Literature Survey

## Information Extraction

There has not been much previous work that has been done on this topic, but each of the individual aspects of the project have been researched in depth by other parties. Given that this is an information extraction task, this literature survey will give a background on different techniques that have been used successfully in that past. The information extraction task is broken down into named entity recognition and relationship extraction.

### Named Entity Recognition

Named entity recognition is the process of identifying entities in text. This task can be broken down into finding the entities and classifying them into classes. The most common classes used for named entities are person names, organisations, and locations but you can build models to identify any class of entity. There are many challenges that come along with this such as coreference which is when an entity is referred to in the text but not by name. For example, “Henry Ford was born in 1863. He is known for founding the Ford Motor Company.”. In the second sentence the entity “Henry Ford” is referred to as “He”. This is called coreference and needs to be resolved before entities can be identified if you want to capture all instances.

Three primary approaches to named entity recognition are knowledge-engineering, supervised learning, and semi-supervised learning \cite{text\_processing\_lecture\_5}:

#### Knowledge-Engineering

Knowledge-engineering approaches use gazetteers and human written rules to determine if a token is an entity. Its strengths are its high performance (on its specific domain) and its transparency. However, its weaknesses are that it requires domain experts to write the rules, changing to another domain requires possibly rewriting all the rules, and you need domain specific gazetteers.

#### Supervised Learning

Supervised learning is a technique that attempts to fix the generalisation problem in knowledge-engineering approaches. Supervised learning systems learn from labelled examples, moving to a new domain only requires a new set of labelled examples. These labelled examples include features which inform the model as to the context in which the token was found. Features for a named entity recognition model would usually include information about the token, previous and future tokens, their part of speech tag, and any other information that might be useful in identifying a specific class of entity. There are a variety of different models that can be built using supervised learning such as decision trees, support vector machines, and neural networks. The advantages of this approach are that the model is easier to generalise towards different domains, depending on your problem the level of expertise to label the data is usually less than would be required to write rules, and you don’t need any domain specific additional information such as gazetteers (Although these can help improve accuracy). Issues with this approach are usually that you need a large amount of annotated data for accurate results. Manually labelling that amount of data can take many hours, and in domains where the labelling might be subjective you would need multiple labellers to ensure high accuracy in your training data.

#### Semi-Supervised Learning

Semi-supervised learning is a similar approach to supervised, but the advantage is that you don’t need as many labelled examples in your training data. You have a small amount of labelled data as part of a larger unlabelled training data set. Using the labelled examples and by looking at the structure of the unlabelled data you attempt to form a model, you are relying on assumptions that either points close to one another share a class, points within a cluster share the same class, or the classes can be inferred by patterns hidden on a lower dimension than the input space (Chapelle, Olivier; Schölkopf, Bernhard; Zien, Alexander (2006). *Semi-supervised learning*. Cambridge, Mass.: MIT Press. ISBN [978-0-262-03358-9](https://en.wikipedia.org/wiki/Special:BookSources/978-0-262-03358-9)).

### Relationship Extraction

Relationship extraction is the process of determining if 2 or more entities in a text are related. Due to needing to know which tokens are entities this process can usually only be performed after named entity recognition. An example of this relevant to the project would be the sentence: “The average apple is 7cm in diameter”. The relationship extraction task is to determine if the two entities, in this case the object “apple” and the size “7cm”, are related. There are techniques to tackle this problem with good results but there are some key challenges. Relationships over multiple sentences are much harder to detect and usually systems will ignore these relationships and only attempt to extract ones within a sentence. Semantic drift is another challenge, it describes the issue that arises when you use a small set of labelled training data for your model. If the model attempts to learn its own rules to generate more training data, an incorrect rule will generate more incorrect examples which will generate more incorrect rules and repeat. This causes an exponential drift away from the initial accuracy of the labelled examples. Ways to combat this include human intervention, in the first few iterations where there are still relatively few examples, a human can manually check the results to see if they are accurate. However, this is not a perfect solution and semantic drift can be a significant factor in decreasing your model’s accuracy.

Four primary approaches to relationship extraction are knowledge-engineering, supervised learning, bootstrapping and distant supervision \cite{text\_processing\_lecture\_6}:

#### Knowledge-Engineering

Knowledge-Engineering approaches follow rules for identifying relationships. They can be split into two different categories: shallow, and deep. Shallow systems use pattern-action rules to determine if a relationship exists. For example:

Pattern: “$Person, $Position of $Organisation”

Action: add-relation(is-employed-by($Person, $Organisation))

Then the sentence “Mr. Wright, executive vice president of Merrill Lynch Canada” would match the pattern and the system would determine that “Mr. Wright” has a relationship to “Merrill Lynch Canada” of type “is-employed-by”.

Deep systems use language rules to determine relationships. The means looking at examples and determining the grammatical relationships between the entities. For example, a subject, person, and an object, organisation, separated by a verb such as “works for” would indicate an is-employed-by relationship.

Advantages of this approach are that it has high precision and is transparent, i.e. a human can read the rules to understand why a relationship has been classified in the way that it was. However, the disadvantages usually outweigh this as it is impossible to write all rules to capture all instances and the approach would need new rules to be written for every different domain.

#### Supervised Learning

Supervised learning for relationship extraction is similar to that for named entity recognition, covered in section XXXXX. The differences come when choosing the features for the training data. Due to already have the entities identified the features would include these classifications. They would also include the tokens between the two entities as this could be informative in classifying their relationship.

To reiterate, the strengths of this approach are its adaptability to new domains and the no requirement for writing complex rule sets. Disadvantages are that the model performance greatly relies on the quality and quantity of training data and the degree of which this data represents the real problem.

#### Bootstrapping

Bootstrapping can be thought of as a self-teaching model that starts by seeding it with a few examples. Given either some pattern examples or some relationship examples the model parse the training text to find relationships that fit these patterns or patterns that fit the relationships given. From here it will build new patterns from relationships found or build new relationships from patterns found. It can do this iteratively until the rule set is deemed large enough.

The strengths of this approach are that it requires no labelled data, just a few examples, which it can very quickly generate more from. The disadvantage to this is semantic drift, if the model incorrectly matches two entities to a relationship it will generate a rule from this pairing. This rule will then allow the model to discover more incorrect relationships. Every iteration the number of incorrect rules will exponentially increase decreasing the accuracy of the model.

#### Distant Supervision

The aim of this approach is to reduce the need for labelled examples. You can think of this approach of being a mixture between bootstrapping and supervised learning. You perform one iteration of bootstrapping on a large set of training text and use this to build your supervised learning model.

The advantages of this is the reduced need for labelled training data and the speed at which the model learns new rules. The accuracy of these models is worse than that of supervised models (or very narrow domain knowledge-engineering models) but it requires much less data labelling.

## Tools and Libraries

### Programming Languages

Python has recently become the go to language for data science and machine learning. Due to this there are a lot of libraries that have been built to assist programmers in these two domains. There are libraries for data manipulation and storage (numpy and pandas), libraries for natural language processing (nltk), and libraries for building machine learning models (sci-kit learn). Python also has the advantage of being easy to write and understand due to its near simplified syntax.

However, due to Python being a relatively new language there are some advantages to using languages that have been around for longer. Another language that has some natural language processing roots is Java. One of the best natural language processing toolkits is StandfordNLP, a library for Java. Many academic projects investigating natural language processing techniques have been written with the assistance of this library and it has been proven to be a very powerful tool. Another advantage Java has over Python is that Java is a compiled language whereas Python is interpreted. This means that the run time of the Java system will be significantly faster once it has been compiled.

C++

## Datasets

One of the requirements for this project will be a large textual dataset that contains mentions of objects and their sizes. Other datasets that would be useful to the project would be any labelled sets that contain information on either objects, sizes, or both.

## Databases

<https://www.scylladb.com/resources/nosql-vs-sql/?utm_source=google&utm_medium=cpc&utm_campaign=NoSQL&utm_content=s_1&utm_medium=cpc&utm_source=google&utm_campaign=8006013281&utm_placement=&gclid=CjwKCAiAuK3vBRBOEiwA1IMhupr37kDvHBBDlnrMW8xO_IMYOXZp-a_v523sfDyPgeiXwsFSl7lb9RoC-3MQAvD_BwE>

There are two main different types of databases, relational databases (SQL) and non-relational (NoSQL). SQL databases are used in situations where one items relationship to another is important. They follow strict standards to ensure low data redundancy and high reliability. However, if the data you need to store does not require relationships between items then a NoSQL database is often faster and more adaptable.

# Requirements and Analysis

The aims and objectives of this project (in chronological order) are as follows:

1. Build a data labelling tool
2. Find a suitable text data source
3. Build an NER model using POS, Wordnet, and regular expressions
4. Label data found with NER model 1
5. Build an NER model to identify objects and sizes in text using machine learning and data found from NER model 1
6. Build a RE model to determine if an object and size in text are related using machine learning
7. Build a feedback loop tool
8. Correct NER model 2 and RE model 1 classifications and feedback into training data
9. Create database to store data
10. Use previous found instances of same objects to inform decision
11. Use object hierarchy to inform decision

Objective 9 marks the end goal of the project from a definition standpoint. Reaching this objective would mean that the project will have implemented the minimum requirements. The models might not be accurate, and the data might be poor, but a database of objects and their relative sizes will have been built using machine learning. Objectives 10 and 11 are additional steps that could help improve the accuracy of the model. These go alongside objectives 5 and 6, a model can quickly be built as a proof of concept initially, however improving the features used and the quality and quantity of the data can drastically change the outcome of the project.

Objective 5 has a caveat, building a machine learning model to identify sizes might not be time efficient. Depending on the results of the regular expressions in objective 3 it might be more effective to stick with this approach and focus on the more difficult task of identifying objects. This is due to the simple nature of sizes, all sizes contain a unit, of which there are a limited number, and all sizes all contain a numeric value. Both things that are easily picked up by a regular expression.

There is also an extension to this project that would be to include sizes of objects relative to others. For example, “That tree is as tall as a house.”, if you know the size of a house from previous text then you can infer that some trees are equal to this height.

The evaluation of the project’s success will be difficult to measure. Testing the machine learning models using the labelled training data is an easy enough task, however depending on how the training data is gathered might skew our models. This means that although it might perform well on the test set, we need to ensure that the test set is an accurate representation of the real text we’re trying to extract information from. We can limit the gap by trying to include examples of all classifications. Sentences with just objects, just sizes, both objects and sizes, sentences where the object and size are related and sentences where they are not. The means that the model will have an accurate representation of all different possibilities.

## Machine Learning Models

Semi-supervised learning is the technique that this project will employ in order to train models. I have researched various approaches to try to ensure that the data remains as clean as possible. Approaches found in \cite{reference3} such as …

Choosing the features for information extraction to inform the model is a difficult task. Tools can be used to analyse training data to determine which features are providing the most information. Features with high correlation can usually be reduced which helps to improve model size, run time, and accuracy. A starting set of features can be found in Speech and Language Processing by \cite{reference1} pp. 4.

### Named Entity Recognition

Other approaches for information extraction such as knowledge-engineering could be a good fit for certain aspects of this project such as recognising sizes. Knowledge engineering approaches as laid out in \cite{reference2} usually approach the problem by using domain experts to write rules to capture the information. Rules for objects such as sizes could be as follows:

* Must contain a unit of size
  + Metres, metre
  + m
  + Kilometre, centimetre, millimetre, micrometre
  + km, cm, mm, µm
  + Feet, foot
  + ft, ‘
  + Inches
  + Inch, “
* Must contain a numeric value
  + 1
  + One
  + 1.2
  + One and a half
* Could include a range of sizes
  + Between 2 and 3
  + 2 – 3
  + 2 to 3

## Databases

The database for this project will be relatively simple and can be created using SQL. Database design guidelines laid out in \cite{reference4} explain the importance of normalising the tables to ensure that information is not stored multiple times. In the case of this project all we need to store is a reference to an object, and a list of found sizes which is why we prefer this method over a NoSQL approach.

This is the database to store the results of our information extraction. We must also use a database to store the training data for the machine learning models. These will not need a relational database structure as all information can be contained in one row but due to the results being stored in an SQL database it makes sense for all data to be stored in the same place.

# Progress

So far, I have created a basic machine learning model for named entity recognition. In order to help label training data for the models I have created a tool that allows me to quickly tag tokens in sentences as either objects or sizes. This also forms the base of the feedback tool I have created.

To assist with the initial capturing of data and to help the basic NER model I have researched into ways to make it more accurate without any initial training data. I have incorporated a model from the Python library NLTK that identifies part of speech tags for tokens. If the POS tag for a token identifies it as a noun, then the tool will flag this for human verification. Another way to assist the model is to use a gazetteer (list of known objects) such as Wordnet. If a token appears in Wordnet as an object, then it will also be flagged for human verification. Trying to identify sizes will initially be done with regular expressions as they’ll be accurate enough to provide training data without the need to have to implement a model.

I have also started looking into potential data sources to extract our information from. Currently I am looking to English Wikipedia as a potential source.

# Conclusions and Project Plan

The plan of work is to first use a basic model, part of speech tagging, Wordnet and regular expressions to extract potential training data from some text source e.g. English Wikipedia. These examples will then be hand labelled to start building a training data set. They will be used to create a more sophisticated model that at first will use the features laid out in Speech and Language Processing by \cite{reference1}. A feedback tool will be implemented here for to more efficiently train the model. This tool will allow me to label found tokens are either correct or incorrect. If the token was misidentified, then the tool will also allow me to correct it. This will give a base level from which the rest of the project can be built upon. Reaching this stage will be critical as the feasibility of the project can start being assessed more closely. Additional investigation and research may be required if the model isn’t performing as expected.

Additionally, more investigation will be done into proper techniques for relationship extraction as well as building a database model for storing the collected results. From here we can start to think about other ways to increase the model’s accuracy by checking for previous records of objects and possibly estimating size based on object hierarchy.

These stages for the most part can all happen simultaneously, other than needing training data to build the second set of models, the stages will progress alongside one another. This is shown in figure 5.1.