Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Querying Knowledge Graphs for Recommendations

Daanish Millwalla

August 19, 2022

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Master of Science in Computer Science (Future Networked
Systems)

Declaration

I hereby declare that this dissertation is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write.

Signed: ______________________ Date: ______________________

Abstract

Traditional recommender systems have used statistical and machine learning models to predict and curate content for users based on their past behaviours and the trends of a larger population. This method involves continuous monitoring and tracking of user activity to train and refine a model. Despite the obvious limitations of needing vast amounts of data to train and test the model, this approach also poses a risk of being invasive to the user.

As an alternative, embracing Tim Berners-Lee's vision of linked data and the semantic web, knowledge graphs can be used to semantically model data that already exists on the world wide web, publicly. The semantic links with entities and their relationships can then be queried and matched with user preferences. Graph databases provide flexibility with schema for the data while providing a query interface that is able to search through semantic annotations within the data.

This dissertation provides an approach to a recommendation system that utilizes knowledge graph as the source of recommendations queried using the preferences of a user. It aims to avoid monitoring user behavior to be secure by design and utilise knowledge graphs to provide deterministic recommendations based on semantic links within the graph.

Acknowledgements

I would like to thank Dr. Declan O'Sullivan for his constant support and guidance in this endeavour. I'm grateful to Trinity College Dublin for accepting me in their esteemed master's program, my classmates and the faculty for the continued support.

I would like to thank my parents, my sister and my aunt for all their support. This journey would not have been possible without them. Lastly, I thank all my friends for their continued encouragement to pursue my dreams.

Contents

List of Figures

List of Tables

Nomenclature

CF	Collaborative Filtering
KG	Knowledge Graph
ML	Machine Learning
NLP	Natural Language Processing
RDF	Resource Description Framework
HTTP	Hyper Text Transfer Protocol

1 Introduction

1.1 Context

Recommender Systems are a mainstay in most content platforms these days. The vast amounts of content and the varied genres available make it difficult for a user to filter and choose what to consume. Content and search platforms collect enormous amounts of data to train statistical models that are able to solve this problem of generating recommendations. One of the most popular models are of the Collaborative Filtering kind that relies on identifying trends and patterns in a large populace to suggest and predict recommendations for individual users.(5)

Underlying these models are vast amounts of data collected about user's behavior and actions on a continual basis.(6) The model is trained and refined continuously as more behavior data is collected and fed in. This raises questions regarding the privacy of the user and increases the onus on security measures to put in place. Data needs to be anonymised and secured against leaks. Care must be taken to ensure even anonymised data cannot be deanonymised combining with other publicly available information.(7)

Another problem commonly encountered is the 'cold start' problem.(8) Without enough seed data to feed into the model, predictions are not of high quality. The system requires usage over time and gather enough usage data to then process into accurate predictions for users. Mitigating this issue by collecting more data raises another problem with scaling the system to handle the large volumes of data and process it in memory.(6)

Improvements in storage capacity and the advent of cloud computing has made sure the technical challenges associated with this has been addressed. Economies of scale have ensured that large scale, production quality systems can be designed to support this. However, the issues with privacy still remain.

A novel approach to providing recommendations may lie in the usage of knowledge graphs. Knowledge Graphs represent data in the form of subject-predicate-object(9), thereby allowing a flexible structure to data that is largely unstructured. Organisation of data in this manner allows it to be reasoned over and linked to other graphs and linked data. (10)

Knowledge Graphs can be used to instead suggest recommendations based on common nodes and predicates between the user's preferences and the representation of content and metadata in graph format. This would allow the recommendations to be idempotent in nature and the behaviour of the system capable of being reasoned about. A simple query can be in place to match nodes representing preferences of the user with nodes or predicates in the graph. This removes the need of monitoring user behavior since the recommendation system already has enough data to match content for the user with, thereby eliminating the privacy risks introduced by data collection. By utilising existing and curated knowledge graphs publicly available, there would be enough data present to make recommendations, thereby dealing with the problem of cold starts.

1.2 Motivation

The project explores the possibility and feasibility of using knowledge graphs as the main driver of recommendations. The usage of knowledge graphs in recommender system is usually limited to being an input to statistical models that are trained to explore the network within the graph to suggest recommendations. The approach proposed in this dissertation differs in the sense that graph nodes and predicate matching, instead, is used to suggest predictions.

Recent issues of user data deanonymisation (7) raises important issues about the means of refining recommender systems that use collaborative filtering. Besides reducing the risk of privacy and security breaches by design, semantic modelling through knowledge graphs also allows richer and more personalised links to be explored within the data given the graph includes accurate representation of real world semantics.

1.3 Solution Outline

Collaborative Filtering models involve extensive amounts of user data collection as well as large amounts of storage and processing. Knowledge Graphs have the potential in flexibly representing natural language and semantics into data that can be queried. There also exist off-the-shelf ML models that can interpret natural language and represent information within it in a knowledge graph. If adequate data about the context can be gathered and fed through the model, a knowledge graph that accurately represents the world can be created. This can become the source of knowledge to query for recommendations with appropriate matches between data about the user and nodes within the knowledge graph.

1.4 Approach

The system would need to be designed to achieve the following goals:

1. Allow user scrutability of its model
2. Avoid being intrusive to user behaviour as much as possible
3. Deterministic with predictions
4. Storage efficient
5. Ability to learn through unstructured data

To be scrutable to the user, it is necessary to provide the user the chance to view their model as assumed by the system as well as the ability to modify it. An interface can be designed that manually records these preferences to create a stereotype for the user. To prototype manually accepting user preferences, a web application can be used.

Using semantic links of a knowledge graph allows matching user preferences with content within the graph and also allows predictions to be deterministic and not based on probabilistic calculations of a model. With the nature of linked data being able to reference multiple data sources, it was possible to reference data that exists outside of the system without importing or pre-processing it first.

Various NLP approaches could be used for the purpose of enriching the knowledge graph. A tokenizer in python such as 'Textacy' could be used to break up parts of speech and form triplets from natural language.(11). While it's performance was robust, it may not capture the finer details of natural language. Machine learning based tools such as Diffbot (12) can capture these implicit links within natural language.

1.5 Design Adopted

The system will have a web-based interface that is able to record the user's preferences and create a simple model of the user's likes. A graph database will host this model. A machine learning model 'Diffbot', that processes natural language into Subject-Verb-Object sections within it will be utilised. These parts of speech can then be made into triplets to be saved inside a graph database. A simple implementation to match preferences with knowledge graph nodes would be to store nodes as literals and perform string matching via the regex clause in the queries.

The use case demonstrated by the system designed in this dissertation is of recommendations about Dublin based on the preferences of a user. The knowledge graph is queried for entities that have some relation with Dublin as well as a relation to an entity that also matches the preference of the user querying it. The system is responsible for collecting the preferences of a user and give recommendations related to Dublin based on those preferences.

Hence. for a given user model such as:

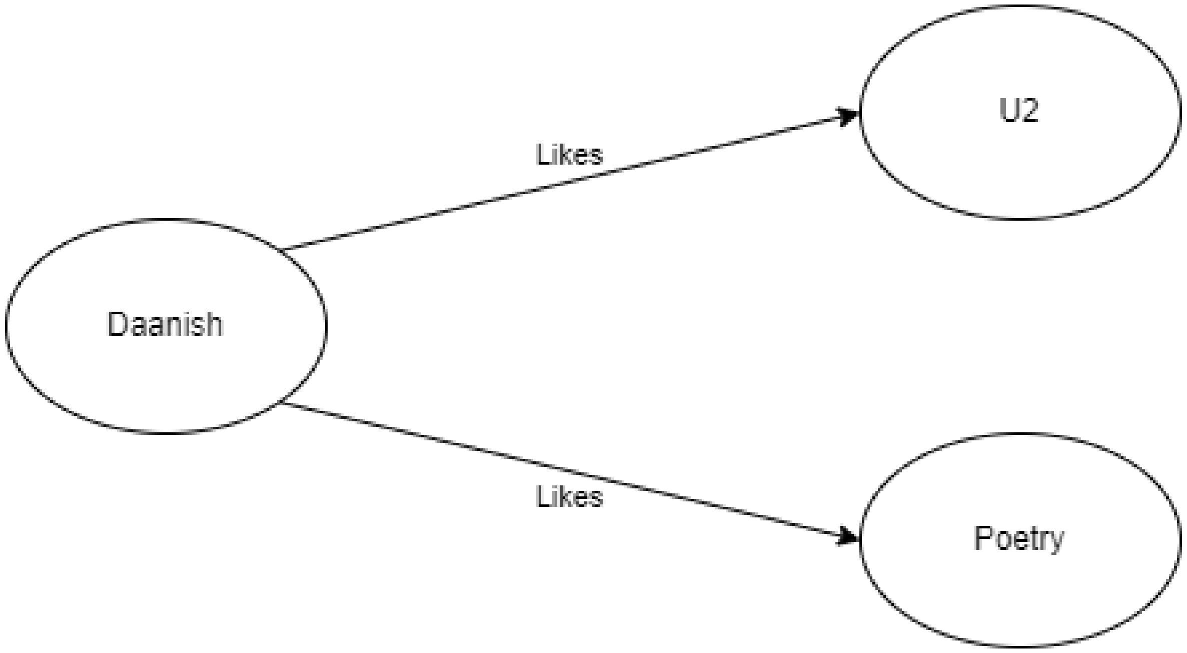


Figure 1.1: User's Model

and a graph database having triplets such as:

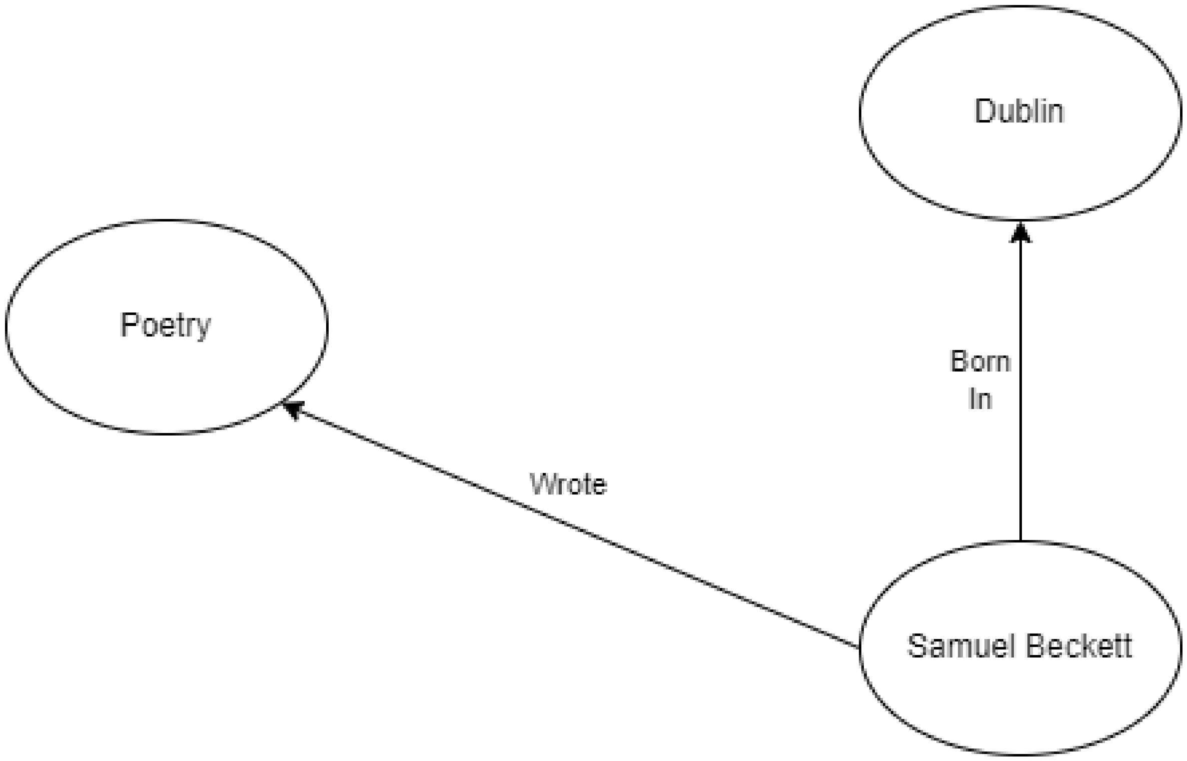


Figure 1.2: External Graph Triplets

The SPARQL query could match the literal nodes 'Poetry' and return 'Samuel Beckett' as a recommendation.

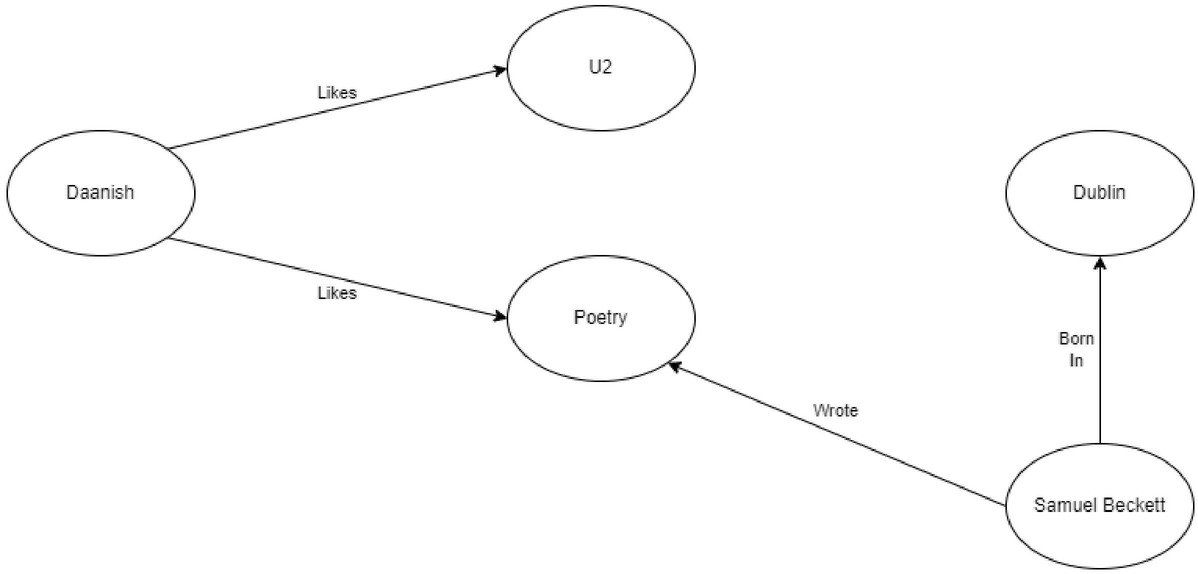


Figure 1.3: External Graph matched with User Preferences

The design choices made in the implementation of the system are further explained in chapter 4.1. These include the various techniques evaluated, the decisions and approach taken keeping in mind the trade-offs associated with multiple approaches.

1.6 Document Structure

This thesis has been organised into 6 chapters. Chapter 1 lays out the idea and motivation behind the problem being addressed as well as an outline of the approach to tackle it. Chapter 2 explores the background of the concepts and techniques used in this thesis. It gives an overview of the evolution and usage of knowledge graphs and Tim Berners-Lee's vision of the Semantic Web and Linked Data. It also explores the traditional approaches to recommendations and some insight into Natural Language Processing(NLP). Chapter 3 explores work that already exists in these domains that inform the decisions made in this thesis. It evaluates the state of the art that exists in the domain of recommender systems and the NewsReader project that brings together NLP with generating Linked Data. Chapter 4 explains the design choices made with the thesis, how the system is build and how it operates. Chapter 5 evaluates the performance of the system and measures its results. Finally, Chapter 6 provides conclusions to the work done in this dissertation and provides reflections and ideas for future work.

2 Background

Certain concepts in this section will be explored to better understand the project setup and the findings derived from it. The concepts also make up certain parts of the overall system proposed in this research.

2.1 Knowledge Graphs

Knowledge Graphs are data structures that represent data in the form of nodes and vertices. Each record has a Subject-Predicate-Object format, with Subjects and Objects being represented by Nodes and Predicates as Vertices. These nodes represent real life entities and the vertices represent the relationship between these entities.(9)

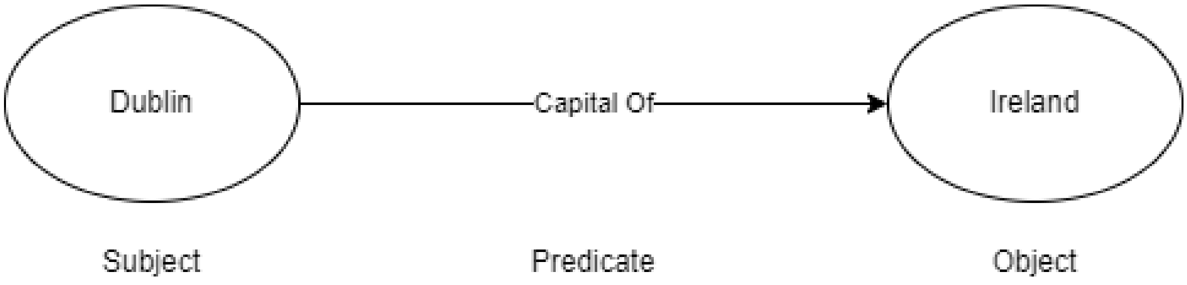


Figure 2.1: Subject-Predicate-Object in a Knowledge Graph

Popularised by the 2012 project 'Google Knowledge Graph'(13), the idea was later embraced by other big tech entities such as Facebook and Microsoft. (14). Knowledge graphs allow capturing complex relationships between real world entities. Data in knowledge graphs do not need to follow a schema as long as they follow the subject-predicate-object pattern, hence providing more flexibility to applications.(15) Query languages over these graphs are powerful means of reasoning over this data which provide not just common relational operations such as joins and unions but also a means to search and discover entities linked over paths of variable length.(16)

With the evolution of semantic web and growth of the linked open data project, Knowledge Graphs provided the underlying mechanism to achieve the goals of these projects. Some popular vendors of Knowledge Graph databases are AllegroGraph(the one used in this project), GraphDB, Neo4j, Virtuoso, etc.

To define the nature and metadata around entities, Ontologies are used with Knowledge Graphs.(9) Ontologies allow reasoning over the inherent relationship within the data and impose a logical schema over it. This extends the capability of the database to not just retrieve explicit data in the form of Subject-Predicate-Object 'Triplets' through querying the data, but also analysing the relationship of the data defined through the ontology.

Machine Learning techniques can also be applied to Knowledge Graphs to unearth proximity of entities as well as hidden relations within the graph that are not explicitly defined.(17) This technique is also known as Knowledge Graph Embedding.(18) The techniques aim to represent nodes and vertices in a knowledge graph in the form of a vector and feed to a machine learning model to discover data that is not explicitly mentioned in the vectors.

2.2 Semantic Web

Semantic Web is the extension of the vision of the web as a 'Web of Documents' by being an interconnected web of data. (19) This enables sharing of data between different data stores. This data can be highly contextualised and hence needs a framework of supporting formats and vocabularies that define its nature, behavior and meaning in an inter-operable manner. This is provided through W3C defined frameworks such as RDF, OWL and SPARQL.

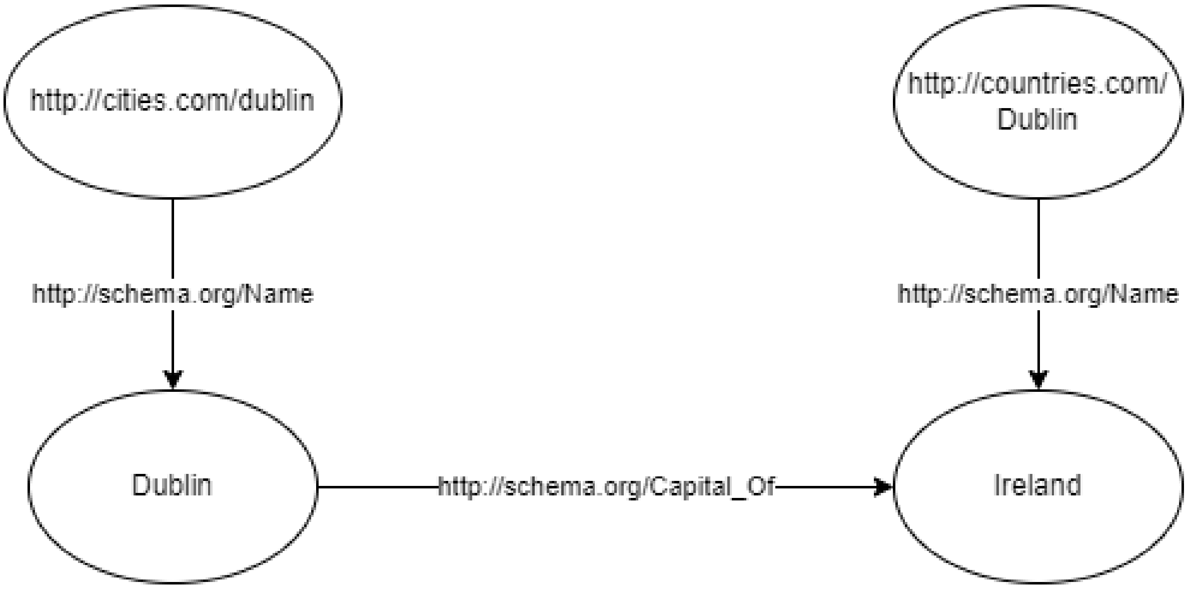


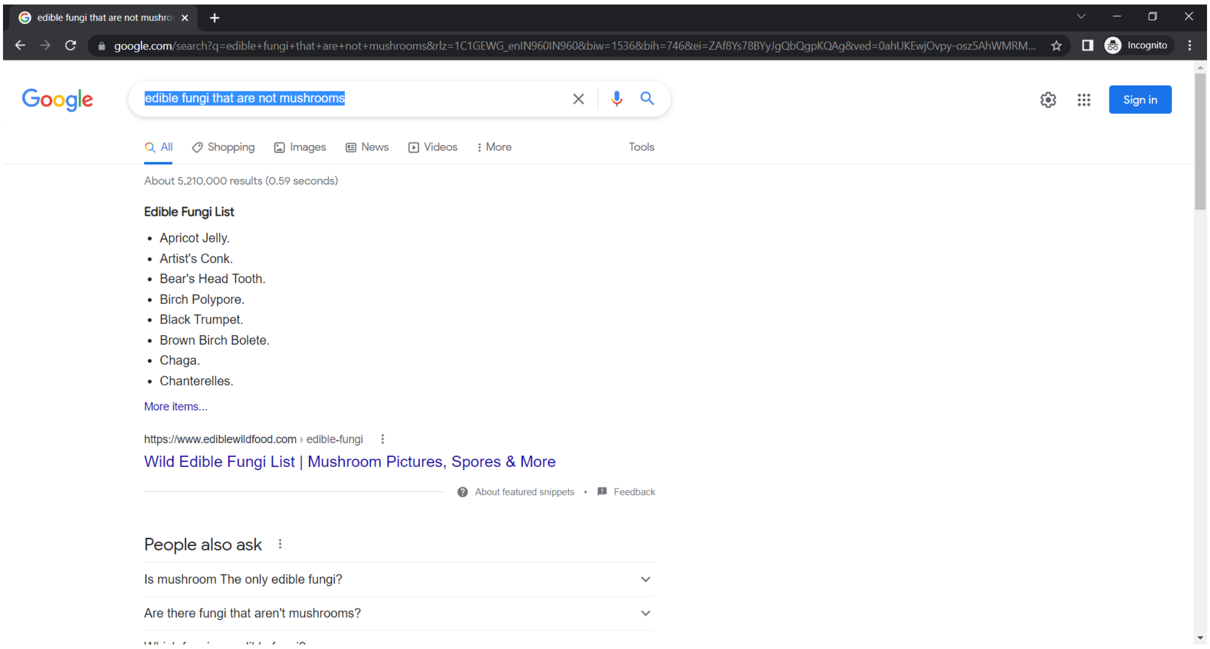
Figure 2.2: Semantic Web as a 'Web of Documents'

Envisioned by Tim Berners-Lee in his seminal article(20), the concept deals with the exchange of data in a machine-operable manner with data that is meaningful for humans.

The underlying infrastructure of the web will enable sharing of knowledge and highly context specific data, defined using existing tools of the web.

The idea was to enable to creation of "intelligent agents" (20) or software agents that would use the data already existing on the web, but curated by its content creators to have consistent vocabulary and formats. These agents would then automate tasks and knowledge exchanges between each other.

Even though the web is a large repository of data, browsing and searching through its contents is not a trivial task. Search portals can only offer data that may be present in the markup text and metadata of web pages, but multimedia content still needs additional effort to be indexed and catalogued. Even with data that is indexed, complex queries can yield irrelevant results. For example, using Google to search for 'edible fungi that are not mushrooms' results in the top hit being a listicle of edible mushrooms. The search does give a link to discussion forums that answer this question, but largely gives results about mushrooms instead.


edible fungi that are not mushro
google.com/search?q=edible+fungi+that+are+not+mushrooms&rlz=1C1GEWG_enIN960IN960&biw=1536&bih=746&ei=ZAf8Ys78BYyJgQbQgpKQAg&ved=0ahUKEwjOvpy-osz5AhWMRM...
Incognito
Google
Sign in
All
Shopping
Images
News
Videos
More
Tools

edible fungi that are not mushrooms

About 5,210,000 results (0.59 seconds)

Edible Fungi List

- Apricot Jelly.
- Artist's Conk.
- Bear's Head Tooth.
- Birch Polypore.
- Black Trumpet.
- Brown Birch Bolete.
- Chaga.
- Chanterelles.

More items...

https://www.ediblewildfood.com › edible-fungi

Wild Edible Fungi List | Mushroom Pictures, Spores & More

About featured snippets • Feedback

People also ask

Is mushroom The only edible fungi?

Are there fungi that aren't mushrooms?

Figure 2.3: Searching web isn't perfect. Search results querying 'edible fungi that are not mushrooms'

Even relatively simple queries can give interesting but ultimately unhelpful results. Searching 'Alan Smith' for the footballer gives multiple results for a English footballer that played as a striker.

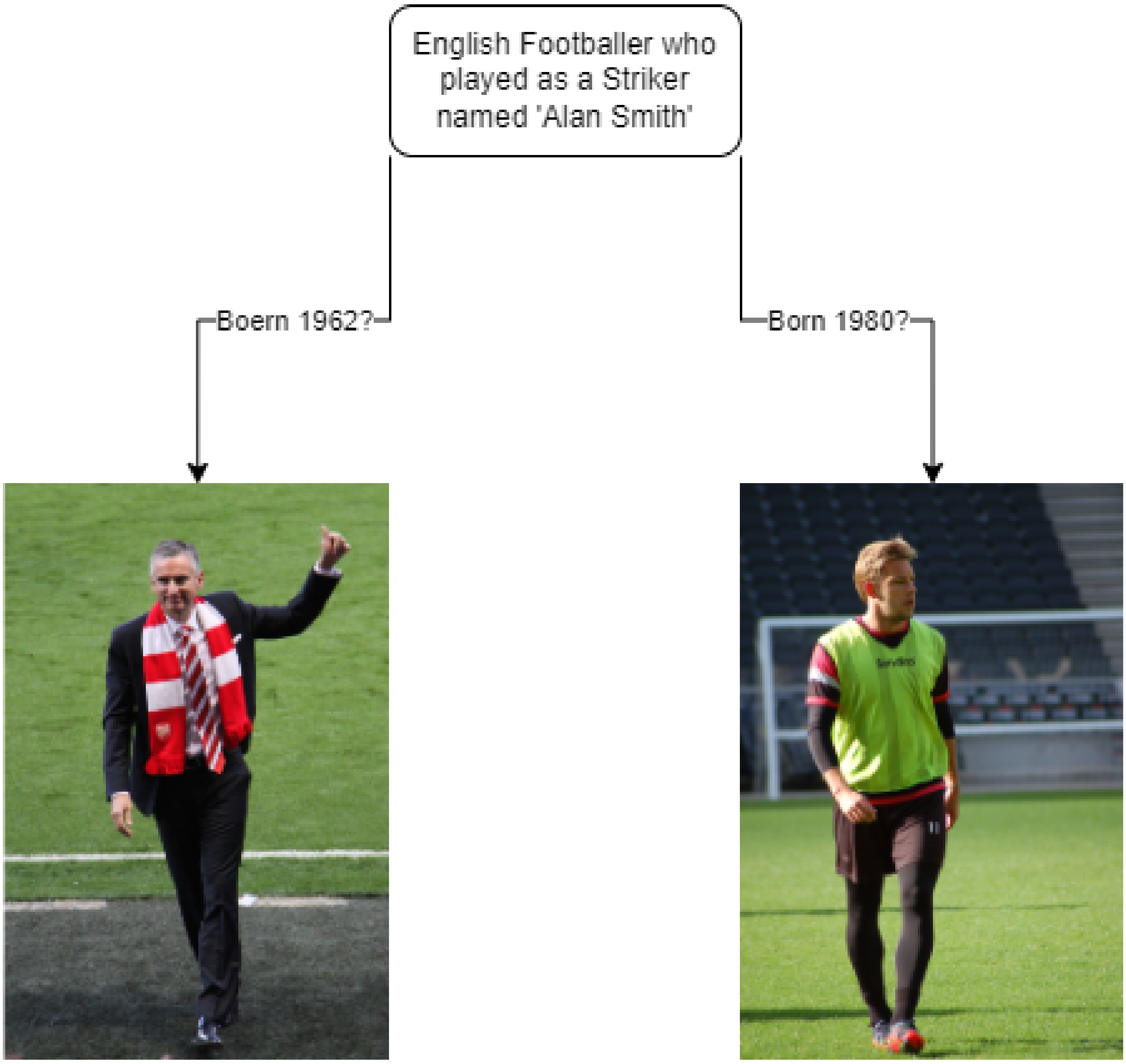

English Footballer who played as a Striker named 'Alan Smith'
Boern 1962?
Born 1980?

Figure 2.4: Which Alan Smith?. Images courtesy of Flickr, sourced from (1) and (2)

This is because the design of web content is focused on interaction with humans.(21) Since it is more driven towards presentation, the use of so called intelligent agents may be impeded. To enable these agents to understand and interpret data in a machine-friendly way, the semantic web proposes adding semantic annotations to give meaning to web content, thereby enabling Tim Berners-Lee's vision for a 'Web of Data'.

To allow the sharing of meaningful data, it is also essential to share a common language of standards and vocabularies. This is referred to as 'Ontologies'.(20) Ontologies further contain a Taxonomy and Inference Rules. Taxonomies defines the classes of entities within a dataset and the kind of relationships that exist between these entity classes. Inference rules further provide more information that can be extracted from patterns of usage of entities within an ontology.

To achieve this goal, the World Wide Web Consortium (W3C) formalised the Web ontology language(OWL).(22) The OWL is designed to represent things, collection of things and the relationship between them. The current version of OWL is OWL2 which was published in 2009 and later revised in 2012. The standard defines things as 'Instances' which belong to a 'Class'. Classes can be instances of classes called 'Metaclasses'. An attribute of a class is called a 'Property'.

2.3 Linked Data

Linked Data, in the context of the Semantic Web, is a collection of disparate datasets that are linked with each other.(23) This is established through the use of common formats, such as the Resource Description Framework(RDF) and through open query endpoints from different systems. This is where knowledge graph database systems enable graphs that comply with the RDF framework, to be available through REST endpoints over HTTP.

Linked Data allows the referencing of entities across wide and disparate sets of data. The presence of open-access endpoints over existing web protocols allows sharing of data from multiple sources.

To enable data on the web to be linked, Tim Berners-Lee lays out four rules or expectations. (24)

1. Represent things as URIs
2. Use HTTP URIs so information about things can be looked up
3. Use open standards like RDF* and SPARQL to provide useful information about a URI
4. Link other related URIs so the web of data expands

Saving data following the above 'rules' allow the web of data to be queried and reasoned over. It links disparate data related to a common entity or concept together in a way that can be followed from one dataset or web location to all other instances referencing that data.

In 2010, Tim Berners-Lee extended this concept (24) to include a 5-star rating system judging the quality of linked data provided by a service. He also introduces 'Linked Open Data' as linked data which is available under an open licence such as Creative Commons. The openness of the data does not affect its rating since its essential that some data remain internal, but at the same time follows the principles of linked data to have meaning.

The quality of linked data as determined by the rating system follows this incremental approach:

1. Available on the web. For open data, it must have an open licence

2. Be structured in order to be machine-readable
3. Structure of the data must be non-proprietary
4. Use W3C Open Standards such as RDF and SPARQL for identification of things
5. Have links to data from other sources

At any given level, the dataset must also include the provisions of the levels above it to achieve that rating.

2.4 Recommender Systems

Recommender Systems are systems that provide predictions for users based on their past behavior. They are models that are designed to help suggest users items of interest based on their activity or the trends within a larger user base. These suggestions could be for online shopping, audio/video content or users and content on social media.

Recommender Systems are typically of two broad approaches: Content Filtering and Collaborative Filtering. There are also approaches that combine both these systems into a hybrid approach.

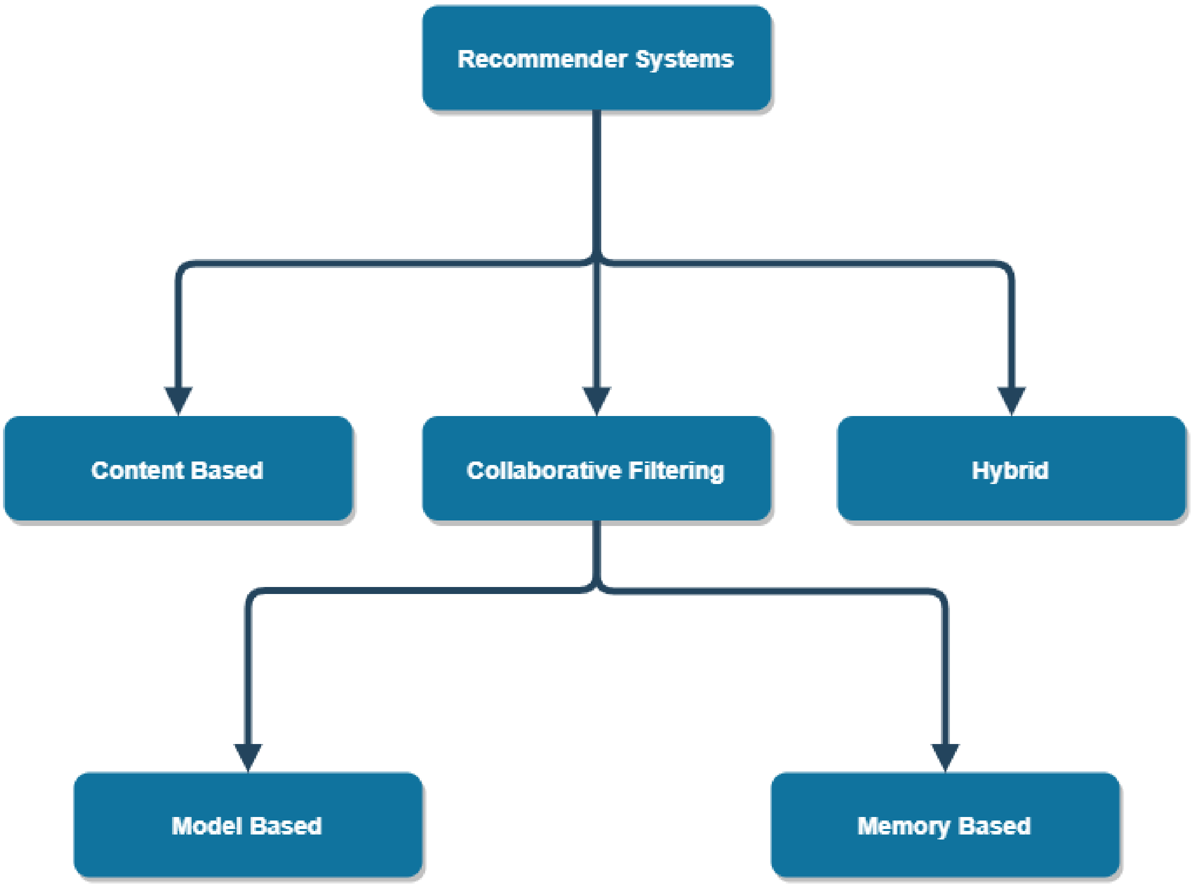


Figure 2.5: Types of Recommender Systems

2.4.1 Collaborative Filtering

Collaborative Filtering is one of the most popular techniques used by recommender systems. The idea behind CF is to make predictions or recommendations for a user based on the data collected about the behaviour of a larger population that the user is a part of.(25) It assumes that if two users have the same reaction or preference on a given topic, there is a high probability they might also have the same reaction and preference for another topic. Hence, preferences that are not common between the two users can be suggested to the other user assuming that there will be a high probability that the trend of sharing common interest continues.

Broadly, collaborative filtering is of two types: Model based and Memory based.

Memory based collaborative filtering collect behavioral data about a larger user base to suggest unknown or unexplored items to a user based on a similarity measure, by grouping users into cohorts.

Model based collaborative filters on the other hand, deploy a machine learning model to collect user data and predict the probability of a user favouring an unknown data item based on statistical methods.

2.4.2 Content Filtering

Content Filtering, as opposed to Collaborative Filtering, observes the user's behavior to generate a model or stereotype of the user. These systems rely on matching the user's preferences with an attribute of the overall content within the system to suggest recommendations. A basic assumption in these types of systems is that the content is well known and annotated. The system is then responsible to guess the highest probability of a user preferring a content item based on their preferences and attributes of the content.

This dissertation can also be classified as a Content Filtering type of recommender but with the caveat that the usage of knowledge graphs removes the need for similarity matching between attributes and preferences. The advantages of semantic flexibility of a knowledge graph overcomes the need to apply machine learning techniques or similarity measures, giving deterministic results.

2.5 Natural Language Processing

Natural Language Processing is the application of linguistics, machine learning and artificial intelligence for the usage of computers with human language. Specifically, it is the usage and development of algorithms that are able to interpret and analyse data in natural language. Due to the growth and popularity of machine learning models, in particular neural networks, it has been increasingly possible to develop systems and models that can interpret human language and break down the data into accurate parts of speech and language tokens. With the advancements in cloud technologies and efficient, high capacity storage and computing power now available, highly sophisticated language models have been developed that can correctly interpret and analyse highly complex patterns of speech and language. The different language models considered for this project were GPT-3 and Diffbot. Ultimately, since Diffbot was ready to use as an off-the-shelf product, it was chosen as part of the system's design.

2.5.1 Methods

Grammar dictates syntax in natural language and hence, the earliest NLP models were rule-based systems that could break down text. These models were known as statistical models. Since these rules were hand-coded they were highly restrictive in scale and couldn't deal with complexities in human language.

Machine learning and in particular, deep learning neural networks were then applied to these models to deal with increasingly complex speech. These models were better to scale since a denser and larger network could accommodate for multiple statistical probabilities of structure of language, and hence could learn with larger and larger volumes of data.

2.5.2 Examples

Highly trained NLP models now exist that can accurately analyse text and natural language. With vast amounts of annotated data now available, initiatives were undertaken to create models that mimic human behavior with text recognition as well as text synthesis. Some of these models are discussed in the following sections.

GPT-3

Generative Pre-Trained Transformer 3 is the latest in the line of highly sophisticated GPT models created by OpenAI designed to translate text between languages, answer questions asked in natural language, paraphrase large texts as well as synthesize textual output in natural language. Publicly accessible through an API, the model can be trained to receive natural language input and return parts of speech within the text in the form of subject, verb and object.

Diffbot

Diffbot is a suit of NLP AI tools that allow extraction of parts of speech from natural langauge. Diffbot has its own offering of a knowledge graph and allows building a graph from natural language text by breaking natural speech into knowledge graph triplets. The

Diffbot API accepts natural language text and returns inferences and analysis of the content with breakdown of subject, verb and objects within the text.

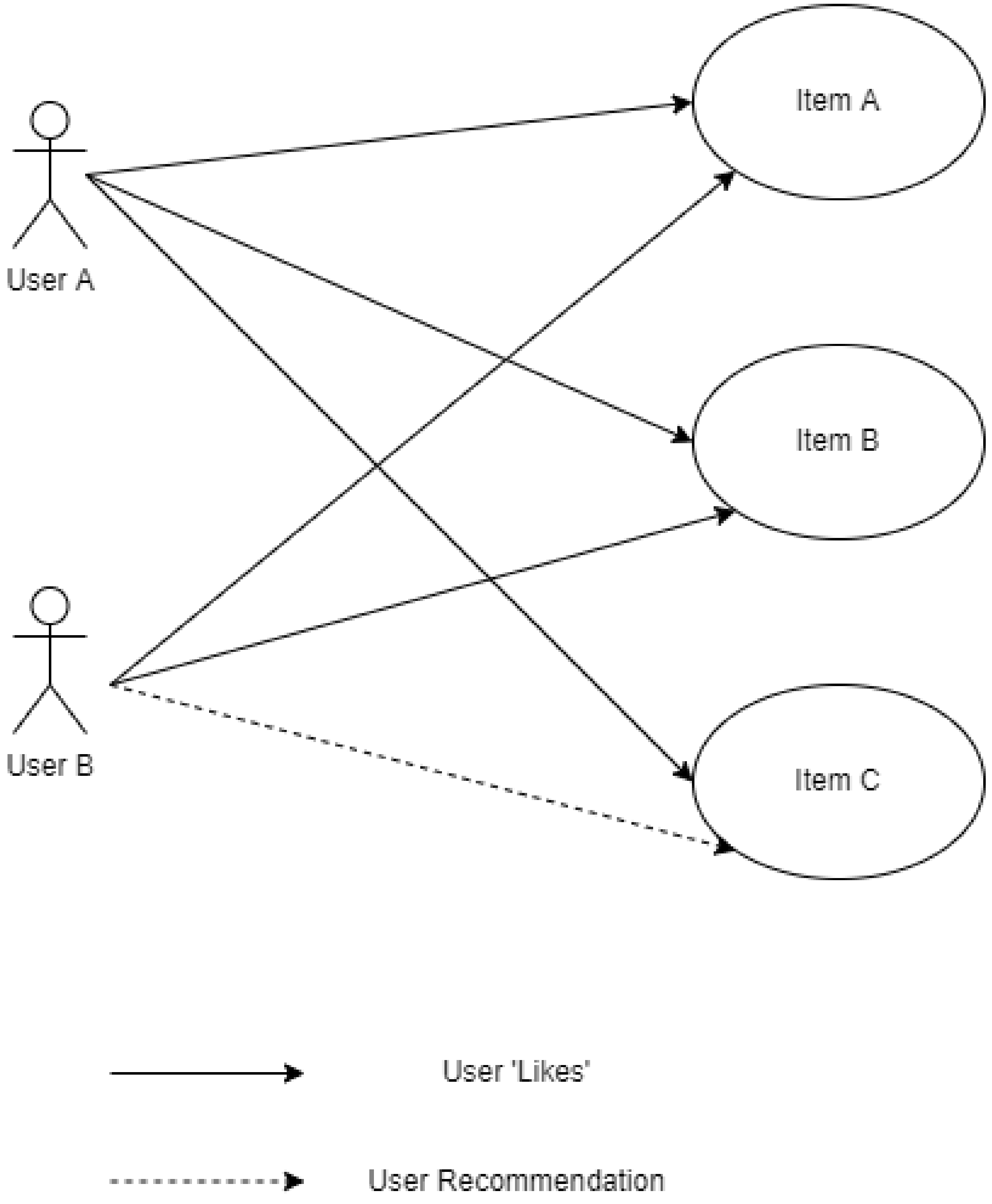


Figure 2.6: Collaborative Filtering - in a nutshell

3 Existing Work

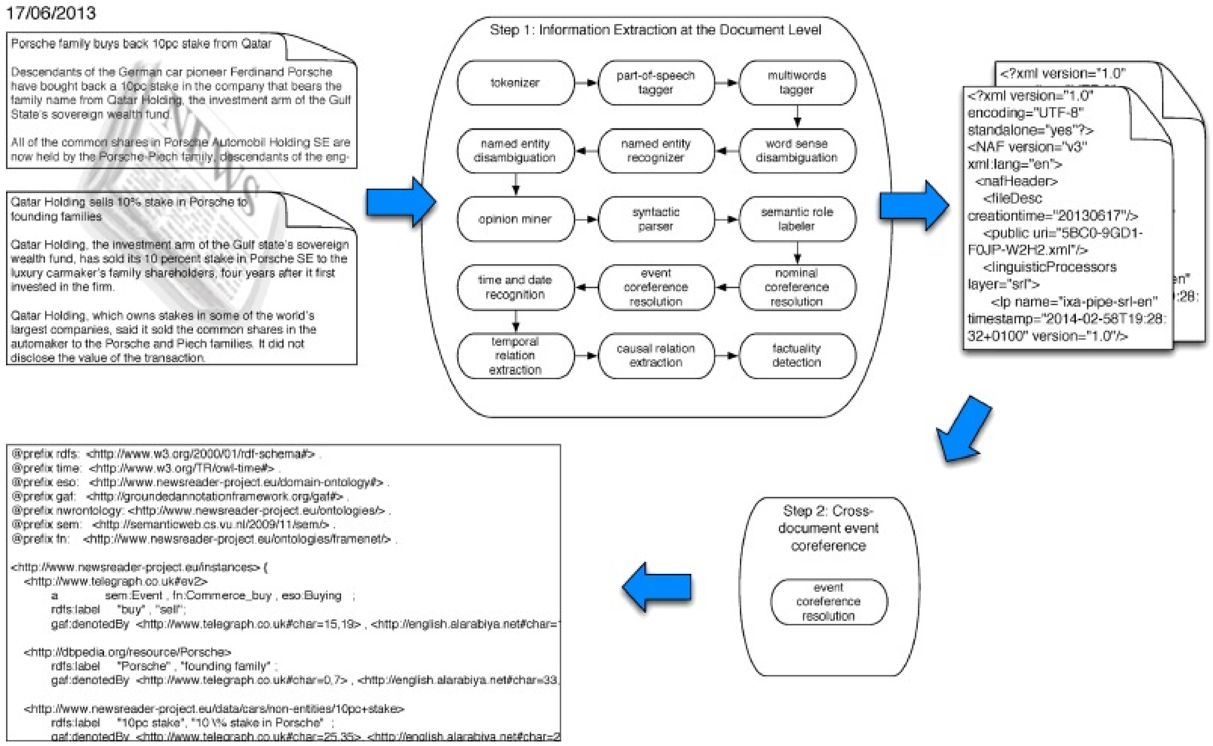
3.1 NewsReader Project

The NewsReader project was a collaborative project between multiple universities. The goal was to build a NLP pipeline that was capable of extracting and identifying similar events and related information across multiple sources.(26) These sources were mostly news articles publicly available from news sites.

The project utilised a knowledge graph to represent this information. By using the RDF standard it was also able to link the data internally to external knowledge graphs and linked data sources.

The idea for this thesis was heavily drawn from NewsReader. This project established that large volume of data can be managed and processed using off the shelf ML tools. It also established that this data can be reasoned over and the language models can be fine tuned to extract all information accurately.

The NewsReader project used a pipeline of language analysis to break down parts of speech and annotate tokens within those parts of speech to entities, events or concepts inside natural language text. This output generated from this pipeline was fed through another step to generate a combined RDF representation of all semantic information with references to the same entities or events combined into one. The RDF document was then ready to be loaded onto a graph databse.(3)



17/06/2013

Porsche family buys back 10pc stake from Qatar

Descendants of the German car pioneer Ferdinand Porsche have bought back a 10pc stake in the company that bears the family name from Qatar Holding, the investment arm of the Gulf State's sovereign wealth fund.

All of the common shares in Porsche Automobil Holding SE are now held by the Porsche-Piech family, descendants of the eng-

Qatar Holding sells 10% stake in Porsche to founding families

Qatar Holding, the investment arm of the Gulf state's sovereign wealth fund, has sold its 10 percent stake in Porsche SE to the luxury carmaker's family shareholders, four years after it first invested in the firm.

Qatar Holding, which owns stakes in some of the world's largest companies, said it sold the common shares in the automaker to the Porsche and Piech families. It did not disclose the value of the transaction.

```
<?xml version="1.0"
encoding="UTF-8"
standalone="yes"?>
<NAF version="v3"
xml:lang="en">
  <nafHeader>
    <fileDesc
creationtime="20130617"/>
    <public uri="5BC0-9GD1-
F0JP-W2H2.xml"/>
    <linguisticProcessors
layer="srl">
      <lp name="ixa-pipe-srl-en"
timestamp="2014-02-58T19:28:
32+0100" version="1.0"/>
```

```
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix time:  <http://www.w3.org/TR/owl-time#> .
@prefix eso:   <http://www.newsreader-project.eu/domain-ontology#> .
@prefix gaf:   <http://groundedannotationframework.org/gaf#> .
@prefix nwrontology: <http://www.newsreader-project.eu/ontologies/> .
@prefix sem:   <http://semanticweb.cs.vu.nl/2009/11/sem/> .
@prefix fn:    <http://www.newsreader-project.eu/ontologies/framenet/> .

<http://www.newsreader-project.eu/instances> {
   <http://www.telegraph.co.uk#ev2>
         a              sem:Event , fn:Commerce_buy , eso:Buying   ;
         rdfs:label     "buy" , "sell" ;
         gaf:denotedBy  <http://www.telegraph.co.uk#char=15,19> , <http://english.alarabiya.net#char=

   <http://dbpedia.org/resource/Porsche>
         rdfs:label     "Porsche" , "founding family" ;
         gaf:denotedBy  <http://www.telegraph.co.uk#char=0,7> , <http://english.alarabiya.net#char=33,

   <http://www.newsreader-project.eu/data/cars/non-entities/10pc+stake>
         rdfs:label     "10pc stake", "10 \% stake in Porsche"  ;
         gaf:denotedBy  <http://www.telegraph.co.uk#char=25,35>, <http://english.alarabiya.net#char=2
```

Step 1: Information Extraction at the Document Level
tokenizer
part-of-speech tagger
multiwords tagger
word sense disambiguation
named entity recognizer
named entity disambiguation
opinion miner
syntactic parser
semantic role labeller
nominal coreference resolution
event coreference resolution
time and date recognition
temporal relation extraction
causal relation extraction
factuality detection
Step 2: Cross-document event coreference
event coreference resolution

Figure 3.1: The NewsReader NLP architecture taken from (3)

This established the possibility of using already developed ML tools to scrape data that is already publicly available on the web, and enable building a knowledge base to query from. Further, building this pipeline would also enable arbitrary facts and knowledge to be entered into the system via a simple interface that just needs textual input from the user.

It also established that with a correctly trained model, entities and events in natural text can also be identified and extracted with confidence.(26) Thus, it was practical to consider using ML models to use natural language sources of data into reasonable knowledge stores.

The NewsReader project consists solely of consulting graphs within its own knowledge stores and hence needs high capacity storage for all the data that is parsed. In contrast with this research, it was determined that in order to be efficient in memory and storage, it was necessary to connect with external knowledge graphs.

3.2 Cold start issues with Collaborative Filtering

Collaborative Filtering models are poor with handling data that was not visible to it during training. (8) The fallback is often to consider accommodating new embeddings within the model or using heuristics and thereby, approximate embeddings. (8). Both approaches only partially address this problem.

A study about methodologies and metrics for cold start evaluation (27) also provided insight into the usage of heuristics to approximate embeddings in a sparse dataset. This further lends credence to the value of using semantic attributes related to the content in recommender engines that do not have enough data.

Another study evaluated using user data and preferences to calcualte similarity metrics. These are then fed to a Back Propagation Neural Network(BPNN) to enhance recommendation accuracy. (28) Even though it did overcome the inaccuracy of cold start problems it did make the overall design more complex by introducing hybrid methodologies for recommendations.

To mitigate cold start issues the system either needs vast volumes of data or needs to implement even more complex design to fine tune its accuracy.

3.3 Data collection and privacy issues with Collaborative Filtering

As described in 2.4, current recommender system involve constant collection and monitoring of user behaviour data. Possession and processing of this data introduces the risk of privacy breaches.

In 2007, researchers from the University of Texas were able to deanonymise Netflix's anonymised dataset released as part of their data science challenge. (7) By combining the anonymised Netflix data with data from IMDB comments, it was possible to identify the users within the Netflix dataset.

Lam et al. (4) demonstrated the various attack surfaces for recommender systems. While passive attacks of manipulating recommendations cannot be completely avoided by design, the 'Exposure' risks identified in this research can certainly be reduced to a great degree by designing systems that do not rely on collecting data for the purpose of generating probabilities and similarty scores.

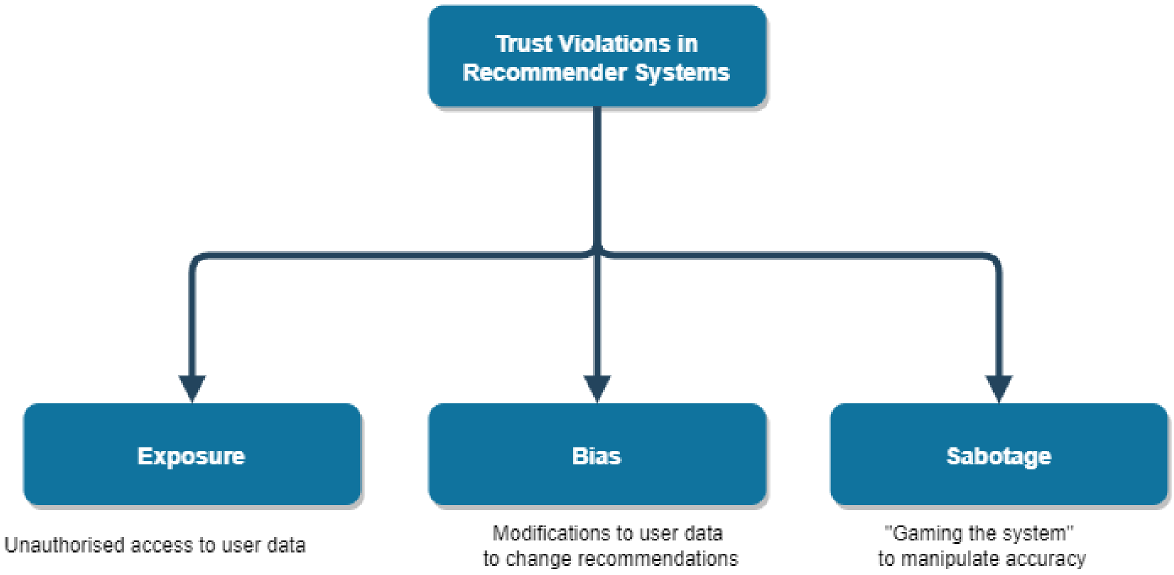

Trust Violations in Recommender Systems
Exposure
Unauthorised access to user data
Bias
Modifications to user data to change recommendations
Sabotage
"Gaming the system" to manipulate accuracy

Figure 3.2: User Trust Violations in Recommenders. Source: Lam et al. (4)

3.4 Exploring Graph Networks with Machine Learning

Collaborative Filtering style techniques can also be used with graph datasets to suggets recommendations as demonstrated in this paper (29) that explores the different techniques used in graph based network of user behavior data. Traditional statistical methods of traversing through the graph network exist but run into problems of cold start and data sparsity.(29).

This paper further delves into the usage and performance of neural network models in discovering knowledge graph embeddings as a means to suggest recommendations. In comparision with collaborative filtering, it performs with high accuracy even with a sparse dataset. To a large extent, this mitigates the cold start problem of Collaborative Filtering.(29)

Despite these advantages, this method still relies on collecting user behavior data and hence is prone to the risks with user privacy as stated in previous chapters. Furthermore, using machine learning approaches to infer relationships within a graph network is not deterministic in nature and can become difficult to reason over extremely large datasets.

3.5 Comparative Review

The following table addresses some of the issues in the existing state of the art systems that this dissertation aims to address. This dissertation doesn't aim to directly replicate a project such as the NewsReader, but rather takes findings from the project and use it to tackle the problem of recommendations. This thesis also identifies the issues that are associated with the approach in the NewsReader project and a plan to mitigate them. Since this dissertation is in the domain of recommendations, the existing approaches to recommendation engines are compared against the approach suggested in this thesis.

Table 3.1: Aspects of existing work addressed by this dissertation

Work	Issues	Fixes
NewsReader Project	High compute capacity for NLP	Usage of off-the-shelf tools for entity and triplet extraction
Collaborative Filtering	Cold start inaccuracy and storage inefficiency	Using linked open data and external knowledge graph sources
Collaborative Filtering	User Privacy	Using knowledge graph for semantically matching user preferences
Content Filtering	Approximating/Guessing user preferences with content	Querying semantic data within knowledge graph

4 Design and Implementation

4.1 Design

The user interface was prototyped with the React framework to make a single page web application. The goal with the interface was to provide a means to stereotype the user model, supply natural language description to be added to the graph database and show recommendations, all of which could be simplified with React's ability to separate design with components and data with Redux.

A crucial part of the system was modelling the user's preferences and the means to gather that information. Calculating the 'Big 5' personality traits of a user was first considered to make an estimation of the user's preferences.(30) A 2021 research on using stock images and the response of users was considered to be adopted as an on-boarding tool to gather initial data of the user. (31) The thresholds and ranges on the Big 5 spectrum would then be used to match against a set of attributes of an entity in the knowledge graph being queried.

This approach had its own limitations. As mentioned in chapter 1.4, this way of modelling a user's preference wouldn't lend itself to scrutability. Even if the user could view and even change their traits on the Big 5 spectrum, the way the system would utilise this model would still be opaque to the user. Additionally, the set of attributes available on an entity was not predictable for external knowledge graph sources. This would have limited the number of external data sources that could have been used to query for recommendations.

To overcome this limitation, it was decided to build a user interface that gave the user a set of preferences to select from that resembled real world entities, concepts or activities. This approach may not be versatile, since users may not like to search and enter their preferences manually in every context. However, this approach lends itself more transparent to scrutiny by the user. It also enables the usage of semantic information inherent in knowledge graphs since preferences can be literally matched with node and predicate labels.

The backend only needed to be a glue layer that could make SPARQL queries and provide APIs to call from the interface. Hence, the Flask framework was chosen for its simplicity and ability to scale. Some advantages that made it appropriate for the system were:(32)

- High Flexibility
- Highly Scalable
- Small Codebase Size
- High Performance

Since the backend also had to interact with the graph database, it was crucial to select a database that had strong Python SDK support. AllegroGraph provided a web interface as well Java and Python support along with other advantages such as:(33)

- Transaction Support for durability
- SOLR integration for indexing
- In-built graph visauliser
- Geospatial and Temporal support

To demonstrate the flexibility of the approach described in this thesis it was essential to also find an external knowledge graph data source which was:

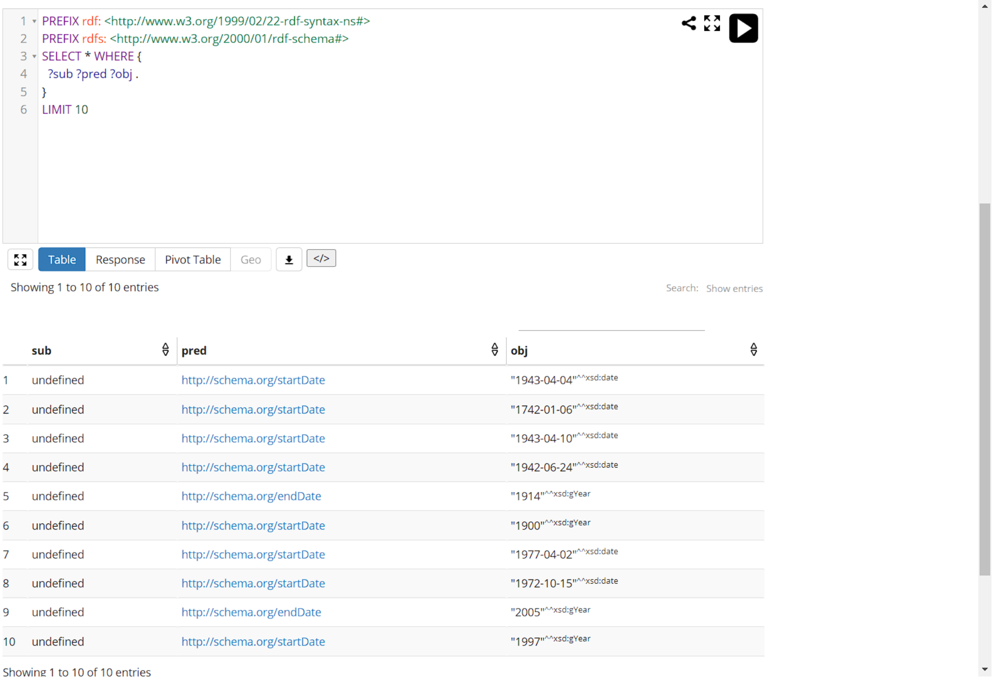
- Highly Curated
- Had public SPARQL endpoint with generous usage limit
- Based on and consisted of large volumes of data

Knowledge Graphs based on Wikipedia were the candidates chosen because of their large data volumes. Two popular projects provided this service in DBPedia and WikiData.

DBPedia has a publicly hosted SPARQL endpoint with a limited dataset available. (34). Even with limited dataset, the endpoint restrcited the number of results that could be returned. (35). The recommended approach from DBPedia was to host a dockerised instance of the DBPedia service. Apart from adding to the hosting costs, this also adds the responsibility of maintaining data of an external source to the system.

WikiData was chosen for its public SPARQL endpoint(36) and the nature of curated wikipedia entries within it(37). There were no limits applied to the querying service with WikiData (36) with an additional advantage of having a label service for entities within its knowledge graph.(38)

Yago was an alternative that was considered. A knowledge graph service about people, cities, countries, movies, and organizations, it was also a candidate that could prove to be a source of recommendations. However, at the time of writing, the service wasn't reliable and would return unusable data as demonstrated in the figure below.



```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT * WHERE {
 ?sub ?pred ?obj .
}
LIMIT 10
```

Showing 1 to 10 of 10 entries

Search: Show entries

	sub	pred	obj
1	undefined	http://schema.org/startDate	"1943-04-04"^^xsd:date
2	undefined	http://schema.org/startDate	"1742-01-06"^^xsd:date
3	undefined	http://schema.org/startDate	"1943-04-10"^^xsd:date
4	undefined	http://schema.org/startDate	"1942-06-24"^^xsd:date
5	undefined	http://schema.org/endDate	"1914"^^xsd:gYear
6	undefined	http://schema.org/startDate	"1900"^^xsd:gYear
7	undefined	http://schema.org/startDate	"1977-04-02"^^xsd:date
8	undefined	http://schema.org/startDate	"1972-10-15"^^xsd:date
9	undefined	http://schema.org/endDate	"2005"^^xsd:gYear
10	undefined	http://schema.org/startDate	"1997"^^xsd:gYear

Showing 1 to 10 of 10 entries

Figure 4.1: Unreliable results from the Yago query service

As seen in the screenshot above, some results would come as 'undefined' which suggests a lack of curation on the dataset in the knowledge graph. This violated one of the preconditions needed of an external data source as mentioned above.

To extract triplets from natural text to update the internal knowledge graph, there were three approaches evaluated. The first one was to use the NewsReader project's NLP pipeline as described in their published study (3). This pipeline consisted of multiple individual components written in multiple languages and needed to be orchestrated together. It also resulted in generation of large volumes of files for intermediary processes thereby making it infeasible. It did however generated highly accurate and richly annotated triplets, even though this level of detail was not essential for this use case.

The second approach was much more simplified. A python library 'Textacy' was evaluated to extract tokens from sentences and identify subject, object and verb parts of speech.(11) The library was capable of extracting this information, however, it could only process one sentence at a time. It was also incapable of cross referencing pronouns to appropriate subjects, hence missing out on implicit information.

Finally, a NLP service called 'Diffbot' was evaluated.(12) Diffbot had a REST API capable of extracting subject-verb-object information from natural language and was capable of referencing pronouns within text. In certain instances, it was also able to infer more information based on articles and pronouns used. Since this was an external service, there was no cost of maintenance as well. Hence, Diffbot was chosen to provide data to update the internal knowledge graph.

4.2 Architecture

The system is composed of three parts:

- User Interface: Front End Web App
- Backend
- Graph Databases: Internal Graph Database and external Knowledge Graphs

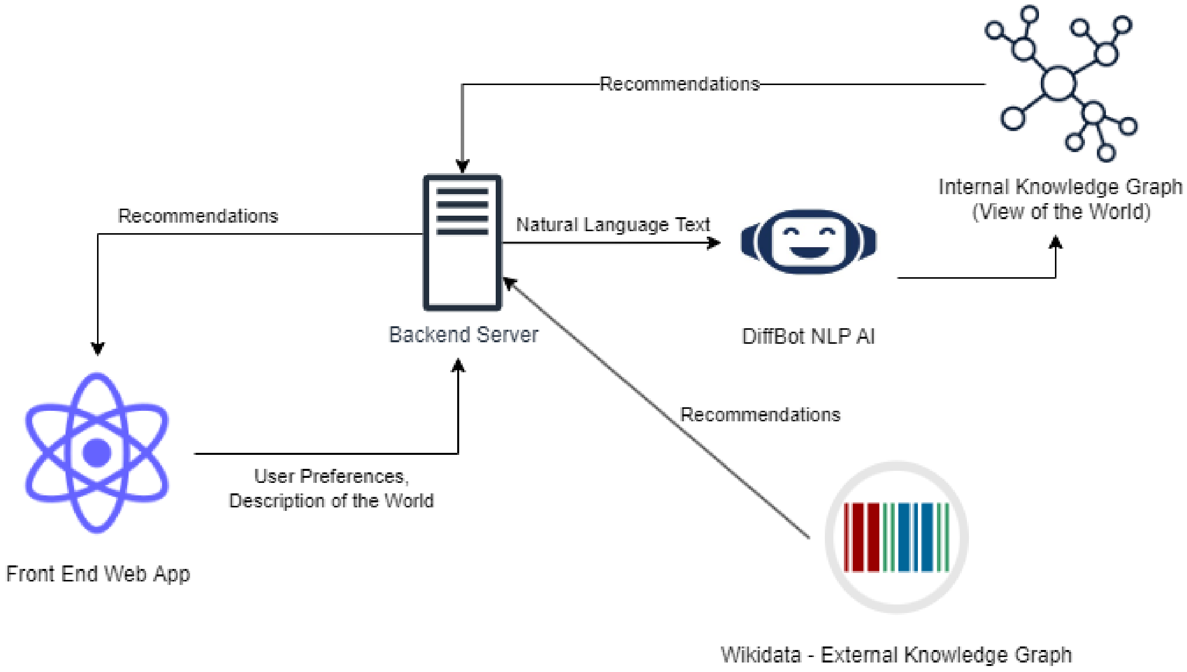


Figure 4.2: Architecture of the system

4.3 Implementation

4.3.1 User Interface

The user interface consists of three screens:

- Preferences - for user preferences
- Recommendations - to get recommendations
- Knowledge Base - for adding statements to be added to knwoledge graph

Preferences

The interface is a simple collection of possible interests for the user to select. The user is able to select or deselect their likes for the kind of recommendations they'd like to prefer.

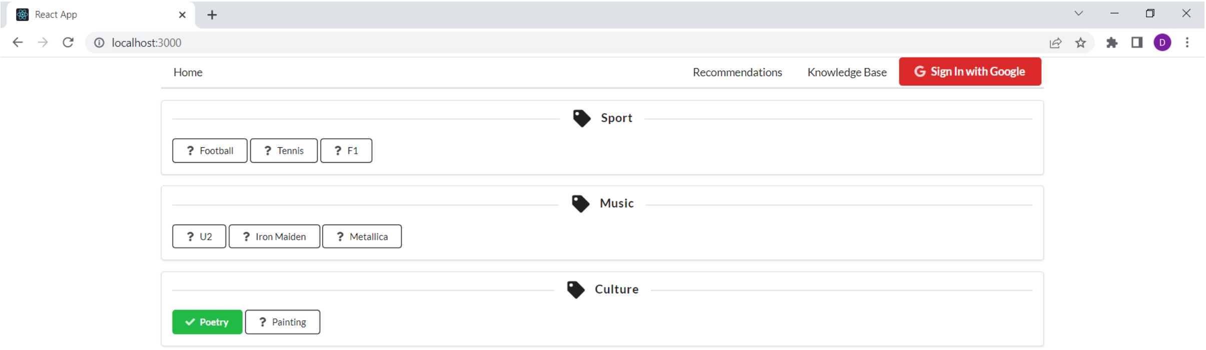
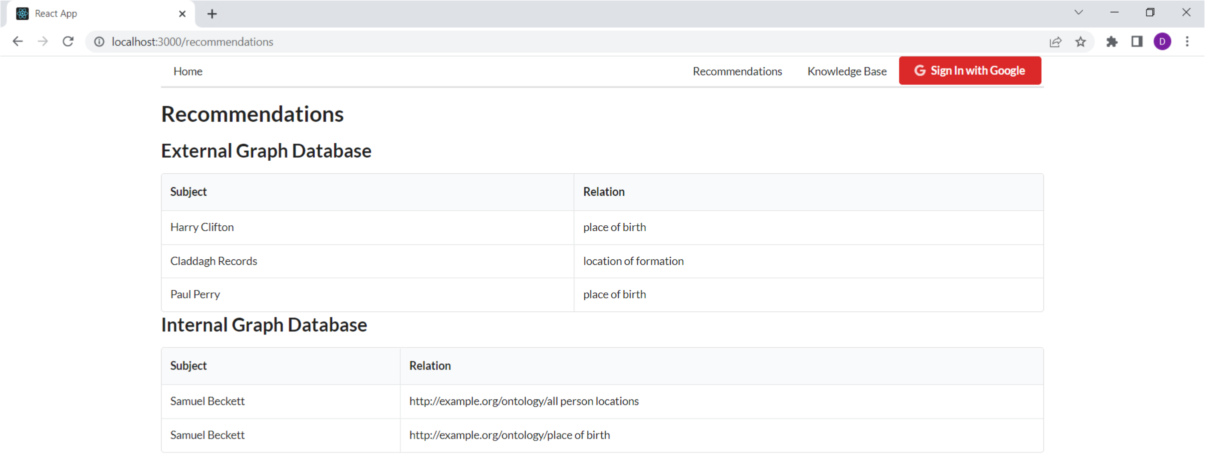


Figure 4.3: User Preferences Screen

Recommendations

The Recommendations screen queries the external and internal graph databases for nodes that match the user preferences. The entities returned from the query and their connection to the user's preferences are then returned.



Recommendations

External Graph Database

Subject	Relation
Harry Clifton	place of birth
Claddagh Records	location of formation
Paul Perry	place of birth

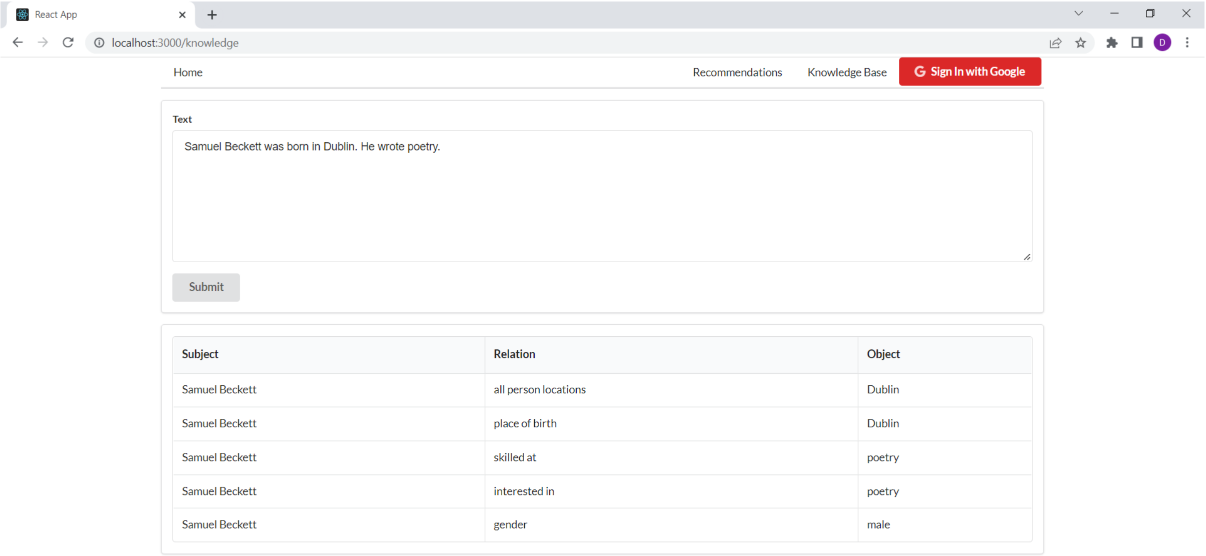
Internal Graph Database

Subject	Relation
Samuel Beckett	http://example.org/ontology/all person locations
Samuel Beckett	http://example.org/ontology/place of birth

Figure 4.4: Recommendations Screen

Knowledge Base

To demonstrate the usage of NLP to add to the existing, internal knowledge graph as well update the internal knowledge graph, this screen allows entering text in natural language that is parsed and interpreted by the NLP service to geenrate triplets. The page also displays the result of the operation.



Text

Samuel Beckett was born in Dublin. He wrote poetry.

Subject	Relation	Object
Samuel Beckett	all person locations	Dublin
Samuel Beckett	place of birth	Dublin
Samuel Beckett	skilled at	poetry
Samuel Beckett	interested in	poetry
Samuel Beckett	gender	male

Figure 4.5: Knowledge Base Screen

4.3.2 Graph Database

The user model consists of the 'likes' of a user which are the user's preferences. The user's identifier as well as the preferences that are saved as string literals within the graph database.

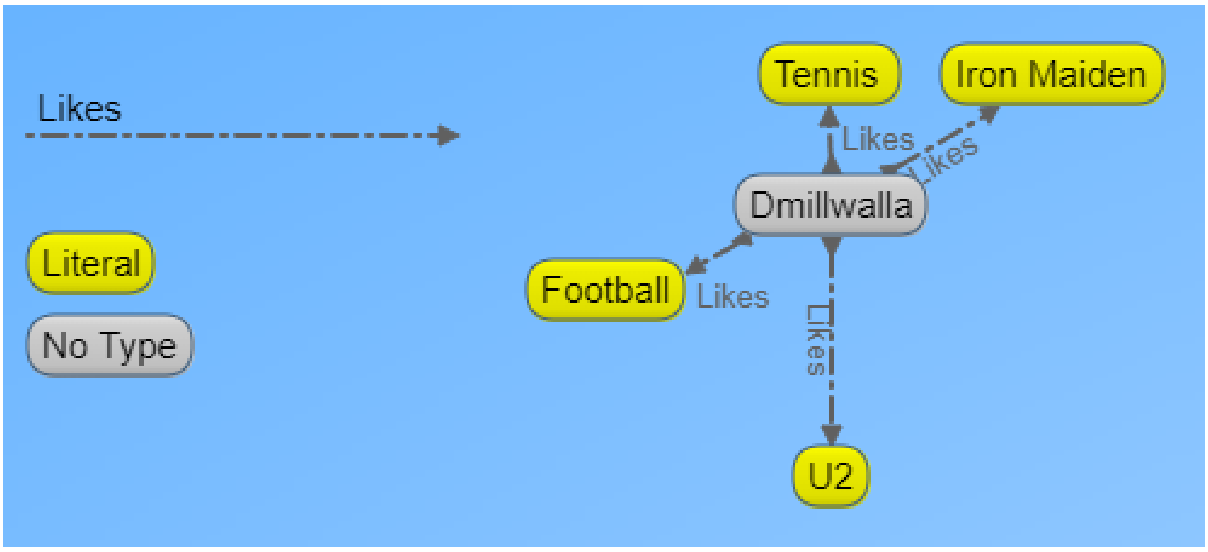


Figure 4.6: Graph of the user's preferences

The view of the world graph consists of entities and their relationships extracted by the Diffbot API. Entities are stored as string literals within the graph database and the relationships have a placeholder ontology prefix.

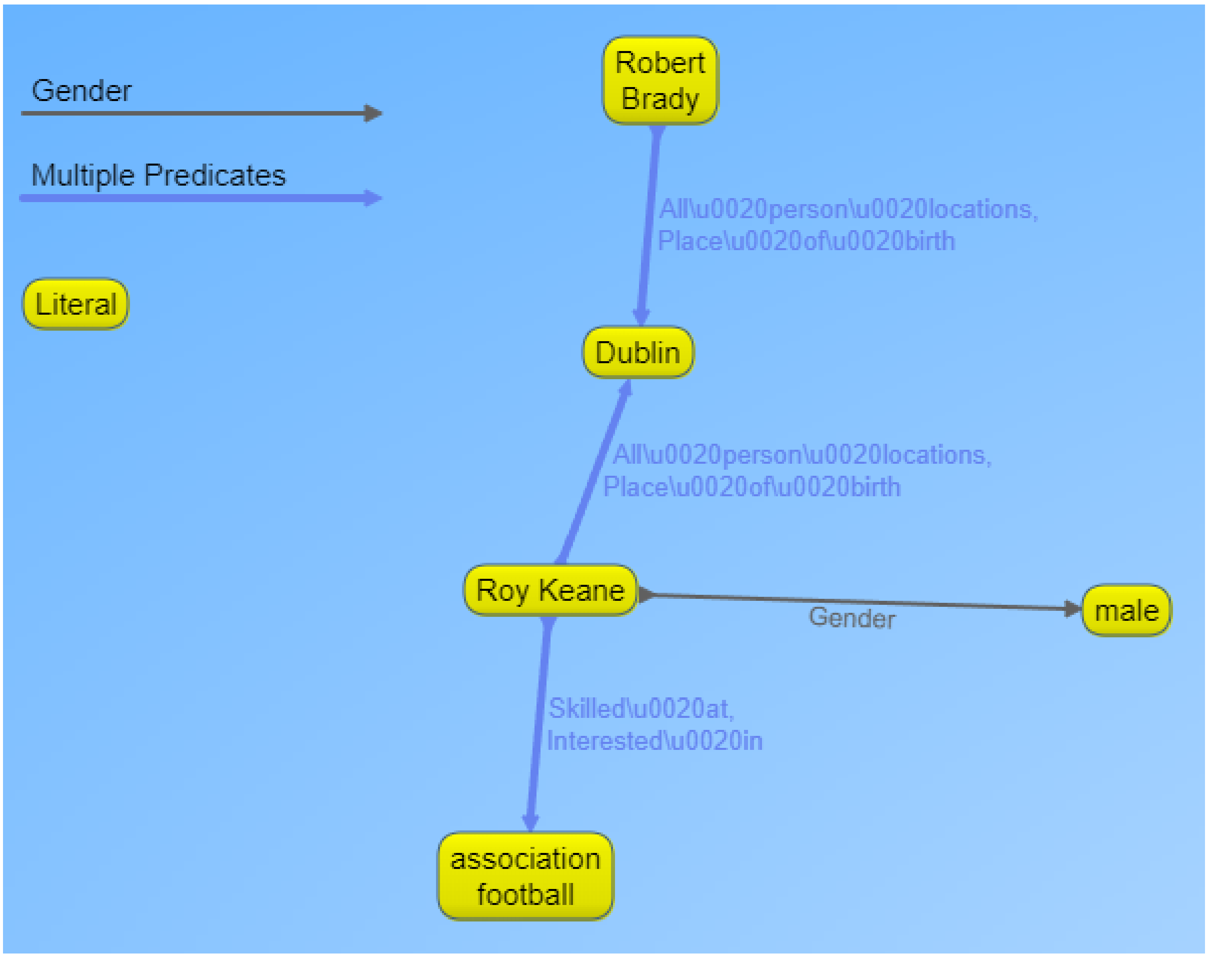

Gender
Multiple Predicates
Literal
Robert Brady
All\u0020person\u0020locations, Place\u0020of\u0020birth
Dublin
All\u0020person\u0020locations, Place\u0020of\u0020birth
Roy Keane
Gender
male
Skilled\u0020at, Interested\u0020in
association football

Figure 4.7: View of the World Graph

Using literals allows simplified queries to be used to extract information from the graph. Having an established ontology can provide more flexibility with the query to match nodes or predicates, but the purpose of this thesis is to demonstrate the simplicity of using graphs. An example query used by this system is:

```
SELECT  ?s ?p WHERE
        {
            ?s ?p "Dublin" .
            ?s ?p1 ?o1 .
            FILTER (  regex(str(?o1), "\\bPoetry\\b", "i")  ) .
        }
        LIMIT 100
```

This query allows the literals from the user preferences('Poetry' in this example) to be directly used for regex matching with the internal and external graph databases.

4.4 System Demonstration

Assume a user logging in for the first time. The preference screen allows the user to select their preferences across different categories.

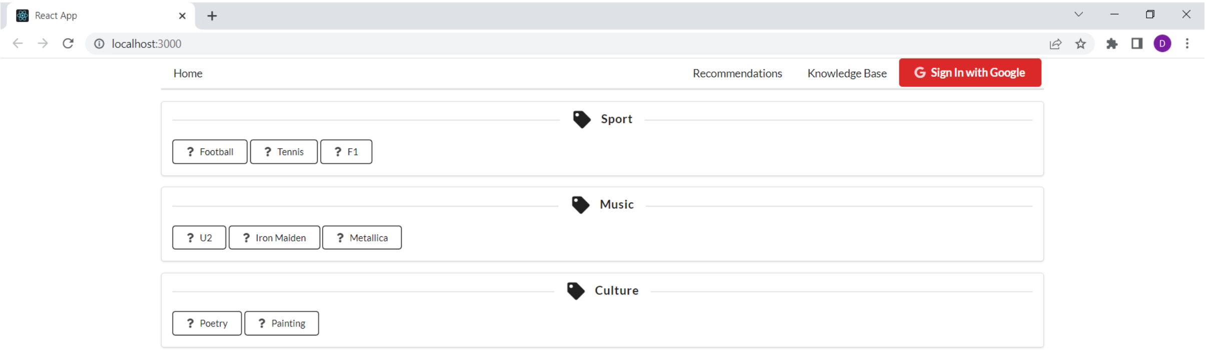


Figure 4.8: User preference selection screen

Assuming the user selects 'Poetry' as a preference, the user model in the internal graph database is updated.

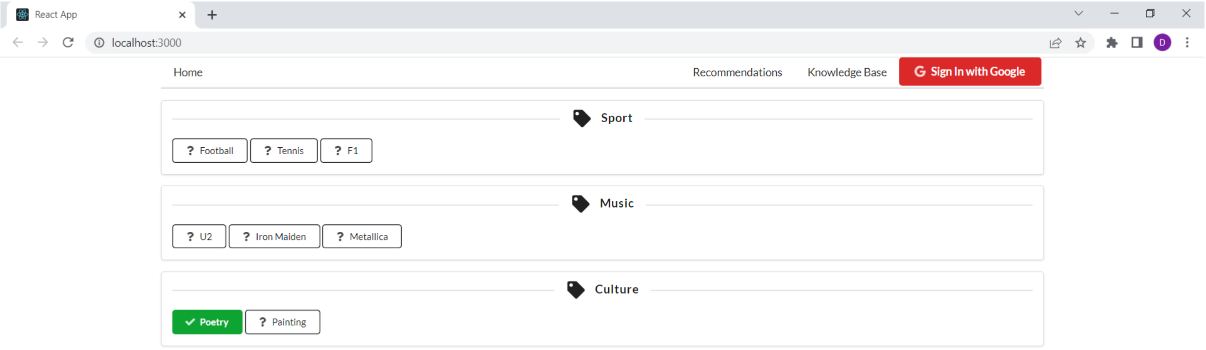
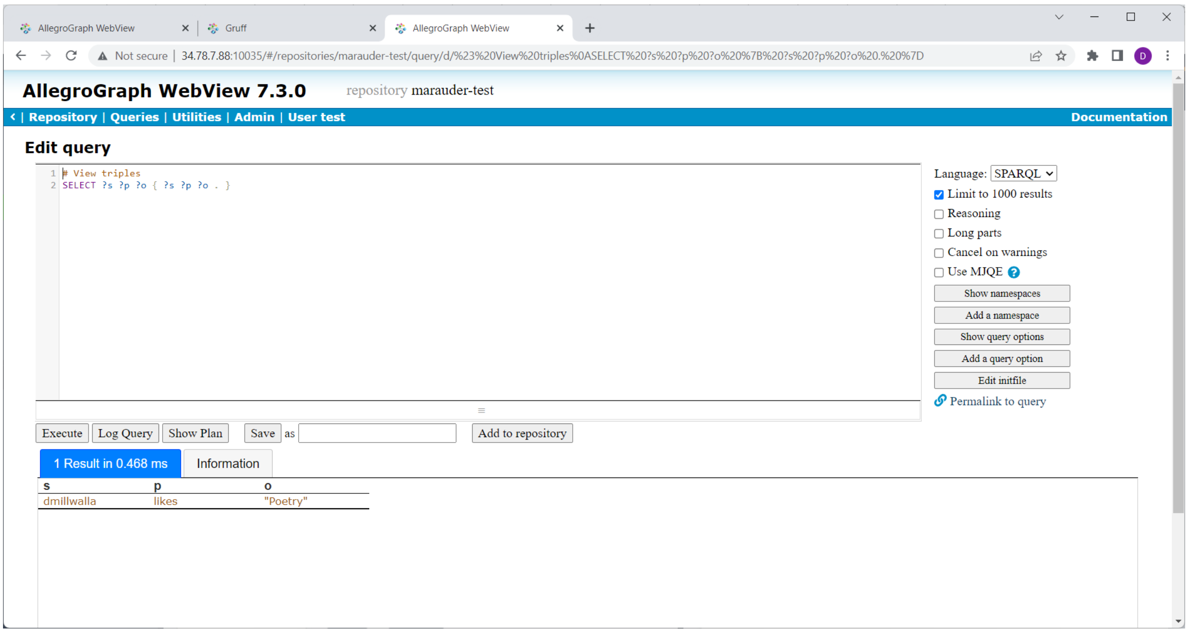


Figure 4.9: Preference update on Screen



```
# View triples
SELECT ?s ?p ?o { ?s ?p ?o . }
```

Language: SPARQL

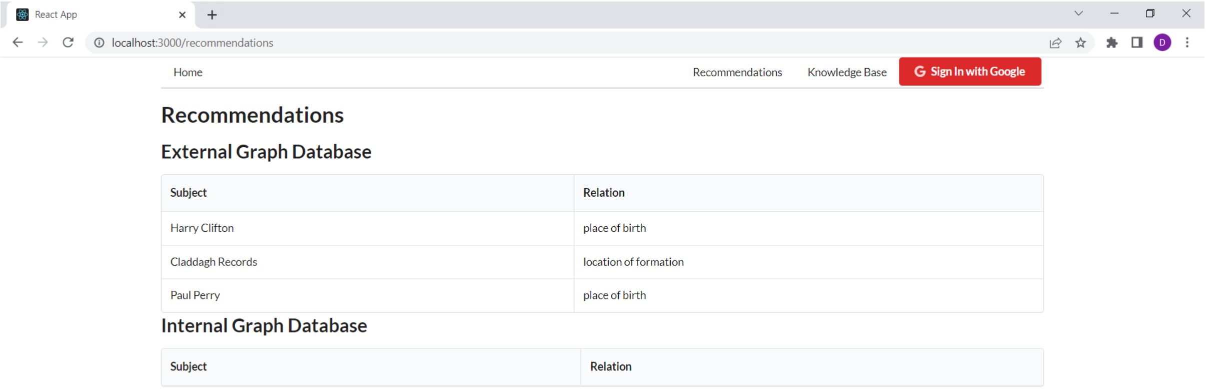
☑ Limit to 1000 results
☐ Reasoning
☐ Long parts
☐ Cancel on warnings
☐ Use MJQE

1 Result in 0.468 ms

s	p	o
dmillwalla	likes	"Poetry"

Figure 4.10: Preference saved in graph database

The user can then go to Recommendations page to look up recommendations related to their preferences(currently 'Poetry') in the context of Dublin. Since the internal knowledge graph doesn't have any information currently, it does not return any recommendations but the external graph database service WikiData returns some recommendations for the user.



Recommendations

External Graph Database

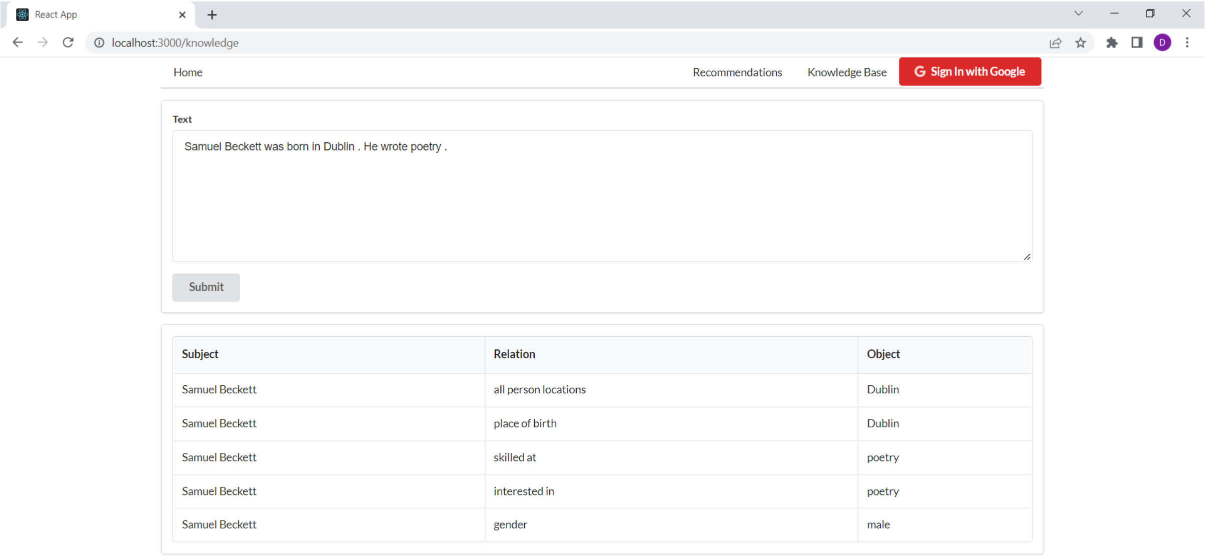
Subject	Relation
Harry Clifton	place of birth
Claddagh Records	location of formation
Paul Perry	place of birth

Internal Graph Database

Subject	Relation

Figure 4.11: Recommendations from WikiData

The Knowledge Base screen has the ability to add knowledge to the system to enhance recommendations. It consists of a text input that can accept natural language and send to the backend where Diffbot breaks down the natural language to knowledge graph triples.

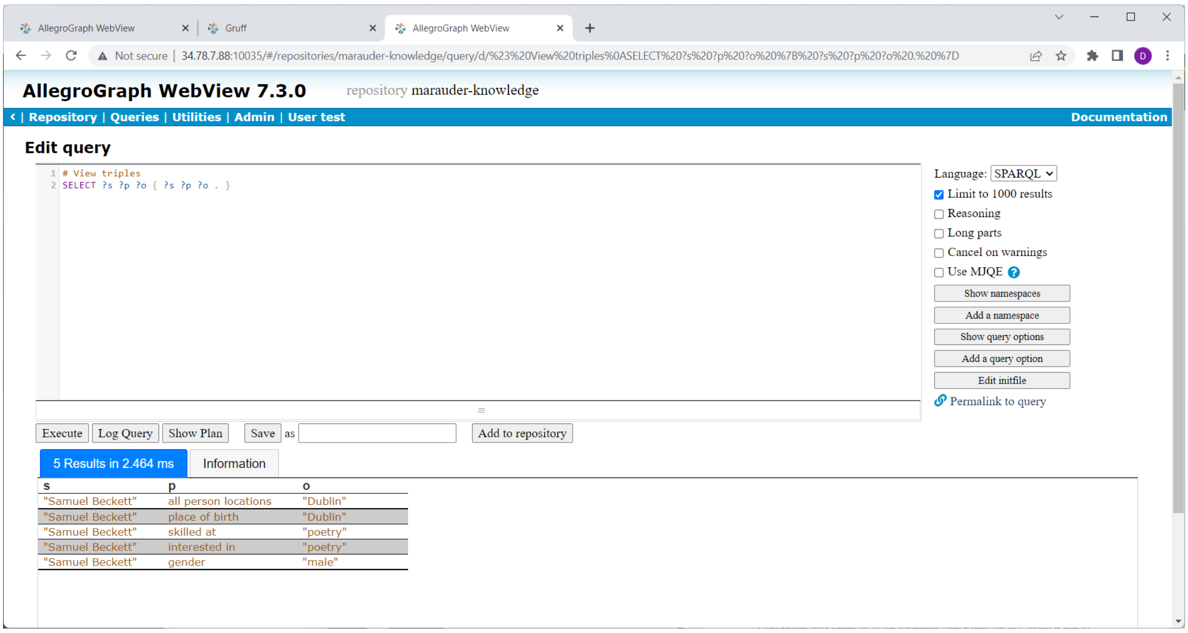


Text

Samuel Beckett was born in Dublin . He wrote poetry .

Subject	Relation	Object
Samuel Beckett	all person locations	Dublin
Samuel Beckett	place of birth	Dublin
Samuel Beckett	skilled at	poetry
Samuel Beckett	interested in	poetry
Samuel Beckett	gender	male

Figure 4.12: Triples broken down by Diffbot


AllegroGraph WebView
Gruff
AllegroGraph WebView
Not secure | 34.78.7.88:10035/#/repositories/marauder-knowledge/query/d/%23%20View%20triples%0ASELECT%20?s%20?p%20?o%20%7B%20?s%20?p%20?o%20.%20%7D
AllegroGraph WebView 7.3.0
repository marauder-knowledge
‹ | Repository | Queries | Utilities | Admin | User test
Documentation
Edit query
Show namespaces
Add a namespace
Show query options
Add a query option
Edit initfile
Permalink to query
Execute
Log Query
Show Plan
Save
as
Add to repository
5 Results in 2.464 ms
Information

```
# View triples
SELECT ?s ?p ?o { ?s ?p ?o . }
```

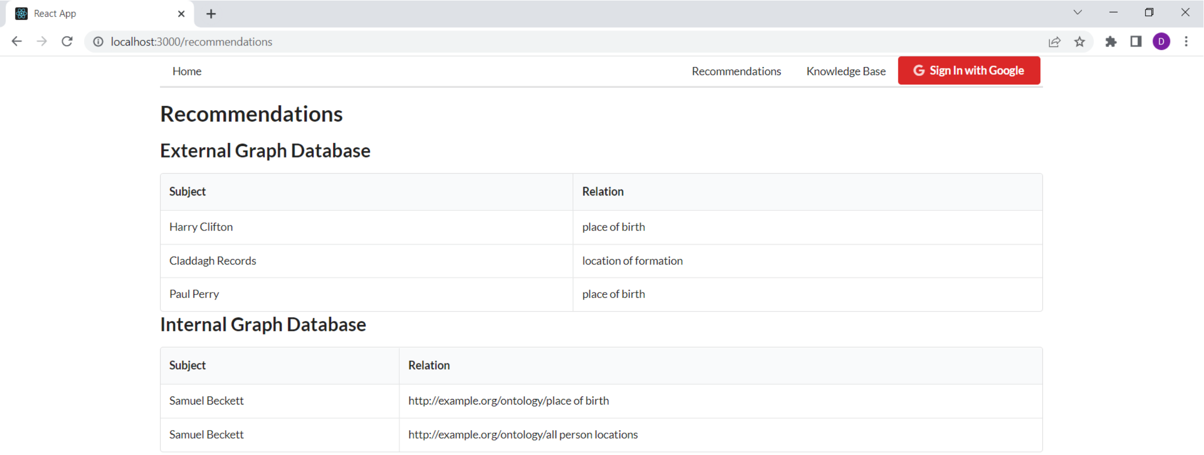
Language: SPARQL

☑ Limit to 1000 results
□ Reasoning
□ Long parts
□ Cancel on warnings
□ Use MJQE

s	p	o
"Samuel Beckett"	all person locations	"Dublin"
"Samuel Beckett"	place of birth	"Dublin"
"Samuel Beckett"	skilled at	"poetry"
"Samuel Beckett"	interested in	"poetry"
"Samuel Beckett"	gender	"male"

Figure 4.13: Triples added to graph database

Going back on the recommendations screen, the system can now pick up the new information added to the internal knowledge graph. The recently added information about Samuel Beckett is now returned to the user.



Recommendations

External Graph Database

Subject	Relation
Harry Clifton	place of birth
Claddagh Records	location of formation
Paul Perry	place of birth

Internal Graph Database

Subject	Relation
Samuel Beckett	http://example.org/ontology/place of birth
Samuel Beckett	http://example.org/ontology/all person locations

Figure 4.14: Recommendations from WikiData and internal knowledge graph

The demonstration shows that the system is able to link the user preferences with the semantic data inside knowledge graphs. It also highlights how this approach is able to access data without knowing about the schema or structure of data external to the system. The internal graph database saves all nodes as literals but WikiData is highly curated with an extensive Ontology. SPARQL queries and semantic web principles offer the flexibility to access both these datasets with similar queries.

The system is able to deliver recommendations without needing to monitor the behaviour of the user. An important consideration here is that the user is responsible for outlining what kind of recommendations they wish to seek. There can be systems designed that monitor the user's interaction with content and try to determine their preferences, but this would compromise the security by design principle of the system described by this thesis. Such a system would still be able to use semantic links of large volumes of linked data and deliver deterministic recommendations.

The codebase for the entire system can be found on the repository: https://github.com/dmillwalla/dissertation. The repository also contains all instructions to set the environment up to run the application.

5 Evaluation

Since the fundamental approach of the version of a recommender system described by this thesis is different, it cannot be compared with the metrics of recall and precision used by Collaborative Filtering. However, the accuracy of this kind of recommender system depends on the accuracy of the NLP model in processing natural text to generate knowledge graph triplets. The performance of this model can be evaluated and its accuracy measured in generating information for the internal graph.

From sample texts passed on to the Knowledge Base screen, the NLP model is able to identify explicit facts as well as infer some amount of implicit knowledge about the world.

For example, for the following text:

```
Samuel Beckett was born in Dublin. He wrote poetry.
```

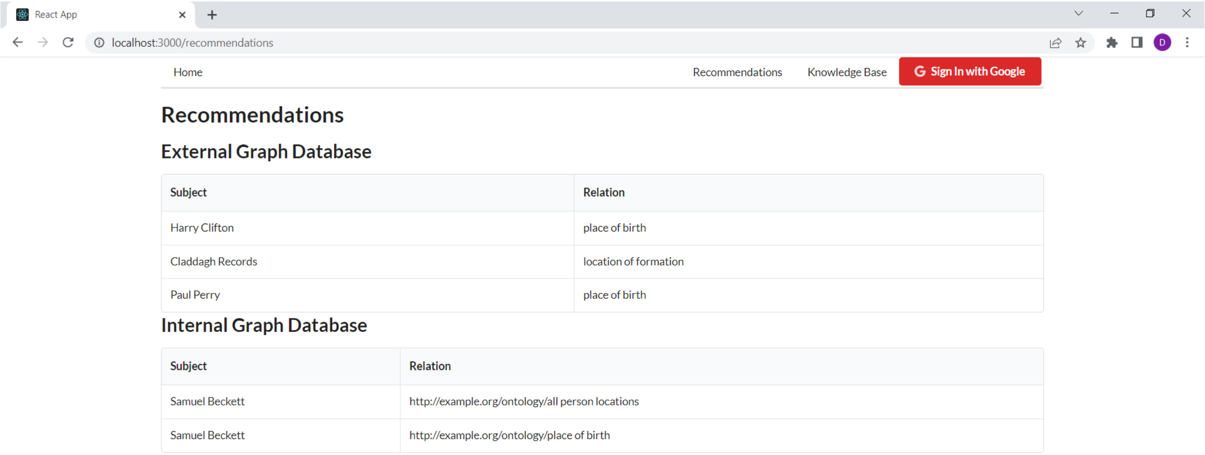
The model is able to infer the following facts:

Table 5.1: Facts Inferred by Diffbot - Samuel Beckett

Subject	**Predicate**	**Object**	**Implicit/Explicit**
Samuel Beckett	all persons locations	Dublin	Explicit
Samuel Beckett	place of birth	Dublin	Explicit
Samuel Beckett	skilled at	poetry	Explicit
Samuel Beckett	interested in	poetry	Explicit
Samuel Beckett	gender	male	Implicit

The NLP model is able to infer by the usage of the pronoun 'He' that Samuel Beckett must also be male.

The recommendations correctly pick up this fact from the graph database when poetry is a part of user's preferences:



Recommendations

External Graph Database

Subject	Relation
Harry Clifton	place of birth
Claddagh Records	location of formation
Paul Perry	place of birth

Internal Graph Database

Subject	Relation
Samuel Beckett	http://example.org/ontology/all person locations
Samuel Beckett	http://example.org/ontology/place of birth

Figure 5.1: Poetry Recommendations

Considering another example:

```
U2 are an Irish rock band from Dublin, formed in 1976. It
   consists of Bono, the Edge, Adam Clayton, and Larry Mullen
    Jr.
```

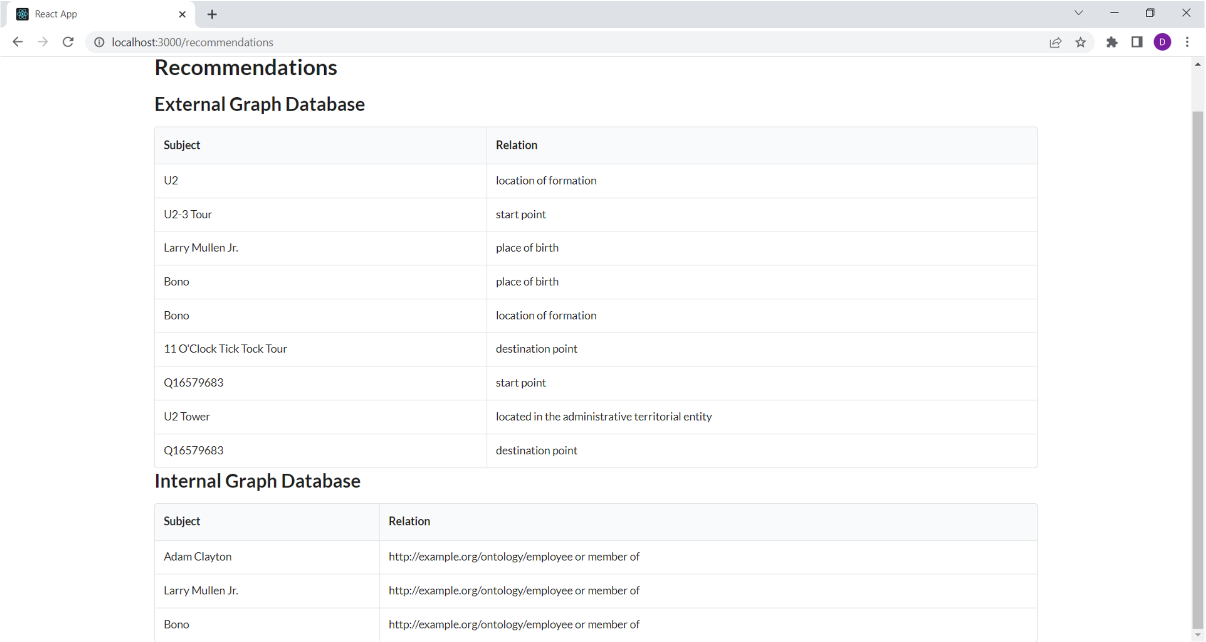
The model is able to infer the following facts:

Table 5.2: Facts Inferred by Diffbot - U2

Subject	Predicate	Object	Implicit/Explicit
Adam Clayton	employee or member of	U2	Explicit
Larry Mullen Jr.	employee or member of	U2	Explicit
Larry Mullen Jr.	work relationship	Bono	Implicit
Larry Mullen Jr.	work relationship	Adam Clayton	Implicit
Adam Clayton	work relationship	Bono	Implicit
U2	organization locations	Dublin	Explicit
Bono	work relationship	Larry Mullen Jr.	Implicit
Adam Clayton	work relationship	Larry Mullen Jr.	Implicit
Bono	work relationship	Adam Clayton	Implicit
U2	founding date	1976	Explicit
U2	industry	rock music	Explicit
Bono	employee or member of	U2	Explicit

The model is again able to identify that individual members of a group are also colleagues and have a working relationship with each other.

The recommendations reflect the added entities related to U2 as well:



Recommendations

External Graph Database

Subject	Relation
U2	location of formation
U2-3 Tour	start point
Larry Mullen Jr.	place of birth
Bono	place of birth
Bono	location of formation
11 O'Clock Tick Tock Tour	destination point
Q16579683	start point
U2 Tower	located in the administrative territorial entity
Q16579683	destination point

Internal Graph Database

Subject	Relation
Adam Clayton	http://example.org/ontology/employee or member of
Larry Mullen Jr.	http://example.org/ontology/employee or member of
Bono	http://example.org/ontology/employee or member of

Figure 5.2: U2 Recommendations

It is however, not perfect. Using the first example, and changing the sentence structure slightly to use this instead:

```
Samuel Beckett wrote poetry. He was born in Dublin.
```

The model is able to infer the following facts:

Table 5.3: Samuel Beckett facts with changed sentence structure

Subject	Predicate	Object	Implicit/Explicit
Samuel Beckett	all persons locations	Dublin	Explicit
Samuel Beckett	place of birth	Dublin	Explicit
Samuel Beckett	gender	male	Implicit

The model fails to pick up an explicitly mentioned fact. There is more room for improvement with the models used and care must be taken to curate the content and training the natural language model with enough variations of data.

The above examples are curated inputs that test Diffbot in a limited manner. To evaluate its performance in a real world scenario, the model is fed excerpts from Wikipedia articles.

For example, consider the following excerpt taken from the 'Irish Art' Wikipedia page (39)

```
The visual arts were slow to develop in Early Modern Ireland,
    due to political disruption, and the lack of patrons in
   either government, the church, and wealthy resident
   landowners or business class interested in art. Yet
   beginning in the late 17th century, Irish painting began
   to develop, especially in portraiture and landscape
   painting. These painters typically looked outside Ireland
   for influence, training and clients who were wealthy
   enough to afford the purchase of art. For example, Walter
   Frederick Osborne developed his open air painting in
   France whereas Sir William Orpen studied in London.
   However, what is now the National College of Art and
   Design in Dublin has existed since founded as the Dublin
   Art School in 1746. Its founder Robert West had studied
   drawing and painting at the French Academy under François
   Boucher and Jean-Baptiste van Loo.
```

The model is able to infer the following facts:

Table 5.4: Irish Art Facts - Part 1

Subject	Predicate	Object	Implicit/Explicit
Robert West	skilled at	drawing	Explicit
Robert West	interested in	drawing	Explicit
Robert West	field of work	drawing	Explicit
National College of Art & Design	founded by	Robert West	Explicit
François Boucher	employee or member of	Academie de France a Rome	Explicit
National College of Art & Design	founding date	1746	Explicit
Robert West	work relationship	François Boucher	Explicit
National College of Art & Design	industry	art of painting	Explicit
William Orpen	all person locations	France	Implicit
François Boucher	skilled at	drawing	Implicit
François Boucher	interested in	drawing	Implicit
Jean-Baptiste van Loo	work relationship	Robert West	Explicit
Walter Osborne	all person locations	France	Explicit
William Orpen	skilled at	open air	Explicit
William Orpen	interested in	open air	Explicit

Table 5.5: Irish Art Facts - Part 2

Subject	Predicate	Object	Implicit/Explicit
François Boucher	work relationship	Robert West	Explicit
François Boucher	skilled at	art of painting	Implicit
François Boucher	interested in	art of painting	Implicit
Robert West	educated at	Academie de France a Rome	Explicit
National College of Art & Design	industry	drawing	Explicit
Robert West	field of work	art of painting	Explicit
Robert West	skilled at	art of painting	Explicit
Robert West	interested in	art of painting	Explicit
Robert West	employee or member of	National College of Art & Design	Explicit
William Orpen	skilled at	art of painting	Explicit
William Orpen	interested in	art of painting	Explicit
François Boucher	employee or member of	National College of Art & Design	Implicit
Robert West	position held	founder	Explicit
Walter Osborne	all person locations	London	Explicit
National College of Art & Design	organization locations	Dublin	Explicit
National College of Art & Design	headquarters	Dublin	Explicit
William Orpen	all person locations	London	Explicit
Walter Osborne	skilled at	art of painting	Explicit
Walter Osborne	interested in	art of painting	Explicit
Robert West	work relationship	Jean-Baptiste van Loo	Explicit
Walter Osborne	all names	Walter Frederick Osborne	Explicit
Walter Osborne	gender	male	Implicit

The model misinterprets certain facts such as Sir William Orpen being in France or Walter Frederick Osborne being in London. The model is largely successful in identifying entities but with certain sentence structures it doesn't correctly identify the exact relationships within them.

To test Diffbot with a difference sentence structure, a biographical page from Wikipedia is chosen. Consider this extract from James Joyce's Wikipedia page(40):

```
James Augustine Aloysius Joyce (2 February 1882 - 13 January
   1941) was an Irish novelist, poet and literary critic. He
   contributed to the modernist avant-garde movement and is
   regarded as one of the most influential and important
   writers of the 20th century. Joyce's novel Ulysses (1922)
   is a landmark in which the episodes of Homer's Odyssey are
    paralleled in a variety of literary styles, particularly
   stream of consciousness. Other well-known works are the
   short-story collection Dubliners (1914), and the novels A
   Portrait of the Artist as a Young Man (1916) and Finnegans
    Wake (1939). His other writings include three books of
   poetry, a play, letters, and occasional journalism. Joyce
   was born on 2 February 1882 at 41 Brighton Square, Rathgar
   , Dublin, Ireland, to John Stanislaus Joyce and Mary Jane
   "May" (née Murray). He was the eldest of ten surviving
   siblings. He was baptised with the name James Augustine
   Joyce according to the rites of the Roman Catholic Church
   in the nearby St Joseph's Church in Terenure on 5 February
    1882 by Rev. John O'Mulloy.
```

Table 5.6: James Joyce Facts

Subject	Predicate	Object	Implicit/Explicit
James Joyce	position held	writers	Explicit
John Stanislaus Joyce	all person locations	Rathgar	Explicit
James Joyce	date of birth	1882-02-02	Explicit
John Stanislaus Joyce	all person locations	Ireland	Explicit
James Joyce	nationality	Republic of Ireland	Explicit
James Joyce	all person locations	Republic of Ireland	Explicit
James Joyce	position held	literary criticism	Explicit
James Joyce	all person locations	Ireland	Explicit
James Joyce	place of birth	Ireland	Explicit
James Joyce	social relationship	John Stanislaus Joyce	Explicit
James Joyce	family member	John Stanislaus Joyce	Explicit
John O'Mulloy	position held	Rev.	Explicit
John Stanislaus Joyce	all person locations	Dublin	Explicit
John Stanislaus Joyce	family member	James Joyce	Explicit
John Stanislaus Joyce	social relationship	James Joyce	Explicit
James Joyce	position held	novelist	Explicit
James Joyce	date of death	1941-01-13	Explicit
James Joyce	all person locations	Rathgar	Explicit
James Joyce	place of birth	Rathgar	Explicit
James Joyce	all person locations	Dublin	Explicit
James Joyce	place of birth	Dublin	Explicit
James Joyce	position held	poet	Explicit
Catholic Church	organization locations	Terenure	Explicit
James Joyce	all names	James Augustine Joyce	Explicit
John Stanislaus Joyce	all names	Murray	Explicit
John Stanislaus Joyce	all names	Mary Jane "May"	Explicit
James Joyce	gender	male	Implicit

With simple sentences, the model is able to make better guesses. It does report some inaccuracies with John Joyce - all names - Murray triplet. But the model manages to uncover all relevant facts within the text.

Despite the drawbacks, the approach of this dissertation demonstrates the ability of NLP and ML being a good fit at enriching the existing knowledge base and allowing the semantic links to be used for recommendations. Even with inaccuracy in reporting facts, Diffbot manages to capture semantic relations within the data, and allows adding triplets that make logical sense if not always factually correct. A further enhancement can be suggested to allow the user to select and discard the facts that do not seem appropriate to ensure data is consistent. Alternate approaches can also be considered where the usage of NLP tools is instead replaced by controlled addition of facts inside the knowledge store through manual input to maintain consistency of data.

The system designed in this dissertation, by design, eliminates an aspect of privacy risk by not hoarding user behaviour data. It also enables efficiency in storage as only missing facts not present in external graph databases can be added to an internal knowledge store to supplement recommendations.

The approach suggested in this thesis clearly shows how semantic annotation of data allows deterministic filtering of content from diverse sources. The schema of external data sources is unknown to the application. Yet, through the use of linked data principles, knowledge across domains can be shared and linked effectively. The designed system accomplishes all of its goals as laid out in Section 1.4 and described in comparative review in Section 3.5.

6 Conclusion

In conclusion, it can be suggested that Knowledge Graphs can be an alternate approach to exploring recommendations. However, there are limitations in this approach that need to be considered. Designing recommender systems in this way assumes that the content to be searched on is well known and can possibly be annotated. Collaborative filtering, on the other hand, can work without knowing anything about the content it delivers as recommendations.

This approach could be a viable alternative for content platforms and go a long way in easing their cold start problems. These platforms also mitigate the limitations of this approach in terms of needing annotated data around its domain. Content platforms usually have highly curated metadata about their content which can be processed into a rich knowledge graph.

6.1 Enhancements and Improvements

6.1.1 Ontology

To restrict and enforce a structure to the internal Knowledge Graph, a topology can be imposed. This also ensures that logical constraints specific to the context of usage can be imposed on the data thereby maintaining data consistency and integrity. While not required within the project, the presence of an ontology simplifies the conceptualisation of the knowledge present within the system. It also allows the queries to be more accurate and match nodes and predicates within the graph efficiently.

6.1.2 GPT or NewsReader pipeline for KG creation

Diffbot is a great option as an off the shelf tool, but it fails to consistently identify triplets within natural language. Sentences structured differently, or using a different pronoun or article often results in implicit as well as explicit facts being omitted or misinterpreted by the model as shown in chapter 5. A highly contextualised NLP model, trained to expect input in a certain tone and manner will be more efficient at extracting the underlying tokens within the textual data. GPT-3 is a commercial model available that can be trained with annotated input to generate consistent output.

Alternatively, the NLP pileine described by the NewsReader project (3) can also be considered to generate triples. The output of this pipeline is in RDF format and can be readily ingested by a graph database if the storage needs of the pipeline and the resulting files can be accommodated.

6.1.3 Evaluating against searches in Google Maps

To test out recommendation quality and accuracy, the system described in this thesis can be tested out against the search feature in Google Maps. To compete with Google's recommendation that come from its own knowledge graph (13), the system would need to query multiple query services and hence was not in scope for this dissertation. However, the Google Maps Platform could be a real world product to compare quality of recommendations against.

6.2 Reflections

The use of knowledge graphs and semantic data modelling is a more deterministic and predictable way of linking multiple distinct data sets. As a recommender engine, with more information collected around user preferences, it can discover more paths in a knowledge graph network. The approach described in this thesis does not totally remove the need for collecting user data to model their preferences and hence still retains an element of the same privacy concerns. However, since the system doesn't need to predict similarities between data sets of user information with the content or knowledge within the platform, it can reduce the privacy risks further by not collecting user behavior data. Allowing the user to take control of their model also provides a level of scrutability and control over the system.

Bibliography

[1] Glenn Salt. Mk dons training day 29th oct 13 (26), Oct 2013. URL https://www.flickr.com/photos/36820778@N04/10563990395. Last visited: 2022-08-15.

[2] Ronnie Macdonald. Alan smith 2, Nov 2014. URL https://www.flickr.com/photos/7332125@N04/15692729721. Last visited: 2022-08-15.

[3] Piek Vossen, Rodrigo Agerri, Itziar Aldabe, Agata Cybulska, Marieke van Erp, Antske Fokkens, Egoitz Laparra, Anne-Lyse Minard, Alessio Palmero Aprosio, German Rigau, Marco Rospocher, and Roxane Segers. Newsreader: Using knowledge resources in a cross-lingual reading machine to generate more knowledge from massive streams of news. *Knowledge-Based Systems*, 110:60–85, 2016. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2016.07.013. URL https://www.sciencedirect.com/science/article/pii/S0950705116302271.

[4] Shyong K. Lam, Dan Frankowski, and John Riedl. Do you trust your recommendations? an exploration of security and privacy issues in recommender systems. In *ETRICS*, 2006.

[5] Yehuda Koren, Steffen Rendle, and Robert Bell. *Advances in Collaborative Filtering*, pages 91–142. Springer US, New York, NY, 2022. ISBN 978-1-0716-2197-4. doi: 10.1007/978-1-0716-2197-4_3. URL https://doi.org/10.1007/978-1-0716-2197-4_3.

[6] Reena Pagare and Shalmali Patil. Study of collaborative filtering recommendation algorithm scalability issue. *International Journal of Computer Applications*, 67:10–15, 04 2013. doi: 10.5120/11742-7305.

[7] Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset, 2006. URL https://arxiv.org/abs/cs/0610105. Last visited: 2022-08-15.

[8] Collaborative filtering advantages & disadvantages, 2022. URL https://developers.google.com/machine-learning/recommendation/collaborative/summary. Last visited: 2022-08-15.

[9] IBM Education. What is a knowledge graph?, 2022. URL https://www.ibm.com/cloud/learn/knowledge-graph. Last visited: 2022-08-15.

[10] Knowledge graphs and linked data gnoss, 2022. URL https://www.gnoss.com/en/graph-linked-data. Last visited: 2022-08-15.

[11] Textacy, 2022. URL https://pypi.org/project/textacy/. Last visited: 2022-08-15.

[12] Mit tech review - diffbot, 2022. URL https://www.technologyreview.com/2020/09/04/1008156/knowledge-graph-ai-reads-web-machine-learning-natural-language-processing/. Last visited: 2022-08-15.

[13] Amit Singhal. Introducing the knowledge graph: Things, not strings, May 2012. URL https://www.blog.google/products/search/introducing-knowledge-graph-things-not/. Last visited: 2022-08-15.

[14] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutiérrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *CoRR*, abs/2003.02320, 2020. URL https://arxiv.org/abs/2003.02320.

[15] Renzo Angles and Claudio Gutierrez. Survey of graph database models. *ACM Comput. Surv.*, 40(1), feb 2008. ISSN 0360-0300. doi: 10.1145/1322432.1322433. URL https://doi-org.elib.tcd.ie/10.1145/1322432.1322433.

[16] Renzo Angles, Marcelo Arenas, Pablo Barceló, Aidan Hogan, Juan Reutter, and Domagoj Vrgoč. Foundations of modern query languages for graph databases. *ACM Comput. Surv.*, 50(5), sep 2017. ISSN 0360-0300. doi: 10.1145/3104031. URL https://doi.org/10.1145/3104031.

[17] Heiko Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8:489–508, 12 2016. doi: 10.3233/SW-160218.

[18] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *ACM Comput. Surv.*, 54(4), jul 2021. ISSN 0360-0300. doi: 10.1145/3447772. URL https://doi-org.elib.tcd.ie/10.1145/3447772.

[19] Semantic web - w3c, 2022. URL https://www.w3.org/standards/semanticweb/. Last visited: 2022-08-15.

[20] TIM BERNERS-LEE, JAMES HENDLER, and ORA LASSILA. The semantic web. *Scientific American*, 284(5):34–43, 2001. ISSN 00368733, 19467087. URL http://www.jstor.org/stable/26059207.

[21] Ian Horrocks. Semantic web: The story so far. In *Proceedings of the 2007 International Cross-Disciplinary Conference on Web Accessibility (W4A)*, W4A '07, page 120–125, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 1595935908. doi: 10.1145/1243441.1243469. URL https://doi-org.elib.tcd.ie/10.1145/1243441.1243469.

[22] Owl, Dec 2012. URL https://www.w3.org/OWL/. Last visited: 2022-08-15.

[23] Semantic web - w3c, 2022. URL https://www.w3.org/standards/semanticweb/data. Last visited: 2022-08-15.

[24] Tim Berners-Lee. Linked data, Jul 2006. URL https://www.w3.org/DesignIssues/LinkedData.html. Last visited: 2022-08-15.

[25] Shuyu Luo. Intro to recommender system: Collaborative filtering, 2022. URL https://towardsdatascience.com/intro-to-recommender-system-collaborative-filtering-64a238194a26. Last visited: 2022-08-15.

[26] Rodrigo Agerri and German Rigau. Robust multilingual named entity recognition with shallow semi-supervised features. *Journal of Artificial Intelligence*, Revised and Resubmitted.

[27] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, page 253–260, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 1581135610. doi: 10.1145/564376.564421. URL https://doi-org.elib.tcd.ie/10.1145/564376.564421.

[28] Yu Shao and Ying-hua Xie. Research on cold-start problem of collaborative filtering algorithm. In *Proceedings of the 2019 3rd International Conference on Big Data Research*, ICBDR 2019, page 67–71, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450372015. doi: 10.1145/3372454.3372470. URL https://doi-org.elib.tcd.ie/10.1145/3372454.3372470.

[29] Seongwon Jang, Simon Kim, and JeongWoo Ha. Graph-based recommendation systems : Comparison analysis between traditional clustering techniques and neural embedding. 2017.

[30] Annabelle Lim. Big five personality traits, 2022. URL https://www.simplypsychology.org/big-five-personality.html. Last visited: 2022-08-15.

[31] Zahid Halim and Aqsa Zouq. On identification of big-five personality traits through choice of images in a real-world setting. *Multimedia Tools and Applications*, 80(24): 33377–33408, Oct 2021. ISSN 1573-7721. doi: 10.1007/s11042-021-11419-5. URL https://doi.org/10.1007/s11042-021-11419-5.

[32] Wojciech Semik. Flask vs. django: Which python framework is better for your web development?, 2022. URL https://www.stxnext.com/blog/flask-vs-django-comparison/. Last visited: 2022-08-15.

[33] Allegrograph system properties - db engines, 2022. URL https://db-engines.com/en/system/AllegroGraph. Last visited: 2022-08-15.

[34] Dbpedia online access, 2022. URL http://wikidata.dbpedia.org/OnlineAccess#1.1%20Public%20SPARQL%20Endpoint. Last visited: 2022-08-15.

[35] Dbpedia sparql, 2022. URL https://www.dbpedia.org/resources/sparql. Last visited: 2022-08-15.

[36] Wikidata query service, 2022. URL https://query.wikidata.org/. Last visited: 2022-08-15.

[37] Andrea Wei-Ching Huang. A preliminary study on wikipedia, dbpedia and wikidata, 2015. URL http://andrea-index.blogspot.com/2015/06/wikipedia-dbpedia-wikidata.html. Last visited: 2022-08-15.

[38] Wikidata sparql tutorial, 2022. URL https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial. Last visited: 2022-08-15.

[39] Irish art, May 2022. URL https://en.wikipedia.org/wiki/Irish_art. Last visited: 2022-08-15.

[40] James joyce, Aug 2022. URL https://en.wikipedia.org/wiki/James_Joyce. Last visited: 2022-08-15.