An NLP-Based Scotch Whisky Recommender Agent

COMP3004 - Designing Intelligent Agents

Robert Soane

June 2021

Contents

1 Introduction

Scotch whisky has been produced as early as 1494 by distilling grains to produce a high proof spirit [1, 2]. According to the Scotch Whisky Association, prior to COVID-19 pandemc the industry "accounted for 75% of Scotland's food and drink exports", and had a year on year growth of 4.4% [3, 4]. With over 130 distilleries, each producing very different flavour profiles, choosing the next whisky to try could be challenging, enthusiast and beginner alike [5, 6].

Limited attempts have been made to apply AI methods to create recommender agents for whiskies. These focus on customer trend data (which this report argues is not the best strategy), or predominantly use discrete attributes about whiskies (distillery, ABV etc) on which to base their recommendations [7, 8].

This project sought to apply NLP techniques to produce a recommender agent and ascertain whether NLP techniques applied to whisky tasting notes can power an effective recommender agent.

In section 2 we discuss the production process of Scotch and it's lexicon. Section 3 covers relevant literature. Sections 4 & 5 describe the approach and implementation of the agent, and in section 6 the agent is evaluated via a questionnaire.

2 Scotch: Distilled

This project concerns itself with the specific domain of Scotch. To fully understand the problem at hand, a basic understanding of the drink is required. In this section I present a brief overview of Scotch, and a summary of its lexicon.

2.1 A brief overview of Scotch

Scotch whisky refers to whisky produced in Scotland fulfilling a set of legal requirements set by the UK Government [9].[1] Grains are allowed to malt (germinate) to develop their sugars, after which sugars are extracted producing a syrup, which is fermented producing a sweet hop-free beer. This is distilled in *pot stills* to increase the alcohol content making *new-make spirit*. This is matured in oak casks for a minimum of 3 years, prior to bottling. At this stage the distiller may choose to dilute the whisky to an ABV of no less than 40% [1, 2].

2.2 Whisky and words

The flavours present in Scotch come from a number of sources, in turn influencing how various whisky flavour profiles are described. To stop the grain germinating, it is heated. Some distilleries use a peat fire to do this. This imparts a smokey flavour onto the grain, which carries through to the end spirit. This smokey flavour is described as *peated* [1, 10].

Maturation provides another opportunity to add flavour to the drink. Age all Scotch in oak casks is resource intensive, and has lead to distilleries purchasing used casks from other drinks manufacturers. Traditionally the sherry industry has supplied used casks to distilleries. More recently, bourbon casks have been used. Any cask which has previously held any drink can be used, be it for the entire maturation process, or at the end such as a *sherry cask finish*. These all add their own flavours to the drink, which is reflected in tasting notes [1, 11].

[1] This report concerns Scotch whisky, and thus *Scotch* and *whisky* are used interchangeably.

3 Background and Literature

3.1 Language Models

In general, Natural Language Processing (NLP) tasks require a language model of some form or another. Artificial Intelligence (AI) based methods cannot process text in its native unstructured form, but need to convert the raw text to a structured form suitable for computer understanding. This is often referred to as *embedding*.

The two dominant model types are *syntactic* and *semantic* models. Syntactic models transform text to a set of 'symbols' which carry no inherent meaning, but can be compared across instances in a dataset, whereas semantic methods (such as those described in subsubsection 3.1.3) retain a contextual understanding of text [12].

A dominant syntactic method for transforming unstructured text into a computer-analysable form is the Bag-of-words (BoW) model. The dataset is tokenized (split into individual words), lemmatized (see subsubsection 3.1.1) and k keywords are extracted (see subsubsection 3.1.2) to form our bag of words $\underset{\sim}{b} \in \mathbb{R}^k$. Each document is transformed to a vector $\underset{\sim}{v} \in \mathbb{R}^k$ such that v_i is the frequency of the word b_i occuring in the document [12, 13, 14].[2]

3.1.1 Stemming and Lemmatization

When dealing with text data, it is not uncommon to have multiple forms of the same word. A syntactic model would view the words 'cat' and 'cats' as two different discrete symbols. A method is needed to reduce words to a normal form.

Porter [15] proposed an algorithm for removing word suffixes to aim for a normal form, this is called *stemming*. With no semantic understanding, the algorithm matches specific suffix patterns and removes them until it is unable to.

A more semantic approach would be *lemmatization*. Instead of algorithmically removing word endings, lemmatization normalises words to a real word root - the dictionary form of the word [16]. One lemmatizer implementation in Python is the WordNetLemmatizer in Python's Natural Language Tool Kit (NLTK), which queries the WordNet corpus to find the root word [13, 17].

3.1.2 Keyword Extraction

For syntactic methods, keyword extraction (KE) is key. For the purposes of this report, a keyword is a word of particular relevance or importance, and from which we might extract useful information. KE refers to strategies based on which those important words can be ranked, keeping the most relevant.

TF-IDF Onesuch method, is Term Frequency Inverse Document Frequency (TF-IDF). This is commonly used with BoW, and is implemented in Scikit-Learn [18]. TF-IDF is a statistic for scoring a words importance based on how frequently they occur in a document, and in the dataset [19].

Scoring as such aims to penalise words that occur too frequently across a document, boosting scores of words in an individual document which appear with disproportionately high frequency.

Graph based KE Another approach for KE is the use of graph-based ranking methods. These methods model words as nodes on a mathematical network graph.[3] A popular example is *Rapid Automatic Keyword Extraction* (RAKE), which finds a set of candidate keywords, and models them as a co-occurence graph. Each node represents a candidate, each edge co-occurence, and it's weight the number of co-occurences. Candidates are ranked according to frequency and degree [21].

[2] It is common to refer to an instance in a text dataset as a *document*

[3] A graph G being a set of nodes V and edges E. For a brief summary see Rashid Bin Muhammad's site http://personal.kent.edu/ rmuhamma/GraphTheory/MyGraphTheory/defEx.htm [20].

Beliga et al. [22], survey a wide range of graph based KE techniques, many of which rely on different centrality measures . Onesuch centrality measure is eigencentrality [23]. Eigencentrality scores each node as a proportion of the sum of the scores of all nodes to which it is connected. Suppose we have a graph, with an adjacency matrix A, we would set

$$x_i = \frac{1}{\lambda} A_{ij} x_j \tag{1}$$

using the summation convention, where x_i is the centrality of the i^{th} node. This reduces to the eigenvector equation

$$\mathbf{A} \cdot \underset{\sim}{x} = \lambda \underset{\sim}{x} \tag{2}$$

This is given with more detail in [24].

3.1.3 Word2vec

Word2vec is a semantic language model developed by Google. Instead of encoding each word as a discrete symbol as with BoW, word2vec embeddings retain similarity between similar words. This is achieved by training an *Artificial Neural Network* (ANN) to predict surrounding words for each given word. The hidden layer's weights represent probabilities of respective surrounding words. These probability vectors are used as embeddings for each word. As a words embedding now reflects likely surrounding words, synonyms are mapped to similar vectors. [25, 26, 27]

3.2 Recommender Agents

Collaborative filtering (CF) is perhaps the most common recommender engine method. CF treats each user as an entity, and provides recommendations to users based on the behaviours of users it considers similar [28, 29]. A simple, less abstract, example would be an online shopping site recommending product B to someone who has bought product A on the basis that a significant proportion of shoppers who buy product A go on to purchase B. It produces a simple filter, making predictions with no product knowledge, just user patterns.

Content based (CB) recommender engines are the opposite. Instead of focussing on user patterns they make predictions on the basis of specific attributes of each entity being recommended [28, 30]. While such a system may use user details, the main knowledge source is the entities being recommended.

3.3 AI applications to Whisky

There is a large gap in the research regarding Artificial Intelligence (AI) applications to whisky. Coldevin built an agent based whisky recommender, choosing to use a CB design [8]. He built a system using specific attributes about the whisky to recommend based on consumer likes or dislikes. Omid-Zohoor and Eghtesadi built another such hybrid (using both CB and CF) agent [7], however again they relied on specific categorical and ratio features. An interesting design choice was to recommend on the basis of a user's entire profile, and the ratings they give. Perhaps unsurprisingly their CB model performed poorly with users who gave large numbers of reviews. The more reviews the more noise.

I think an agent would be more effective if it takes specific user preferences at a given point in time, and makes a recommendation on that basis. Instead of attempting to offer users a whisky which best matches all varied whiskies they like, users get recommendations of the style they want at the time. This is closer to Coldevin's approach.

Wishart completed what seems to be the only study of whisky which uses NLP [31]. Working with industry experts he selected 84 different whiskies and extracted descriptors from their tasting notes.

These were coded and used to cluster the whiskies. These clusters were reviewed and evaluated by industry experts. He later proposed a set of 12 flavour dimensions for Scotch whisky [32].

While groundbreaking, Whisharts work differs somewhat from that carried out in this project. Where Wishart aimed to find key flavours in Scotch, working with industry experts, I aim to produce an agent which exhibits intelligence as an industry expert by suggesting whiskies, based only on their tasting notes.

4 Approach

4.1 Requirements

The following broad requirements were defined:

- **Agent and Environment** - The recommender should be an agent acting in an environment. The environment being the contents of the Master of Malt (MoM) website, and an interface with a user. The agent could be considered as part of a backend of a web app with outputs in a Python dictionary/JSON format.[4]
- **Speed** - The agent should be able to recommend within a couple of seconds.
- **Customisable** - An end user should be able to filter by price, volume and ABV.
- **Updateable** - The agent should be able to automatically update it's database, and retrain its models.
- **Input Types** - The agents should recommend based on *likes* & *dislikes* of whiskies supplied by a user, or from a user's written tasting notes.

4.2 The Data

In order to build a whisky specific language model (discussed further in subsection 4.4), a large corpus of tasting notes was required.

Product data for a large range of scotch whiskies was scraped from masterofmalt.com, this dataset contains a selection of attributes for each whisky.[5,6] Names and URLs are hashed together using MD5 to provide an ID.

It was observed that whiskies which are discontinued tend to be listed without a price, whereas those which are out of stock are listed with a price. For this reason, simplicity's sake, price was taken as an indication of stock. Those without a price were still recorded for two reasons; users may wish to make recommendations based on liking or disliking them, and they add to the corpus of tasting notes.

4.3 Choice of Recommender Method

While an online spirits shop may use CF recommenders to recommend products based on those being viewed (such as recommending gin to customers who like gin etc.), a CF system may fail for recommending whiskies based on tastes.

It is not unlikely that a whisky drinker may wish to buy two very different whiskies in one order, just to compare them, or they might enjoy a large range of whiskies. One might like a large variety of whiskies, and purchase many different styles frequently. It seems unlikely that the shopping habits of whisky enthusiasts is sufficient to recommend a whisky based on specific tastes.

For that reason, a CB recommender model was chosen, using tasting note data.

[4] **Note:** The webapp was beyond the scope of this project

[5] masterofmalt.com is a major UK whisky and spirits retailer.

[6] The scraping process is discussed in subsubsection 5.2.2

4.4 NLP Methods

Word2vec and BoW were both considered as language models. While there are many pretrained models available, these are likely unsuitable due to whisky's lexicon.

Word2vec encapsulates far more semantic data, however re-training word2vec regularly with new data would be expensive. As a quicker model to train BoW was chosen. TF-IDF, RAKE and an eigencentrality ranking measure discussed in subsubsection 4.4.3 were considered for KE.

4.4.1 The Ideal Vector and Similarity

BoW maps each input to a vector. To make recommendations, the agent must map user input to a vector in the same space as the BoW model. This *Ideal Vector* (IV) represents a hypothetical whisky which best represents the input. Cosine similarity can be used to ascertain which whiskies in the database best match the input. Cosine similarity indicates the angle between vectors [28]. As

$$\underset{\sim}{u} \cdot \underset{\sim}{v} \coloneqq |\underset{\sim}{u}||\underset{\sim}{v}| \cos\theta \tag{3}$$

for $\underset{\sim}{u}, \underset{\sim}{v} \in \mathbb{R}^k$, by keeping all vectors normalised, this reduces such that the cosine similarity of $\underset{\sim}{u}$ & $\underset{\sim}{v}$ is their scalar product.

Calculating cosine similarity for a large dataset is straightforward. Consider our dataset of m whiskies as

$$\mathbf{D} \in \mathbb{R}^{m \times n} \tag{4}$$

with each row representing the corresponding whisky's vector, and our IV $\underset{\sim}{v} \in \mathbb{R}^n$. The product of

$$\mathbf{D} \cdot \underset{\sim}{v} = \underset{\sim}{c} \tag{5}$$

or

$$\begin{pmatrix} d_{11} & d_{12} & ... & d_{1n} \\ d_{21} & d_{22} & ... & d_{2n} \\ ... & ... & \ddots & ... \\ d_{m1} & d_{m2} & ... & d_{mn} \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ ... \\ v_n \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ ... \\ c_n \end{pmatrix} \tag{6}$$

gives cosine similarities between IV and each whisky in D, where $\underset{\sim}{c}$ is our vector of cosine similarities.

4.4.2 Tasting Notes

As shown in table 1, whisky tasting notes are keyword dense.[7] Most words are candidate keywords, however some KE techniques (such as RAKE) aim to find keywords from far less keyword dense text. This must be considered when choosing a KE method.

4.4.3 Eigencentrality based Rapid Automatic Keyword Extraction (eRAKE)

As discussed in section 3.1.2, co-occurrence graphs can be useful for KE. RAKE is one such method where primitive centrality measures are used to rank nodes. Another method uses eigencentrality. This steps beyond words which themselves have a high co-occurrence and rewards words with significantly edges to words with high co-occurrences. This could be a compromise for retaining semantic data. While losing full semantic data we are at least stepping beyond merely looking at frequencies, selecting descriptors with larger amounts of influence across the dataset.

As the graph is undirected, our adjacency matrix is hermitian and thus finding it's eigenvectors is trivial, SciPy's *eigh*() function can be used [33, 34].

[7] All table data from masterofmalt.com

Table 1: A selection of whisky tasting notes from Master of Malt

Whisky	Tasting Notes
Laphroaig 10 Year Old	**Nose:** *"This opens on big, smoky muscular peat notes. There are spices, and liquorice, as well as a big dose of salt. This whisky has become slightly sweeter in recent years, and it appears beautifully on the nose, amidst the classic iodine/sticking plasters and cool wood smoke we love."* **Palate:** *"Seaweed-led, with a hint of vanilla ice cream and more than a whiff of notes from the first aid box (TCP, plasters etc). The oak is big, and muscles its way into the fore as you hold this whisky over your tongue. An upsurge of spices develop – cardamom/black pepper/chilli."* **Finish:** *"Big and drying, as the savoury, tarry notes build up with an iodine complexity."*
Talisker 10 Year Old	**Nose:** *"A fresh and fragrant nose. Through thick, pungent smoke comes sweet pear and apple peels, with pinches of maritime salt from kippers, seaweed."* **Palate:** *"It's a bonfire of peat crackling with black pepper, with a touch of brine and dry barley. A welcome delivery of orchard fruit provides a delicate and beautiful balance."* **Finish:** *"In a long finish, bonfire embers toast malt and crystallise a sugary underlay"*

5 Implementation

Implementation was split into the following two broad phases:

- **Data exploration:** Exploring the dataset, and potential prototypical methods in a Jupyter Notebook [35]. Choosing a method to implement.
- **Agent design:** Designing an agent in Python based on work from previous phase.

These are discussed in sections 5.1 & 5.2 respectively.

5.1 Data Exploration

5.1.1 Effective Lemmatizing

Given whisky's specific lexicon, an interesting problem occurs when trying to use gerneral purpose lemmatizers. In whisky 'peated' is a verb describing how the grain was processed, and thus the 'peaty' (smokey) flavour has been imparted on the finished spirit. In English, peat refers to a natural fuel made from dead plant matter, and 'peated' does not exist [36]. 'Peated' and 'peaty' should both reduce to 'peat', however as they are not considered as verbs in WordNet. For this reason a separate custom whisky lemmatizer was built.

The *WhiskyLemmatizer* was built on top of scikit-learn's WordNet lemmatizer [18]. I manually created a dictionary of whisky-specific words and corresponding root forms. For each input, the lemmatizer first checks the dictionary. If the word is not in the dictionary it then uses scikit-learn's lemmatizer. As WordNet can be slow, every result from WordNet is cached in the dictionary.

After experimentation with this WhiskyLemmatizer, words in the cache which are mapped to words which further reduce were automatically updated to map to the leaf word. A set of stopwords was manually produced in an iterative process based on the lemmatizer's outputs.

Table 2: Times of TF-IDF, RAKE and eRAKE with various lemmatizers in seconds.

	TF-IDF	RAKE	eRAKE
Unlemmatized	*1247*	*0.441*	-
WordNet Lemmatized	*1461*	*130.2*	-
WhiskyLemmatizer	*1209*	*4.947*	*47.6*

Note: eRAKE was only applied to the WhiskyLemmatized corpus as the eRake implementation included the WhiskyLemmatizer.

5.1.2 Comparing KE Strategies

As over time the lexicon may change, the agent should perform KE with each training cycle. KE takes a significant proportion of time for model building. For this reason time and accuracy are of equal importance. Perfect KE is of little use if it takes forever.

An adapted implementation of TF-IDF [37] the *rake-nltk* python package [38], and a new implementation eRAKE were applied to the dataset with a range of lemmatizers to extract the top 300 keywords. The methods were timed, and the top 20 keywords recorded. These can be found in tables 2 and 3.

As is clear, TF-IDF KE took orders of magnitude longer than RAKE and eRAKE. Qualitatively evaluating the keywords extracted by RAKE vs eRAKE, eRAKE produces more useful keywords. This is perhaps unsurprising, as RAKE aims to find keywords from a corpus with both a relatively high frequency of each keyword, and a higher frequency of stopwords. By applying it to tasting notes it is perhaps being misused. As highlighted in subsubsection 4.4.2, tasting notes are very feature dense, however different tasting notes have different features characterising them. It is likely that the most frequent features are penalised due to stopword potential. It is interesting to see that WordNet lemmatizing had little impact in terms of which words were extracted.

Table 3: Top 20 keywords from each of TF-IDF, RAKE and eRAKE with various lemmatizers.

Keyword Extraction	Lemmatizer	Keywords
TF-IDF	None	*vanilla, quite, juicy, jam, zest, liquorice, crème, waxy, mixed, oak, zesty, smoke, marzipan, drizzle, hazelnut, beeswax, joined, juice, brûlée, box*
	Wordnet	*vanilla, quite, juicy, jam, zest, liquorice, crème, waxy, mixed, oak, zesty, smoke, marzipan, drizzle, hazelnut, beeswax, joined, juice, brûlée, box*
	Whisky	*vanilla, zest, jam, quite, juicy, sweet, fruit, waxy, liquorice, crème, smoke, develop, oak, mixed, drizzle, hazelnut, marzipan, join, dry, beeswax*
RAKE	None	*with, winesky, while, touch, torten, time, theres, saucepan, salty, pan, or, nose, musty, muscular, more, marketplace, little, like, just, its*
	Wordnet	*with, winesky, while, touch, torten, time, theres, saucepan, salty, pan, or, nose, musty, muscular, more, marketplace, little, like, just, its*
	Whisky	-
eRAKE	Whisky	*fruit, sweet, spice, oak, vanilla, smoke, honey, malt, chocolate, apple, dry, pepper, orange, cream, butter, fresh, nut, peel, rich, barley*

Table 4: Whiskies considered in clustering evaluation.

Highland Park 12 Year Old, Bowmore 15 Year Old, Arran 10 Year Old, Edradour 10 Year Old, Old Pulteney 12 Year OId, Laphroaig 10 Year Old, Ardbeg 10 Year Old, Blair Athol 12 Year Old - Flora and Fauna, Talisker 10 Year Old, GlenAllachie 15 Year Old, Aberlour A'Bunadh Batch 68

5.1.3 Clustering

Table 5: Clustering of whiskies from each BoW model.

KE	Cluster Features	Whiskies	KE	Cluster Features	Whiskies
TF-IDF	*spice, vanilla, sweet*	Highland Park, Bowmore	TF-IDF*	*fruit, malt, spice*	Highland Park, Bowmore
	malt, honey, sweet	Arran		*malt, fruit, oak*	Arran
	fruit, malt, spice	Edradour		*fruit, malt, spice*	-
	oak, malt, vanilla	Old Pulteney		*oak, malt, vanilla*	Old Pulteney
	smoke, peat, malt	Laphroaig, Ardbeg, Blair Athol, Talisker		*smoke, peat, malt*	Laphroaig, Ardbeg, Talisker
	chocolate, malt, sherry	GlenAllachie, Aberlour A'Bunadh		*sherry, malt, fruit*	GlenAllachie, Blair Athol, Aberlour A'Bunadh, Edradour
RAKE	*musty, muscular, fire*	Laphroaig, Ardbeg, Highland Park, Old Pulteney, GlenAllachie, Blair Athol, Bowmore, Aberlour A'Bunadh, Edradour	eRAKE	*malt, vanilla, sweet*	Highland Park, Bowmore
	little, musty, fire	-		*sherry, malt, fruit*	GlenAllachie, Blair Athol, Aberlour A'Bunadh, Edradour
	little, like, musty	-		*malt, fruit, oak*	Arran
	time, like, fire	Talisker		*fruit, malt, spice*	-
	like, muscular, musty	Arran		*smoke, peat, vanilla*	Laphroaig, Ardbeg, Talisker
	little, time, like	-		*oak, malt, vanilla*	Old Pulteney

TF-IDF and RAKE refer to these extractions applied both WordNet and unlemmatized. They produced the same results. TF-IDF* is TF-IDF used with WhiskyLemmatizer

Cluster Features refers to the three most prominent features at the centers of the cluster.

For the purpose of sanity checking, and ensuring sufficient information is retained in each BoW, k-means clustering was applied on a BoW model based on 300 keywords from each KE. The clusters of whiskies in Table 4 were considered, the corresponding clusters are shown in Table 5.[8]

There is little difference between TF-IDF and TF-IDF*, apart from Edradour being clustered with

[8]I chose these whiskies as I have enough basic knowledge of them to qualitatively approximately evaluate the sensibleness of the clustering and help make a decision. This isn't a rigorous approach, and future work would need a more rigorous evaluation at this stage. This approach was used due to time constraints.

GlenAllachie and Aberlour (two heavily sherried expressions) in TF-IDF*, and Blair Athol. Blair Athol is a relatively sherried expression, and thus appearing in a peat heavy cluster (Laphroaig, Ardbeg, Talisker) seems strange. It's TF-IDF* placement seems far more sensible.

When considering tasting notes, Blair Athol is described as *"Nutty with sherried notes. Gentle peat. Crisp. ... Peat smoke, syrup ..."* [39]. This highlights a limitation of BoW and its applications to this problem. By three mentions of peat and smoke, Blair Athol was grouped with peat and smoke heavy whiskyies.

RAKE's clusters are clearly nonsense, however this isn't surprising considering it's main keywords. The eRAKE clusters are very similar to those in TF-IDF*. On the basis of this, and the data in subsubsection 5.1.2, eRAKE was chosen to move forward with.

5.2 Agent Design

The agent was designed as a single Python class, with a few helper classes and functions. All inputs/outputs use Python dictionaries. While this may seem strange, this is with the view that a web frontend could be designed to send data in JSON formats.

5.2.1 The Database

An SQLite database is used to store all whisky data and models. This is included in Python and allows multiple agents to access the data at the same time. This also allows one agent to update the models and all other agents will use the up to date model. It was observed that SQLite runs faster than Pandas [40], however Pandas is still used for some manipulations once data is loaded from the database.

When loaded, the agent checks if there's a database. If there isn't, it is built from the pre-existing *scotch.csv* file.

5.2.2 Web Scraping

The initial dataset was collected using a rough script using Python and *Beautiful Soup* [41]. The agent was designed using this dataset of ∼14,000 whiskies. As per the requirements, and to ensure the agent's autonomy, a method was written to fetch new whiskies. If the initial dataset is not present, the agent will automatically fetch all data.

The 'new whiskies' page on MoM's website is parsed and each listing is checked to confirm it is Scotch. When each ID is created, it is checked against those already in the database. If three consecutive listings are already in the database, the agent stops, assuming all new products have been included. Sometimes one or two re-stocked whiskies are listed in succession. Setting three as the threshold reduces the risk of stopping prematurely.

While this function could be set to run periodically, this hasn't been implemented to avoid unnecessarily using MoM's server.

5.2.3 Model Training

eRAKE is used to lemmatize all tasting notes, and extract 300 keywords for each model. These are then vectorised and the dataset is transformed to a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, with each row a normalised vector on a 300-dimensional hypersphere. Each $\mathbf{M}$ is stored with each whisky's corresponding ID. Four models are created, described in Table 6.

5.2.4 Producing Recommendations

As discussed in section 4.4.1, cosine similarity is used to provide recommendations. Where recommendations are made based on more than one model, the mean of the similarities is used. As Pandas can

Table 6: Description of models created for agent.

Model	Description
Nose	Model based purely on keywords extracted from *nose* tasting notes
Palate	Model based purely on keywords extracted from *palate* tasting notes
Finish	Model based purely on keywords extracted from *finish* tasting notes
General	Model based on keywords extracted from all tasting notes. Vectorising description as well as tasting notes for each whisky. This reflects some whiskies being listed without tasting notes, but taste indications in main description.

be quite slow due to its single-threaded nature, especially when converting a long list of entries into a dataframe. To minimise this effect, the user's filtering input (price, ABV etc.) is used to get the set of all IDs from which a recommendation can be made. Only these whiskies are queried from the database.

5.3 User Reviews

The agent accepts user reviews which can be incorporated when training. Acknowledging that expert notes may be better, only the tasting notes are used for KE. When including reviews, the IV is calculated by

$$\underset{\sim}{IV} = |\underset{\sim}{t} + \underset{\sim}{r} \cdot \min(\frac{n}{W}, 1)| \tag{7}$$

where t and r are the vectorised tasting notes and reviews, and n and W are the number of reviews and minimum weight for the tasting notes.

5.3.1 Dream Dram

An interesting feature implemented is the *Dream Dram* recommender. This takes unstructured text describing a 'dream' whisky and makes a recommendation on that basis. This works in much the same way as the other recommendations, however the IV is generated on the basis of the text input instead of by querying for specific whiskies. This option could be incorporated into a Whisky chat bot at a later date.

6 Evaluation

6.1 Survey

To evaluate the performance of the agent, a sample dataset of potential user inputs and agent outputs was produced. A description of each input can be found in Table 7. Random sampling was used to produce baseline recommendations. These were used in a survey which 30 whisky enthusiasts completed.[9]. They were asked to rate each recommendation on the basis of the corresponding input. While the agent recommends 10 whiskies by default, only 3 recommendations were given from the baseline and agent to avoid making the survey too cumbersome.

Participants were given all information available to the agent for each output including description and tasting notes (and some information the agent does not use such as ABV, Price etc), however they were instructed not to factor price in their ratings as price parameters were set by the hypothetical user. There was an option to leave text feedback.

[9]See section 8 regarding ethics.

Table 7: Description of inputs in evaluation dataset.

Input Reference	Rationale
ATN1	Replicating a user who has tried and developed tastes for a variety of whiskies available at supermarkets, but hasn't tried much beyond.
ATN2	Replicating a significant partiality towards heavily peated whiskies.
ATN3	Replicating an enjoyment of both peated and sherried whiskies.
ATN4	A user with niche and specific whisky tastes.
GC	Producing recommendations based on general inputs without considering specific tasting notes.
DD1	Dream Dram recommendation from a very peat heavy input.
DD2	Dream Dram recommendation describing a very oily and fruity whisky.

6.2 Results

As can be seen in figures 1 and 2, most recommendations were rated higher than the mean baseline.

A paired one-tailed t-test was conducted on the baseline scores and corresponding recommender scores for each sample input. There was a significant increase in scores from the recommender ($M = 6.28$, $SD = 1.13$) compared with the baseline ($M = 4.71$, $SD = 1.15$), $t(12) = 2.22$, $p < 0.05$. There was a mean increase of 1.57 points.

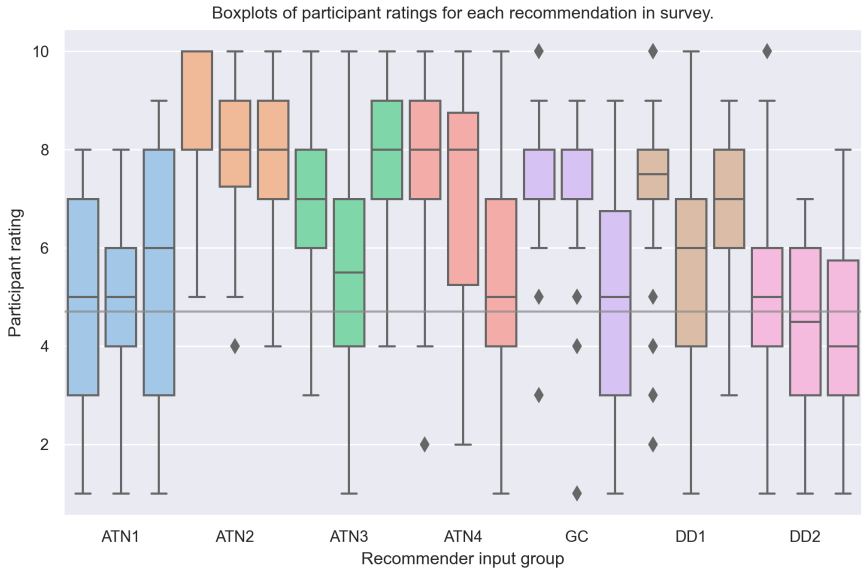

Figure 1: Boxplots of participant ratings for each recommendation, grouped by sample input. The grey line indicates the mean baseline rating.

6.3 Discussion

Despite promising results, it is clear that some recommendations received a large range of ratings. The small samples used, both of enthusiasts and inputs/recommendations, are far from ideal.

On reading the free-text responses, it's clear that some participants weren't confident in their whisky knowledge. Some participants hadn't followed the instructions, for example one participant stated they

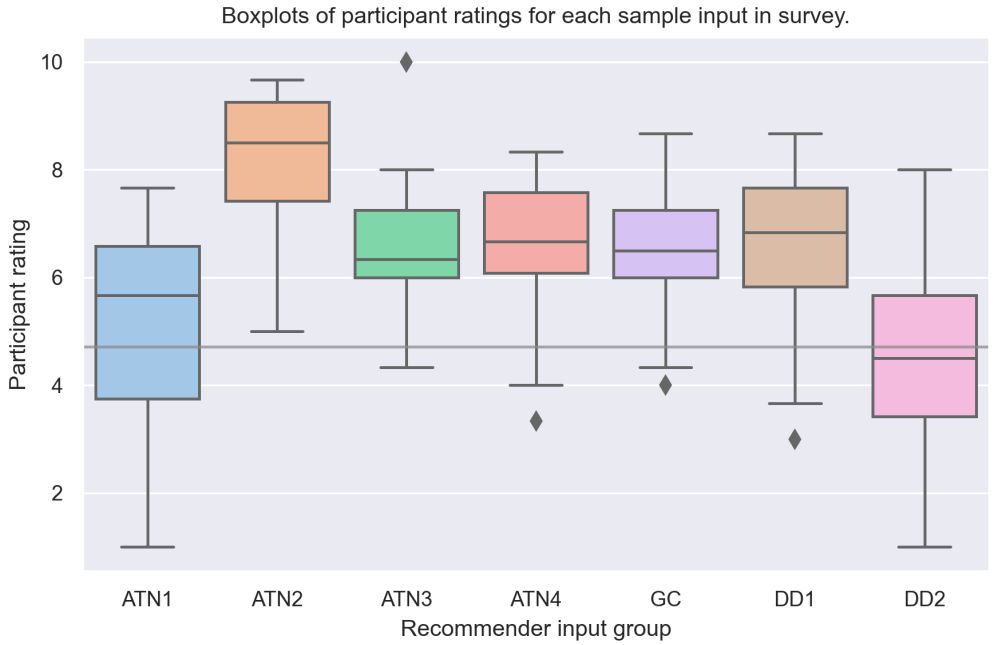

Figure 2: Boxplots of participant ratings for each sample input. The grey line indicates the mean baseline rating.

had penalised recommendations with high prices.

A few interesting points were made which are worth considering. One participant suggested that the input for ATN1 was contradictory, perhaps explaining its low score. This suggests that (as intended) this recommender method is not ideal for taking a users entire taste profile, rather recommending a whisky to try based on similarity between a small number of tried whiskies. It would be interesting to investigate whether such a system is possible in the specific case of whisky.

Many users pointed out that many recommendations were very niche, suggesting an extra filter could be incorporated allowing a user to filter out single-cask/independent bottles. This also could have impacted the accuracy of participant's scores.

7 Conclusion

This project sought to investigate whether NLP could be applied to free text tasting notes to produce an effective Scotch recommender agent.

Acting on an environment consisting of a retailers website, an autonomous agent was designed with capacity to maintain an updated language model for Scotch whisky. This model can reflect changes in the Scotch lexicon, and be applied effectively to recommend whisky on the basis of taste. This could have significant positive consequences for whisky buyers.

Due to the sparsity of the field, there is a wealth of space for exploration, with a few suggestions listed in subsection 7.1. Despite promising early results, a more comprehensive study is needed with larger samples and/or industry experts.

7.1 Suggestions for Future Work

The following suggestions are made for future work and research to build upon this project.

- Development of a front end UI.

- Further comparisons and research into KE methods for whisky tasting notes.
- Investigations into semantic language models and alternative similarity measures.
- Work on whisky clustering to research flavour profile metrics.
- Further evaluation of the agent.
- Work with industry experts to evaluate and improve the agent.

8 Statement on Ethics

I have followed the guidance given in the module about research ethics in the user study in my coursework for this module.

References

[1] K. Jacques, T. Lyons, and D. Kelsall, *The Alcohol Textbook 4th edition*, 4th ed. Nottingham University Press, 2003.

[2] M. Pyke, "THE MANUFACTURE OF SCOTCH GRAIN WHISKY," *The Distillers Company Ltd., Glenochil Research Station, Menstrie, Clackmannanshirc, Scotland)*, vol. 71, pp. 209–218, 1965.

[3] "Facts & figures." [Online]. Available: https://www.scotch-whisky.org.uk/insights/facts-figures/

[4] "Scotch whisky export figures 2019." [Online]. Available: https://www.scotch-whisky.org.uk/newsroom/scotch-whisky-exports-surge-amidst-backdrop-of-tariff-uncertainty/

[5] "How many whisky distilleries are in scotland?" Oct 2020. [Online]. Available: https://whiskytastingcompany.com/blogs/news/how-many-whisky-distilleries-are-in-scotland

[6] T. Powell, "The beginner's guide to scotch whisky," Jan 2021. [Online]. Available: https://foodism.co.uk/guides/scotch-whisky-regions-guide/

[7] A. Omid-zohoor and A. Eghtesadi, "Whisky Recommender."

[8] T. M. Coldevin, "Building an Multi-Agent Whisky Recommender System," no. February, 2005.

[9] "The scotch whisky regulations 2009," *Legislation.gov.uk*, 2009. [Online]. Available: https://www.legislation.gov.uk/uksi/2009/2890/contents/made

[10] G. N. Bathgate, "The influence of malt and wort processing on spirit character: the lost styles of Scotch malt whisky," *Journal of the Institute of Brewing*, vol. 125, no. 2, pp. 200–213, 2019.

[11] J. Mosedale and J.-L. Puech, "Wood maturation of distilled beverages," *Trends in Food Science & Technology*, vol. 9, no. 3, pp. 95–101, mar 1998. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0924224498000247

[12] E. Cambria and B. White, "Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]," *IEEE Computational Intelligence Magazine*, vol. 9, no. 2, pp. 48–57, may 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6786458/

[13] E. L. Steven Bird, Ewan Klein, *Natural Language Processing with Python*. O'Reilly Media Inc, 2009.

[14] Y. Zhang, R. Jin, and Z. H. Zhou, "Understanding bag-of-words model: A statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1-4, pp. 43–52, 2010.

[15] M. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, mar 1980. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/eb046814/full/html

[16] K. Jayakodi, M. Bandara, I. Perera, and D. Meedeniya, "WordNet and cosine similarity based classifier of exam questions using bloom's taxonomy," *International Journal of Emerging Technologies in Learning*, vol. 11, no. 4, pp. 142–149, 2016.

[17] "What is wordnet?" 2010. [Online]. Available: https://wordnet.princeton.edu/

[18] P. Fabian, G. Varoquaux, A. Gramfort, M. Vincent, B. Thririon, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[19] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," *Proceedings of the first instructional conference on machine learning*, vol. 242, no. 1, pp. 29–48, 2003.

[20] R. B. Muhammad, "Graph theory: Definitions and examples." [Online]. Available: http://personal.kent.edu/~rmuhamma/GraphTheory/MyGraphTheory/defEx.htm

[21] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic keyword extraction," *Text Mining: Applications and Theory*, pp. 1—-277, 2010.

[22] S. Beliga, A. Mestrovic, and S. Martincic-Ipsic, "An Overview of Graph-Based Keyword Extraction Methods and Approaches," *Journal of Information and Organizational Sciences*, vol. 39, no. 1, 2015. [Online]. Available: https://hrcak.srce.hr/140857

[23] P. Bonacich, "Some unique properties of eigenvector centrality," *Social Networks*, vol. 29, no. 4, pp. 555–564, oct 2007. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0378873307000342

[24] M. E. J. Newman, "Mathematics of networks," *Networks*, pp. 109–167, 2010.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 2013, pp. 1–12.

[26] C. McCormick, "Word2Vec Tutorial - The Skip-Gram Model," pp. 1–39, 2017.

[27] Q. Liu, M. J. Kusner, and P. Blunsom, "A Survey on Contextual Embeddings," 2020. [Online]. Available: http://arxiv.org/abs/2003.07278

[28] P. Melville and V. Sindhwani, "Encyclopaedia of Machine Learning: Recommender Systems," *Encyclopaedia of Machine Learning*, pp. 829–838, 2010. [Online]. Available: https://link.springer.com/content/pdf/10.1007/978-1-4899-7687-1_964.pdf%0Ahttps://link.springer.com/referenceworkentry/10.1007%2F978-0-387-30164-8_705%0Ahttps://link.springer.com/book/10.1007/978-0-387-30164-8%0A%0Ahttp://vikas.sindhwani.org/recommender.p

[29] J. Herlocker, J. Konstan, and J. Reidl, "Explaining Collaborative Filtering Recommendations," Mineapolis, 2000. [Online]. Available: http://www.grouplens.org

[30] R. J. Mooney and L. Roy, "Content-based book recommending using learning for text categorization," *Proceedings of the ACM International Conference on Digital Libraries*, pp. 195–204, 2000.

[31] D. Wishart, "Classification of Single Malt Whiskies," 2000, pp. 89–94. [Online]. Available: http://link.springer.com/10.1007/978-3-642-59789-3{_}14

[32] ——, "The flavour of whisky," *Significance*, vol. 6, no. 1, pp. 20–26, mar 2009. [Online]. Available: http://doi.wiley.com/10.1111/j.1740-9713.2009.00337.x

[33] M. Hubbard, "Lecture: Best approximation (2)/eigenvalues and eigenvectors (1), 17/03/2020," *MATH3036-1-UNUK-SPR-1920 Scientific Computation and Numerical Analysis*, Mar 2020.

[34] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, p. 357–362, 2020.

[35] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, "Jupyter notebooks – a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Schmidt, Eds. IOS Press, 2016, pp. 87 – 90.

[36] L. Jenner, "peat," 2019.

[37] P. Prakash, "Keyword extraction: Keyword extraction in python," Dec 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/11/words-that-matter-a-simple-guide-to-keyword-extraction-in-python/

[38] V. B. Sharmer, "rake-nltk," Jun 2018. [Online]. Available: https://pypi.org/project/rake-nltk/

[39] "Blair athol 12 year old - flora and fauna." [Online]. Available: https://www.masterofmalt.com/whiskies/blair-athol-12-year-old-whisky/

[40] T. pandas development team, "pandas-dev/pandas: Pandas," Feb. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3509134

[41] L. Richardson, "Beautiful soup documentation," *April*, 2007.