

# An NLP-Based Scotch Whisky Recommender Engine

COMP3004 - Designing Intelligent Agents

Robert Soane

June 2021

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Scotch: Distilled</b>	<b>2</b>
2.1	A brief overview of Scotch . . . . .	2
2.2	Whisky and words . . . . .	2
<b>3</b>	<b>Background and Literature</b>	<b>2</b>
3.1	Language Models . . . . .	2
3.1.1	Stemming and Lemmatization . . . . .	3
3.1.2	Keyword Extraction . . . . .	3
3.1.3	Word2vec . . . . .	4
3.2	Recommender engines . . . . .	4
3.3	Machine learning applications to Whisky . . . . .	4
<b>4</b>	<b>Approach</b>	<b>4</b>
4.1	Requirements . . . . .	4
4.2	The Data . . . . .	4
4.3	NLP Methods . . . . .	5
4.3.1	The Ideal Vector and Similarity . . . . .	5
<b>5</b>	<b>Implementation</b>	<b>5</b>
5.1	Data Exploration . . . . .	6
5.1.1	Whisky Words . . . . .	6
5.1.2	Model Building . . . . .	6
5.1.3	Clustering . . . . .	6
5.2	Agent Design . . . . .	6
5.2.1	Web Scraping . . . . .	6
5.2.2	Model Training . . . . .	6
5.2.3	Recommendations . . . . .	6
5.2.4	Speed . . . . .	6
5.2.5	Dream Dram . . . . .	6
<b>6</b>	<b>Results</b>	<b>6</b>
<b>7</b>	<b>Discussion</b>	<b>6</b>

## 1 Introduction

## 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. For this reason, in this section I present a brief overview of Scotch, and a summary of its lexicon. From thereon in whisky-specific terms can be used without explanation.

### 2.1 A brief overview of Scotch

Distilled spirits have been in production as early as 1310 in France, with fruit wines being fortified by distillation to make brandy. The earliest records of Scotch whisky dating back to 1494. Where brandy is a distillation of fermented fruit juices (wines), whisky is produced by distilling fermented grains [1, 2].

Scotch whisky refers to whisky produced in Scotland fulfilling a set of legal requirements set by the UK Government [3]<sup>1</sup>. Grains are allowed to malt (germinate) to develop their sugars, after which the sugars are extracted to produce a syrup called *wort*. The wort is fermented to produce a sweet hop-free beer. The beer is distilled in *pot stills* to increase the alcohol content significantly to produce *new-make spirit*. This is matured in oak casks for a minimum of 3 years, before the whisky is bottled. 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

When describing whisky, there two important distinctions to be made. Malt versus grain, and single versus blended. *Malt* refers to a whisky wherein the only grain used is *malted barley*, whereas *grain* whisky can be made with mixtures of any grains [4]. *Single* refers to whisky where all whisky in the bottle has come from the same distillery, whereas *blended* whisky is a blend from any number of distilleries. It is important to note that single whiskies are usually themselves blends, but blends of casks from the same distillery [5].

The flavours present in Scotch come from a number of places, and this in turn influences how various whisky's flavour profiles are described. A descriptor often used to describe whiskies is *peat*. To stop the malting process, the grain is heated, some distilleries (particularly those on the island of Islay) use a peat fire to carry this out. This imparts a smokey flavour onto the grain, which carries through to the end spirit. This smokey flavour is often described as *peated* [1, 6].

The maturation process provides another opportunity to add flavour to the drink. The requirement to 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, and 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, and this is reflected in whisky tasting notes [1, 7].

## 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

---

<sup>1</sup>This report concerns Scotch whisky, and thus *Scotch* and *whisky* are used interchangeably.

form, but need to convert the raw text to a structured form suitable for the computer to understand. This is often referred to as *embedding*.

The two predominant model types are *syntactic* and *semantic* models. Syntactic methods 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 subsection 3.1.3) retain a general understanding of the text [8].

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 subsection 3.1.1) and  $k$  keywords are extracted (see subsection 3.1.2) to form our bag of words  $\underline{b} \in \mathbb{R}^k$ . Each document<sup>2</sup> is transformed to a vector  $\underline{v} \in \mathbb{R}^k$  such that  $v_i$  is the frequency of the word  $b_i$  occurring in the document [8, 9, 10].

### 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 proposed an algorithm for removing word suffixes to aim for a normal form, this is called *stemming*. With no semantic understanding, the algorithm searches for specific suffix patterns and removes them until it is unable to [11].

A more semantic approach would be *lemmatization*. Instead of algorithmically removing word endings lemmatization aims to normalise words to a real word root, that is a lemmatizer would reduce the word to the dictionary form of the word [12]. A lemmatizer implementation in Python is the WordNetLemmatizer in the Python Natural Language Tool Kit (NLTK), which queries the WordNet corpus to find the root word [9, 13].

### 3.1.2 Keyword Extraction

For syntactic methods, keyword extraction 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. Keyword extraction refers to strategies based on which those important words can be ranked, and only the most relevant kept.

**TF-IDF** One such method, is Term Frequency Inverse Document Frequency (TF-IDF). This is commonly used with BoW, and is implemented in Scikit-Learn [14]. TF-IDF is a statistic for scoring a word’s importance based on how frequently it occurs in a document, and how frequently it occurs in the dataset [15].

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

**Graph based keyword extraction** Another approach for keyword extraction is the use of graph-based ranking methods. These methods model words as nodes on a mathematical network graph<sup>3</sup>. A popular example is the *Rapid Automatic Keyword Extraction* (RAKE) algorithm, which splits finds a set of candidate keywords, and models them as a co-occurrence graph.

Each node represents a candidate, each edge co-occurrence, and its weight the number of co-occurrences. The candidates are then ranked according to frequency and degree (sum of weights) [17].

Beliga et al., survey a wide range of graph based keyword extraction techniques, many of which rely on different centrality measures [18]. One such centrality measure which may be useful for this problem

---

<sup>2</sup>It is common to refer to an instance in a text dataset as a *document*

<sup>3</sup>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> [16].

is eigencentality [19]. Essentially, eigencentality aims to assign each node as a proportion of the sum of all nodes to which it is connected. Suppose we have a graph, with an adjacency matrix  $A$ , with  $x_i$  being the centrality of the  $i^{th}$  node, we would set  $x_i = \frac{1}{\lambda} A_{ij} x_j$  (using the summation convention). This reduces to the eigenvector equation  $\mathbf{A} \cdot \underline{x} = \lambda \underline{x}$ . This is given with more detail in [20].

### 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 the surrounding words for any given word. The weights of the hidden layer represent probabilities of respective surrounding words. These probability vectors are used as embeddings for each word. As a words embedding now reflects the likely surrounding words, synonyms are mapped to similar vectors. [21, 22, 23]

## 3.2 Recommender engines

There are two main classes of recommender engine described in this section.

### Collaborative Filtering

### Content Based Recommenders

## 3.3 Machine learning applications to Whisky

# 4 Approach

## 4.1 Requirements

The following broad requirements were set out for the agent:

- **Backend** - The agent should be considered as part of a backend of a web app<sup>4</sup>. Relevant inputs and outputs should be in a python dictionary/JSON format.
- **Speed** - The agent should be able to produce a recommendation 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 to include new whiskies, and retrain its language model.
- **Input Types** - The agents should recommend based on *likes & dislikes* of whiskies supplied by a user, or from a users written tasting notes.

## 4.2 The Data

In order to build a whisky specific language model (discussed further in subsection 4.3), a large corpus of tasting notes was required. This could also fulfill the requirements of a dataset from which to make predictions.

Product data for a large range of scotch whiskies was scraped from masterofmalt.com<sup>5</sup>, this dataset contains a selection of attributes for each whisky. It was observed that whiskies which are discontinued tend to be listed without a price, whereas those which are merely out of stock are listed with a price.

<sup>4</sup>**Note:** The webapp was beyond the scope of this project

<sup>5</sup>masterofmalt.com is a major UK whisky and spirits retailer.

For this reason, and for the sake of simplicity, price was taken as an indication of whether a whisky is likely to be in stock. Those without a price were still recorded in the dataset 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 NLP Methods

Word2vec and BoW were both considered as candidate language models. While there are many pretrained models available, these are likely to be unsuitable as briefly explained in subsection 5.1.1.

Word2vec encapsulates a far greater amount of 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 eigencentality ranking measure (which will be referred to as eRAKE) were considered for keyword extraction.

#### 4.3.1 The Ideal Vector and Similarity

The BoW model maps each input to a vector. To make recommendations, the agent must map the 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 user’s input. Cosine similarity can then be used to ascertain which whiskies in the database best match the input. Cosine similarity indicates the angle between vectors [24]. As for  $\underline{u}, \underline{v} \in \mathbb{R}^k$ ,  $\underline{u} \cdot \underline{v} := |\underline{u}||\underline{v}| \cos \theta$  by storing all vectors normalised, this reduces such that the cosine similarity of  $\underline{u}$  &  $\underline{v}$  is simply their scalar product.

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

Calculating cosine similarity for a large dataset is relatively straightforward, and this method was implemented in the agent. Consider our dataset of  $m$  whiskies as a matrix  $\mathbf{D} \in \mathbb{R}^{m \times n}$  with each row representing the corresponding whisky’s vector, and our IV  $\underline{v} \in \mathbb{R}^n$ . As demonstrated in equation 1, the product of  $\mathbf{D} \cdot \underline{v} = \underline{c}$  where  $\underline{c}$  is our vector of cosine similarities.

## 5 Implementation

The implementation was split into two broad phases, *data exploration* and *agent design*. These are discussed in sections 5.1 & 5.2 respectively.

## 5.1 Data Exploration

### 5.1.1 Whisky Words

### 5.1.2 Model Building

### 5.1.3 Clustering

## 5.2 Agent Design

### 5.2.1 Web Scraping

### 5.2.2 Model Training

### 5.2.3 Recommendations

### 5.2.4 Speed

### 5.2.5 Dream Dram

## 6 Results

## 7 Discussion

## 8 Conclusion

## 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, Clackmannanshire, Scotland*), vol. 71, pp. 209–218, 1965.
- [3] “The scotch whisky regulations 2009,” *Legislation.gov.uk*, 2009. [Online]. Available: <https://www.legislation.gov.uk/ukxi/2009/2890/contents/made>
- [4] P. Valaer, “Scotch Whisky,” *Industrial and Engineering Chemistry*, vol. 32, no. 7, pp. 935–943, 1940.
- [5] B. C. Smith, C. Sester, J. Ballester, and O. Deroy, “The perceptual categorisation of blended and single malt Scotch whiskies,” *Flavour*, vol. 6, no. 1, pp. 1–9, 2017.
- [6] 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.
- [7] 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>
- [8] 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/>
- [9] E. L. Steven Bird, Ewan Klein, *Natural Language Processing with Python*. O’Reilly Media Inc, 2009.

- [10] 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.
- [11] 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>
- [12] 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.
- [13] “What is wordnet?” 2010. [Online]. Available: <https://wordnet.princeton.edu/>
- [14] P. Fabian, G. Varoquaux, A. Gramfort, M. Vincent, B. Thirion, 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.
- [15] 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.
- [16] R. B. Muhammad, “Graph theory: Definitions and examples.” [Online]. Available: <http://personal.kent.edu/~rmuhamma/GraphTheory/MyGraphTheory/defEx.htm>
- [17] S. Rose, D. Engel, N. Cramer, and W. Cowley, “Automatic keyword extraction,” *Text Mining: Applications and Theory*, pp. 1—277, 2010.
- [18] 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>
- [19] 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>
- [20] M. E. J. Newman, “Mathematics of networks,” *Networks*, pp. 109–167, 2010.
- [21] 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.
- [22] C. McCormick, “Word2Vec Tutorial - The Skip-Gram Model,” pp. 1–39, 2017.
- [23] Q. Liu, M. J. Kusner, and P. Blunsom, “A Survey on Contextual Embeddings,” 2020. [Online]. Available: <http://arxiv.org/abs/2003.07278>
- [24] P. Melville and V. Sindhvani, “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.sindhvani.org/recommender.p](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.sindhvani.org/recommender.p)