# Independent Project: Final Report

## Bee Species Prediction Using Feature Specifications in Text Format

Shivangi Dhiman (2018265)
Tejas Dubhir (2018110)

Advisor: Dr. Swapna Purandare

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

# Problem Statement:

To predict the species of a bee, given its description in text format by using Machine Learning, Information Retrieval and Natural Language Processing techniques.

# Background:

Bees are insects that play a significant role in the pollination of food crops and the production of honey and wax. Because of this, they are considered a crucial ecological and economic resource. There are over 20,000 known species of bees around the globe - for example, Apis mellifera (Common Honey Bee), Apis dorsata (Giant Honey bee), Bombus terrestris (Buff-tailed Bumblebee), etc. Each species has its own characteristic traits that are unique to it. One of the most common or recognizable features of bees are their striped furry bodies.

However, many species possess similar characteristics which can make the task of bee identification or bee taxonomy difficult. For example, almost all of the species of honey bees have varying dark and light striations. Bumblebees also have black and yellow stripes but are often larger than honey bees. Carpenter bees are also similar to Bumblebees but they usually have shorter hair and a black body.

Unfortunately, many species of bees are now enlisted as ‘endangered. Many scientists believe this is due to various agricultural practices and the use of insecticides. Thus, in the past few years, it has become important to correctly classify bees as it would help in their proper surveying, studying and monitoring. But, due to such a large number of known species, it is infeasible to manually identify and classify all the bees, based on their physical traits.

To overcome this challenge, we have created an automated bee identification system. We have employed Information Retrieval, Natural Language Processing and Machine Learning techniques to automate this process. Using a list of features as queries/inputs, we aim to predict the species of the bee.

# Literature Review:

### 1. Snake Species Identification by Using Natural Language Processing:

- Used Weka for data extraction.
  1. Data tokenization: using weka, separated words and formed vectors.
  2. Stemming: bringing words to their natural form and removing -ing, -s, -es, etc.
  3. Removing symbols, articles and punctuations.
- Feature extraction using TF_IDF:
  $Normalised\ TF\ =\ Number\ of\ terms\ that\ occurred\ in\ the\ text/Total\ number\ of\ words\ in\ the\ text$
  $IDF\ =\ log(\ Total\ number\ of\ texts/\ No.\ of\ text\ in\ which\ selected\ term\ is\ appeared)$
  $TF\ -\ IDF\ =\ Normalized\ TF * IDF$
- Then the K-NN, Naive Bayes, SVM, and Decision Tree models were trained.
  - Decision tree 71.67% .

- SVM 68.33%.
- Naïve Bayes 61.11% .
- k-NN by 55.56% .

**2. Identifying Search Keywords for Finding Relevant Social Media Posts:**

- The initial ranking, which is less accurate but very efficient, is to identify a large shortlist of likely keywords, or in other words, to remove those unlikely keywords. Re-ranking refines the ranking of the shortlisted words.
- Used the Double Ranking Approach.
    - Step 1: Providing keywords to be searched.
    - Step 2: Finding all the keywords from the set from the dataset.
    - Step 3: Using the initial ranking algo to rank the found keywords. And the unsuitable words should be removed from consideration.
    - Step 4: Using the reranking algo, rerank all the keywords in the obtained new set, from the dataset to get the more refined set of found words.
    - Step 5: from the new rank list, a subset is selected and checked if the dataset contains the keywords from the new set or not.
    - Step 6: If the results are satisfactory, end the process, otherwise repeat from step 2.

**3. Zhang H. (et. al), 2018, Identifying Offensive Posts and Targeted Offense from Twitter**

- The paper was divided into completing two subtasks:
    - Classifying/predicting the tweets if they are Offensive (OFF) or Not offensive (NOT)
    - Differentiating between Targeted Offense (TIN) and Untargeted Offence.
- Following techniques were applied:
    - Convolutional Neural network.
    - Bidirectional LSTM.
    - Bidirectional LSTM + GRU.
    - An ensemble of the above 3 architectures.
- The authors set up their own dataset by collecting the tweets through python libraries and manually finding the tone of the text body.
- For preprocessing they used NLTK and Keras. They completed preprocessing in the steps which are:
    - Tokenization.
    - Cleaning and normalization.

**4. Armouty, B. and Tedmori, S., 2019, April. Automated Keyword Extraction using Support Vector Machine from Arabic News Documents. In 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT) (pp. 342-346). IEEE.**

- Statistical feature extraction methods used:
    - Term Frequency - Inverse Document Frequency (TF-IDF)

    - First Occurrence
- Supervised learning model:
    - Support Vector Machine Classifier
- To preprocess the data, the text was split into sentences and tokenized.
- The frequency of each word was compared to that in other documents.
- The dataset was imbalanced - with a very large number of non-keywords as compared to keywords in the documents.
    - To balance the data, they made the number of 'non-keywords' closer to the number of 'keywords' using the Downsampling Method.
- The data values were also normalized between 0 and 1, to make it comparable.
- Support Vector Machines with the Radial Basis Function (RBF) kernel was used
- SVM gave a precision of 0.77, recall of 0.58 and an F1 score of 0.65

**5. C. Zhang, H. Wang, Y. Liu, D. Wu, Y. Liao, B. Wang, "Automatic Keyword Extraction from Documents Using Conditional Random Fields" in Journal of CIS 4:3(2008), pp.1169-1180, 2008.**

- The authors proposed a sequence labelling method - 'Conditional Random Fields' (CRF), to effectively extract keywords from 600 Chinese academic documents.
- The documents were divided into 10 datasets consisting of title, abstract, full-text, headings, subheading and references.
- SegTag was used to preprocess and compile the data.
- For their dataset, the most probable label sequence was determined by:

$$Y' = arg\,max\, P(X|Y)$$

Where Y' was determined using the Viterbi Algorithm.

- Because of its ability to relax the assumption of conditional independence of the observed data, CRF was also able to avoid the label-bias problem (the ability of a model to completely ignore the current observation when predicting the next label).
- Used 10-fold cross-validation method to compare CRF with other models.
- CRF (Precision: 0.66, Recall: 0.41, F1 score: 0.51) and SVM (Precision: 0.80, Recall: 0.33, F1 score: 0.46) significantly outperformed the rest.

**6. Mouratis, T. and Kotsiantis, S., 2009, November. Increasing the accuracy of discriminative of multinomial bayesian classifier in text classification. In 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology (pp. 1246-1251). IEEE.**

- The authors combined Discriminative Multinomial Bayesian Classifier with a feature selection technique that evaluates the worth of an attribute by computing the value of the chi-squared statistic with respect to the class.
- For data preprocessing, the texts were tokenized, stemmed and represented in the vector form.
- Since SVMs produce good precision but poor recall, the authors tried to incorporate a Bayesian network classifier, with Frequency Estimate (FE) as the parameter.

- Using the Chi-squared method for feature ranking, it was found that an alpha of 0.01 produced the best results.

**7. Rose, S., Engel, D., Cramer, N. and Cowley, W., 2010. Automatic keyword extraction from individual documents. Text mining: applications and theory, 1, pp.1-20.**

- The authors proposed a method called Rapid Automatic Keyword Extraction (RAKE): use a list of stopwords and phrase delimiters to detect the most relevant words or phrases in a text.
    - Step 1: the text is split into a list of words and the stopwords are removed to give a list of 'content words'.
    - Step 2: the algorithm creates a matrix of words and stores the number of times they co-occur.
    - Step 3: the words are given a score, which is the degree of a word in the matrix, the degree of the word divided by its frequency, or simply the word frequency.
- If two keywords or keyphrases appear together in the same order more than twice, a new keyphrase is created regardless of how many stopwords the keyphrase contains in the original text.
- A keyword is chosen if its score belongs to the top T scores where by default, T is one-third of the content words in the document.

**8. Mahata, D., Shah, R.R., Kuriakose, J., Zimmermann, R. and Talburt, J.R., 2018, April. Theme-weighted ranking of keywords from text documents using phrase embeddings. In 2018 IEEE conference on multimedia information processing and retrieval (MIPR) (pp. 184-189). IEEE.**

- The authors used word-embedding algorithms to extract keywords from the arxiv dataset.
- Preprocessing: stemming, tokenization, removal of stopwords.
- To train this preprocessed data, two toolkits - Word2Vec and Fasttext were used.
    - The Word2Vec algorithms include skip-gram and CBOW (continuous bag of words) models, using either hierarchical softmax or negative sampling.
        - In the CBOW model, the distributed representations of surrounding words are combined to predict the word in the middle.
        - In the Skip-gram model, the distributed representation of the input word is used to predict the context.
    - Fasttext: used for efficient learning of word representations and sentence classification.
- It was found that the models trained using Fasttext performs better than the models trained using Word2Vec on all the three tasks

**9. Slobodan Beliga, Keyword Extraction: A Review of Methods and Approaches**

Keyword extraction is a well-known problem in the fields of Text mining, information retrieval, and natural language processing. Assigning keywords to large bodies of texts saves a lot of computation which in turn leads to saving a lot of time. The following sub-topics are covered in the paper:

- Keyword extraction methods

    - Simple Statistics Approaches: These don't require a training dataset, but just some statistical techniques are used for such approaches namely n-gram statistics, word frequency, TFI-DF, word co-occurrences, PAT Tree, etc. These do not give preferable results in cases where the keywords' frequency is relatively lower.
    - Linguistics Approaches: Lexical, syntactic, semantic and discourse analysis can be considered as a part of such methods.
    - Machine Learning Approaches : Mostly supervised techniques are preferred. The dataset needs to be manually annotated for training and thus makes the tasks difficult for the researchers.
- A new graph-based method called Selectivity-Based Keyword Extraction together with experimental results on Croatian News articles
- Supervised learning: The task is to classify whether the given text contains the keywords or not. It is a binary classification problem.
  The following are the observations from various papers:
    1. Noun phrases can be used instead of frequency and n-grams, and then POS tags can be added to them.
    2. Statistical associations between keyphrases and enhancing the coherence of the extracted keywords can be done with the Naive Bayes Algorithm
- Vector Space Model: It is the method where the sentences are broken into vectors and stored as word frequency. It has several disadvantages such as the meaning of the sentence gets lost in the process, the words are independent of each other which means that "not", "bad" and "not bad" have entirely different meanings, if there are sentences that have different meanings but have the same words, they would be considered the same.
- Thus, instead of VSM, graph-based models are preferred. In this article, the study is done on the levels namely semantic and pragmatic, syntax, morphology, phonetics, and phonology.

### 10. Automatic Keyword Extraction for Text Summarization: A Survey

- Text Extraction:
    - Simple Statistical Approach:
        - Training set is not required, extraction is based on statistics derived from the frequency and the position of a word and a list of keywords is generated.
        - Techniques Used:
            - Term Frequency (TF) (HP Luhn, 1957)
            - Term Frequency - Inverse Document Frequency (TF-IDF) (J. Ramos, et al., 2003)
            - Word occurrences (Y. Matsuo, et al., 2004)
            - PAT-tree, n-gram statistical data (J. D. Cohen, et al., 1995)
        - After extraction, Aprioiri techniques, such as finding support and confidence were then used to infer the strength of the keywords.
    - Linguistics Approach:
        - Linguistic features such as lexical analysis (R. Barzilay, et al., 1999) (applied using electronic dictionary, tree tagger, WordNet, n-grams, POS pattern),

syntactic analysis (A. Hulth, 2003) (noun phrase, chunks), discourse analysis (G. Salton, et al., 1997) etc., were incorporated.

- Machine Learning Approach:
  - Training data is required.
  - Commonly used training models were
    - Hidden Markov models (J. M. Conroy, et al., 2001),
    - Support Vector Machine (SVM) (K. Zhang, et al., 2006),
    - Naive Bayes (E. Frank, et al., 1999),
    - Bagging (A. Hulth, 2003).
    - Keyword Extraction Algorithm or KEA (I. H. Witten, et al., 1999) is one of the most prevalent algorithms where the article is converted into a graph and each word is treated as a node and whenever two words appear in the same sentence, the nodes are connected with an edge for each time they appear together. Then the number of edges connecting the vertices are converted into scores and are clustered accordingly. The cluster heads are treated as keywords.
- Hybrid Approach: These approaches combine the Machine Learning and Statistical approaches to find the best features (J. K. Humphreys).

- Text Summarization
  - Statistical Based Approach
    - Similar to the text extraction method, statistical-based approaches make use of term frequency (TF), IDF, the position of the word, etc. (J. R. Thomas, et al., 2016; M.Chandra, et al., 2011, M. R. Murthy, et al., 2011)
  - Machine Learning-Based Approach
    - Labelled data is used for training.
    - Hidden Markov models (J. R. Thomas, et al., 2016) and SVM classifiers.
    - Naive Bayes (J. D. M. Rennie, et al., 2003)
    - Hidden Markov Models (L. Rabiner, et al., 2003)
    - Maximum Entropy, Neural Network (T. Jo, et al., 2010)
    - Support Vector Machine (L. N. Minh, et al., 2005)

Results obtained from the papers reviewed:

| **Paper** | **Method** | **Score (%)** | **Precision** | **Recall** | **F1 Score** |
|---|---|---|---|---|---|
| Rusli et.al, 2019 | Decision Trees | **71.67** | | | |
| Rusli et.al, 2019 | SVM | 68.33 | | | |
| Rusli et.al, 2019 | Naive Bayes | 61.11 | | | |
| Rusli et.al, 2019 | K-NN | 55.56 | | | |
| Wang et.al, 2016 | Double Ranking Algo | **81.3** | | | |

| | | | | | |
|---|---|---|---|---|---|
| Armouty et.al, 2019 | SVM | - | 0.77 | **0.58** | **0.64** |
| Armouty et.al, 2019 | Naive Bayes | - | **0.8** | 0.43 | 0.56 |
| Armouty et.al, 2019 | Random Forest | - | 0.71 | 0.57 | 0.63 |
| Zhang et.al, 2008 | Conditional Random Fields | - | 0.6637 | 0.4196 | **0.5125** |
| Zhang et.al, 2008 | Logit | - | 0.3248 | **0.5388** | 0.4067 |
| Zhang et.al, 2008 | SVM | - | **0.8017** | 0.3327 | 0.4653 |
| Mouratis et.al, 2009 | Multinomial Naive Bayes | **89.03** (Average accuracy) | - | - | - |
| Mouratis et.al, 2009 | Sequential Minimal Optimization | 81.96 (Average accuracy) | - | - | - |

| **Paper** | **Method** | **F1 micro** | **Accuracy** |
|---|---|---|---|
| Zhang H. et.al, 2019 | Convolutional Neural Network (on training data) | 0.8020 | 0.8387 |
| Zhang H. et.al, 2019 | Bidirectional LSTM with Attention (on training data) | 0.7851 | 0.8246 |
| Zhang H. et.al, 2019 | Bidirectional LSTM + Bidirectional GRU (on training data) | 0.7893 | 0.8301 |
| Zhang H. et.al, 2019 | MIDAS Submission 1 on test data (CNN) | 0.7964 | 0.8395 |
| Zhang H. et.al, 2019 | MIDAS Submission 2 on test data (Ensemble of CNN, BLSTM with Attention, BLSTM + BGRU) | 0.8066 | 0.8407 |

| Paper | Method | Accuracy |
|---|---|---|
| Mahata V. et.al, 2018 | Word2Vec Skipgram | 76.58% |
| Mahata V. et.al, 2018 | Word2Vec CBOW | 69.11% |
| Mahata V. et.al, 2018 | Fasttext Skipgram | 96.27% |
| Mahata V. et.al, 2018 | Fasttext CBOW | 93.92% |

Some of the first bees to emerge in spring, members of the genus Andrena vary greatly in size and appearance. Females have patches of velvety hairs between the eyes.

# Work done, Approaches:

## Data Collection:

One major issue that we faced was the limited availability of a dataset that consisted of the physical traits of different species of bees. Thus, we first scraped and collected data from several websites, research papers, field guides, books and articles and tried to separate the species description from the titles and created a dataset.

We also implemented Image to Text conversion methods using python's cv2 library (pytesseract) to extract text from older editions of books, for which a text format was not available. One book that we used majorly for the extraction of data was 'The Bees of the World' (Second Edition) by Charles D. Michener. We first converted the pages of the PDF to image and then extracted the text.

After compiling all the resources the format of the dataset was decided - the title i.e. the name of the species followed by the description of the species containing the physical characteristics of the bees which would then be compared with the input query text. This step took most of the time as the number of bees species available is around 20,000 and manually collecting such a large amount of data was a tedious event.

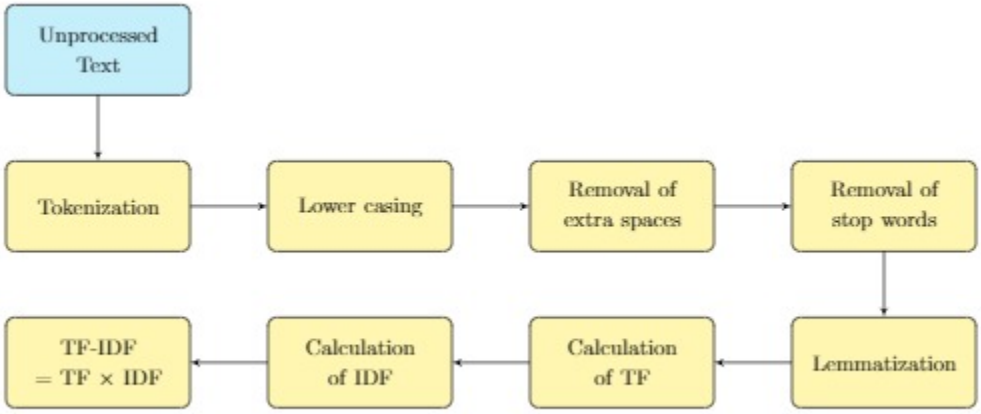
We eventually prepared a sample dataset containing 102 species of bees and their detailed description in order to create the algorithm and validate it using those samples.
A small component of the smaller dataset:

| | |
|---|---|
| Mining Bees (Andrena) | Some of the first bees to emerge in spring, members of the genus Andrena vary greatly in size and appearance. Females can be recognized by patches of velvety hairs between the eyes. Mining bees carry pollen on their hind legs and on hairs between the abdomen and the thorax. Preferred Crops: Apple, cherry, peach, and pear. Nesting Behavior: Solitary. Andrena nest in small tunnels in the ground. |
| Mason Bees (Osmia) | Like leafcutter bees, Osmia have large jaws and big heads. They range in color from metallic blue to green, occasionally black. Their abdomens often have a rounded appearance. Osmia are called mason bees because they use mud to make their nest cells. Several species are managed for agricultural production. Mason bees carry pollen on specialized hairs on the abdomen. |
| Small Carpenter Bees (Ceratina) | Certina are small mostly hairless bees that vary in color from dark metallic blue to green. They emerge in the spring and stay active until fall. Small carpenter bees have rudimentary pollen-carrying hairs. They may transport pollen by swallowing it and regurgitating it back at the nest. This behavior has been observed in primitive bees. Preferred Crops: Apple, cane berries, cherry, pear, and strawberry. |
| Long-Horn Bees (Eucerini) | They are solitary bees with about 500 species in 32 genera in the tribe Eucerini. Long-horn bees have hairy bodies and legs with black and tan markings. One common distinguishable feature is their long antennae. Long-horn bees are commonly found feeding on pollen on sunflowers. These bees don't produce honey and live a solitary existence where they nest in small tunnels. Long-horn bees generally have pale bands on black fuzzy bodies and two long antennae. Their six legs are hairy and a dark tan color. |

## Dataset Preprocessing:

The obtained text descriptions of the species were then preprocessed using the following steps:

1. **Tokenization**: the string containing the text was broken down into smaller strings by the space character (' '). Then the obtained description was stored in an array.
2. **Lower casing:** the words in the array were then converted to the lowercase in order to maintain uniformity among the words which were the same but differed in their capitalisation. For instance, "NorthWestern", "Northwestern", and "northwestern".
3. **Remove extra spaces:** the extra space between words needed to be removed as it could have led to confusion in subsequent processes (ex: whether the white space character is a part of the word or not). We felt that this could also later help us in bi-gram indexing, which is a future aspect of this project.
4. **Remove stop-words:** irrelevant words which do not add to the meaning of the sentence (such as: the, a, then, etc.) had to be removed so that only the meaningful keywords remain in the dataset.
5. **Lemmatization:** the remaining words were then converted to their root form. This means that words like 'pollinated', and 'pollinating' were represented by one word - 'pollinate' - and hence were considered the same.




Flow chart for preprocessing and score calculation.

## Ranks Prediction:

In order to create a predicted rank list, we applied three ranking algorithms to the list of queries and merged them all into one, to finally return the top 15 most likely species. The three rank lists are individually made by raw TF-IDF extraction, binary TF-IDF extraction and log-normalized TF-IDF extraction.

Raw TF-IDF

Term Frequency - Inverse Document Frequency (TF-IDF) is a numerical statistic that is used to represent the importance of a word in a document in a collection or corpus.

Once the preprocessed data is obtained, the next step is to calculate the term frequency for each of the terms in each of the sample classes. The measure of the originality of a word can be compared by comparing the number of times a word appears in a text with the number of samples the word appeared in. It shows the importance of the word in the document as well.

$$TF = Number\ of\ times\ the\ term\ occurred\ in\ the\ text / Total\ number\ of\ words\ in\ the\ text$$

On the other hand, iDF i.e. the inverse document frequency describes the amount of information a word is providing across the sample documents.

$$IDF = log(Total\ number\ of\ texts\ /\ Number\ of\ texts\ in\ which\ the\ selected\ term\ appeared)$$

Therefore,
$$TF - IDF = TF * IDF$$

Now, the TF-IDF score of a word shows how significant it is to the text and here, it will tell us how relevant the word is in describing the physical characteristics of the specified bee species.

Binary TF-IDF

After the vanilla TF-IDF calculation, in order to reduce the spontaneity in the algorithm, we calculate the rank list through Binary TF-IDF extraction.

In this method, the TF calculation is different from the raw TF-IDF calculation. If the text contains the word, the binary TF is 1, otherwise, the binary TF is 0. Thus, no matter how many times the word occurs in a document, its binary TF will always be 1.

$$Binary\ TF\ =\ max(1,\ Number\ of\ times\ the\ term\ occurred\ in\ the\ text)$$

The IDF is calculated in the same way as above, and thus, the final scores are calculated by the formula:
$$Binary\ TF\ -\ IDF\ =\ Binary\ TF\ *\ IDF$$

Log Normalised TF-IDF:

The log normalized TF is calculated by the following formula:
$$Log\ Normalised\ TF\ =\ log(1\ +\ Number\ of\ times\ the\ term\ occurred\ in\ the\ text)$$

Here 1 is added to normalise the term and prevent the value from reaching large extremes. The final score is calculated by multiplying the log normalised TF to IDF.

$$Log\ normalized\ TF\ -\ IDF\ =\ Log\ Normalized\ TF\ \ *\ IDF.$$

---

Thus, the final rank list is prepared by voting the candidates of all the above-obtained rank lists and merging them.

## Machine Learning:

We trained 6 models which had the species of the bees in the form of one hot encoded classes as the ‘y’, and the vectorised description of the species as ‘x’. We then train

1. Naive Bayes
2. Support vector machine with linear kernel and degree 3.
3. Random forest
4. K-nearest neighbors
5. logistic regression
6. neural network model with random state = 1 and maximum iterations = 500,
   with the above mentioned dataset. We had a testing set containing 9 samples which were species from the dataset, and the query text was a sequence of random words along with some words from the description of the species in a jumbled fashion.

### Graphical User Interface

To display the working of our model, we created a Graphical User Interface (GUI) using Python's tkinter module. It consists of a textbox where the user can enter the list of features/queries of the bee they want to identify the species of. On clicking on the 'Get Output' button, a list of 15 most likely species would be displayed.

## Results:

The queries entered for ranking were selected from the sample dataset itself and the terms in the description were randomly selected and then shuffled in order to introduce some randomness while retrieval. Four such queries were created and all the 3 ranking algorithms were run for each of them. All of the preprocessing steps are applied to the text of each of the queries. The output of one of the queries is shown below. The three algorithms returned the following results:

Log-normalisation algo: [
'Family: OXAEEOAE',
'Genus Andrena Fabricius:',
'Velvet ants (Mutillidae)',
'Genus: Agapostemon',
'Subgenus: Seladonia',
'Genus: Osmia',
'Genus: Halictus Sweat Bees',
'Genus: Lasioglossum',
'Subfamily: Dasypodinae ',
'Genus: Dialictus ',
'Genus: Melissodes',
'Subfamily: Andreninae',
'Andrenidae ',
'Genus: Ceratina',
'**Genus: Andrena**']

Boolean TF-IDF algo:
['Genus: Melissodes',
'Subfamily: Hylaeinae',
'SYMPHYTA (Sawflies and wood-wasps)',
'Dufoureinae',
'Genus: Agapostemon',
'Social bumblebees and Cuckoo bumblebees',
'Gall wasps (Cynipidae)',
'Genus: Lasioglossum',
'Family: OXAEEOAE',
'Velvet ants (Mutillidae)',
'Genus: Ceratina',
'Subfamily: Andreninae',

'Genus: Dialictus ',
'Subfamily: Dasypodinae ',
'**Genus: Andrena**']

Raw TF-IDF algo:

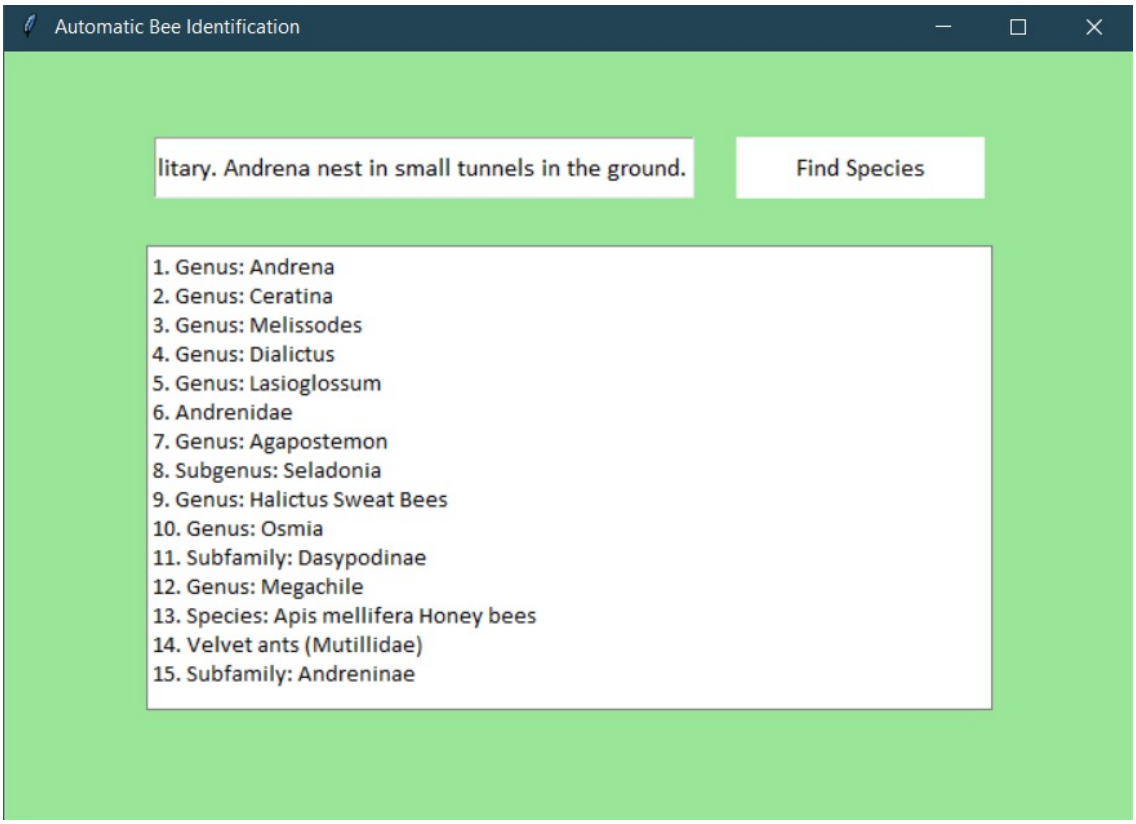
['Subfamily: Andreninae',
'Velvet ants (Mutillidae)',
'Species: Apis mellifera Honey bees',
'Genus: Megachile',
'Subfamily: Dasypodinae ',
'Genus: Osmia',
'Genus: Halictus Sweat Bees',
'Subgenus: Seladonia',
'Genus: Agapostemon',
'Andrenidae ',
'Genus: Lasioglossum',
'Genus: Dialictus ',
'Genus: Melissodes',
'Genus: Ceratina',
'**Genus: Andrena**' ]

Here, the most probable species is written in BOLD, and as we can see, all the rank lists have the same number one contender. This means that the final rank list will have this as the number one too. The rest of the ranks are decided by merging the three lists and weighting the elements on the basis of their comparative ranks. Therefore the final ranking comes out to be :

Gall wasps (Cynipidae)
Family: OXAEEOAE
Genus: Osmia
Subgenus: Seladonia
Velvet ants (Mutillidae)
Genus: Halictus Sweat Bees
Genus: Agapostemon
Andrenidae
Subfamily: Andreninae
Genus: Melissodes
Genus: Lasioglossum
Subfamily: Dasypodinae
Genus: Dialictus
Genus: Ceratina
**Genus: Andrena**

It is now confirmed that the number one element on the list is the same in the final list as in the intermediate lists.
We checked this for all the four sample queries created by randomly selecting words from the description of 4 randomly selected species and then shuffling their text content. For each of the following the most probable species was predicted correctly.

When this is combined with the GUI, we get such an output:




The accuracies obtained by the machine learning models were :

1. Naive Bayes Accuracy Score -> 77.77777777777779
2. SVM Accuracy Score -> 77.77777777777779
3. Random Forest Accuracy Score -> 33.33333333333333
4. KNN Accuracy Score -> 11.11111111111111
5. Logistic Regression Accuracy Score -> 88.88888888888889
6. Neural Network Accuracy Score -> 77.77777777777779

# Future Work

- Bi-grams/multi-grams indexing for ranked retrieval.
- The bigger dataset used for training, the more accurate results would be returned.
- Use of the vector space model.
- Application of machine learning techniques such as SVM, naive bayes and random forest.

- Using deep learning models for better predictions like LSTM and RNNs.
- Use Computer Vision technique to identify the species of bees from an image.

[1] B. Armouty and S. Tedmori. 'Automated keyword extraction using support vector machine from Arabic news documents'. In: 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT). IEEE. 2019, pp. 342–346.
**[2] C. Bingham. 'The Fauna of British India, Including Ceylon and Burma. Hymenoptera-Vol.1. Wasps and Bees.' In: (1897).**
**[3] C. D. Michener. The bees of the world. Vol. 1. JHU press, 2000.**
[4] J. Ramos et al. 'Using tf-idf to determine word relevance in document queries'. In: Proceedings of the first instructional conference on machine learning. Vol. 242. 1. Citeseer. 2003, pp. 29–48.
[5] About Bees, https://idtools.org/id/bees/exotic/bees classification.php.
[6] Alper. NLP: Classification and Recommendation Project. July 2020, towardsdatascience.com/nlp-classification-recommendation-project-cae5623ccaae.
[7] S. Bob. HYMENOPTERA Bees, Wasps, Sawflies &; Ants. May 2014, bobs-bugs.info/2014/01/02/hymenoptera-bees-wasps-ants-etc.
[8] S. Kincaid. Common Bee Pollinators of Oregon Crops. 2017, oregon.gov
[9] Palaearctic Osmiine Bees, blogs.ethz.ch/osmiini/phylogeny-and-classification.
[10] The Most Beneficial Types of Bees (With Identification Guide and Pictures). Apr. 2021, leafyplace.com/types-of-bees.
[11] Wood Wasps and Sawflies, amentsoc.org/insects/fact- files/orders/hymenoptera-symphyta.html.

# References

- A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in: Proceedings of the 2003 conference on Empirical methods in natural language processing, ACL, 2003, pp. 216-223.

- E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, C. G. Nevill-Manning, "Domain-specific key-phrase extraction," In 16th International Joint Conference on Artificial Intelligence, vol. 2, 1999, pp. 668-673.

- F. Schilder, R. Kondadadi, "Fastsum: fast and accurate query-based multi-document summarization," in: Proceedings of the 46th annual meeting of the association for computational linguistics on human language technologies, ACL, 2008, pp. 205-208.

- G. Salton, A. Singhal, M. Mitra, C. Buckley, "Automatic text structuring and summarization," Information Processing & Management, vol. 33 (2), 1997, pp. 193-207.

- H. P. Luhn, “A statistical approach to mechanized encoding and searching of literary information,” IBM Journal of research and development, vol. 1 (4), 1957, pp. 309-317.

- I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, C. G. Nevill-Manning, “Kea: Practical automatic key-phrase extraction,” in: Proceedings of the fourth ACM conference on Digital libraries, ACM, 1999, pp. 254-255.

- J. D. M. Rennie, L. Shih, J. Teevan, D. R. Karger, “Tackling the poor assumptions of nave Bayes classifiers,” In Proceedings of International Conference on Machine Learning, 2003, Pp. 616–623

- J. D. Cohen, et al., “Highlights: Language- and domain-independent automatic indexing terms for abstracting,” JASIS, vol. 46 (3), 1995, pp.162-174.

- J. K. Humphreys, “Phraserate: An html key-phrase extractor,” Dept. of Computer Science, University of California, Riverside, California, USA, Tech. Rep.)

- J. M. Conroy, D. P. O'leary, “Text summarization via hidden Markov models,” in: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2001, pp. 406-407.

- J. Ramos, “Using tf-idf to determine word relevance in document queries,” in: Proceedings of the first instructional conference on machine learning, 2003, pp. 1-4.

- J. R. Thomas, S. K. Bharti, K. S. Babu, “Automatic keyword extraction for text summarization in e-newspapers,” in: Proceedings of the International Conference on Informatics and Analytics, ACM, 2016, pp.86-93.

- K. Zhang, H. Xu, J. Tang, J. Li, “Keyword extraction using support vector machine,” in: Advances in Web-Age Information Management, Springier, 2006, pp. 85-96.

- L. N. Minh, A. Shimazu, H. P. Xuan, B. H. Tu, S. Horiguchi, “Sentence extraction with support vector machine ensemble,” In Proceedings of the First World Congress of the International Federation for Systems Research, 2005, pp. 14-17.

- L. Rabiner, B. Juang, “An introduction to hidden Markov model,” Acoustics Speech and Signal Processing Magazine, vol. 3(1), 2003, pp.4–16.

- L. Shi, F. Wei, S. Liu, L. Tan, X. Lian, M. X. Zhou, “Understanding text corpora with multiple facets, in: Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on, IEEE, 2010, pp. 99-106.

- M. Chandra, V. Gupta, and S. Paul, “A statistical approach for automatic text summarization by extraction,” In Proceeding of the International Conference on Communication Systems and Network Technologies, IEEE, 2011, pp. 268–271.

- M. R. Murthy, J. V. R. Reddy, P. P. Reddy, S. C. Satapathy, "Statistical approach based keyword extraction aid dimensionality reduction," In Proceedings of the International Conference on Information Systems Design and Intelligent Applications. Springer, 2011.

- R. Barzilay, M. Elhadad, "Using lexical chains for text summarization," Advances in automatic text summarization, 1999, pp. 111-121.

- T. Jo, "Ntc (neural network categorizer) neural network for text categorization," International Journal of Information Studies, vol. 2(2), 2010.

- Y. Matsuo, M. Ishizuka, "Keyword extraction from a single document using word co-occurrence statistical information," International Journal on Artificial Intelligence Tools, vol. 13 (01), 2004, pp. 157-169.

- Z. L. Min, Y. K. Chew, L. Tan, "Exploiting category-specific information for multi-document summarization," in Proceedings of COLING, ACL, 2012, pp. 2093–2108.

- https://towardsdatascience.com/natural-language-processing-document-search-using-spacy-and-python-820acdf604af

- https://medium.com/analytics-vidhya/automated-keyword-extraction-from-articles-using-nlp-bfd864f41b34

- https://towardsdatascience.com/nlp-classification-recommendation-project-cae5623ccaae

- https://monkeylearn.com/keyword-extraction/

- https://medium.com/@bedigunjit/simple-guide-to-text-classification-nlp-using-svm-and-naive-bayes-with-python-421db3a72d34

- http://www.bobs-bugs.info/2014/01/02/hymenoptera-bees-wasps-ants-etc/
- https://www.oregon.gov/ODA/shared/Documents/Publications/IPPM/ODABeeGuide.pdf
- https://ir.library.oregonstate.edu/downloads/m613n331f
- https://www.jstor.org/stable/25084960?seq=1
- https://www.amentsoc.org/insects/fact-files/orders/hymenoptera-symphyta.html
- https://jhr.pensoft.net/article/27704/
- https://riversedgewest.org/sites/default/files/resource-center-documents/Carril%20et%20al%202018%20Bees%20GSENM%20%281%29.pdf
- https://www.biotaxa.org/Zootaxa/article/view/zootaxa.3765.2.4
- https://blogs.ethz.ch/osmiini/phylogeny-and-classification/

- https://leafyplace.com/types-of-bees/
- https://idtools.org/id/bees/exotic/bees_classification.php