Categorizing Ingredients Using Machine Learning and Natural Language Processing Methods

CS-E4870 - Research Project in Machine Learning and Data Science
Kristel Ariko

I. INTRODUCTION

The past 10 years I have had a discount card for S Group companies and during the recent years I have concentrated my purchases to the companies which belong to the S Group to gain as much bonus points as possible. The bonus points and transactions of the purchases within last 12 months can be seen in the S Groups website [1].

The intention of this work is to collect distinct ingredients from the recipes and cluster similar ones together. This information could be used in a food recommendation program which is based on the similar ingredients. For example, if the user of the program fills in its daily needs for nutrients and pre-picks some ingredients, then the program could suggest the rest of the ingredients to fulfill the daily needs. Moreover, the program could also suggest which ingredients are possible to replace with others by keeping the nutrient values same.

The clustering of the ingredients is performed using noisy text (name) of the ingredients and additional information mentioned already above, nutrients. The methods that are used come from the field of Machine Learning and Natural Language Processing. Some of the methods are trained on labeled data that is later used as a comparison for the other methods trained on non-labeled data. In the final conclusion, this work assumes that the known labels of the ingredients are difficult to collect and therefore it selects the method that is best for predicting labels of the ingredients based on the name of the ingredient and its nutrients. This approach keeps additional data gathering simple and does not require labeled data.

The works starts with introducing the dataset and describing Machine Learning and Natural Language Processing methods. Then goes to details in the training and evaluating. Finally, the results are presented and conclusions are drawn.

The source code of this work can be found in https://github.com/letsirk/eiv-research-project.

II. DATASET

The dataset includes about 500 distinct ingredients and is fetched from S Group's website [1] by polling its API interface. The name of the ingredient consists of 1-5 Finnish words in any order and it does not follow any common rule. Furthermore, the dataset is extended by crawling external information, such as nutrients and hierarchical grouping, from S Group's online store [2].

During the crawling the values of the nutrients are converted to the same format (e.g. gram) and ingredients (e.g. toilet paper) which do not have nutrient information, are excluded. Thus, the dataset is reduced to 363 ingredients including following nutrients: Fat (Rasvaa), Carbohydrate (Hiilihydraattia), Protein (Proteiinia), Dietary Fiber (Ravintokuitua), Energy (Energiaa), Salt (Suolaa) and Sugar (Sokeria).

The hierarchical grouping consists of 4 groups whereas the first group is a general group including all similar ingredients (e.q. tomatoes and cucumbers) and the last group is a more detailed group including only exactly same ingredients (e.g. tomatoes). The intention of this work is to cluster similar ingredients and therefore only the first group is taken into account. Thus, the final dataset involves 9 columns and 363 rows (Table II). The context of the first group is described in Table I.

TABLE I: The labels of the dataset.

Id	Name
351	Maito
994	Juomat
417	Juustot
497	Valmisruoka
100	Hedelmät ja vihannekset
701	Leipä
80	Liha
1337	Lemmikit
1144	Makeiset, jäätelöt ja naposteltavat
895	Kuivatuotteet
807	Leivonta ja maustaminen
1784	Munat
1075	Pakasteet
476	Rasvat ja öljyt
262	Kala
12129	Terveystuotteet

From now on, when this work mentions words *labels* and *features*, they are referring to the first group of the hierarchical grouping (Group 1 in Table II) and to nutrients (Rasvaa, Hiilihydraattia, Proteiinia, Ravintokuitua, Energiaa, Suolaa and Sokeria in Table II).

III. MACHINE LEARNING

Machine Learning (ML) is a data analysis method raised from Artificial Intelligence. The method involves learning from data, recognizing patterns and making decisions with minimal human intervention. Thus, the data is fed into a ML algorithm which builds the logic based on the given data, without being programmed to perform specific logic. The goal is to figure out the structure of the data. [3]

There are four types of ML methods: (1) supervised learning, (2) unsupervised learning, (3) semisupervised learning

TABLE II: Example of the dataset including ingredient's name, label and features.

Name	Group 1	Rasvaa (Fat)	Hiilihydraattia (Carbohydrate)	Proteiinia (Protein)	Ravintokuitua (Dietary Fiber)	Energiaa (Energy)	Suolaa (Salt)	Sokeria (Sugar)
maustettu rahka 200g irish cream	Maito:351	7.0	16.0	7.2	0.0	652.0	0.08	0.0
lempi 200g 28% lton crème fraiche	Maito:35	28.0	2.5	2.3	0.0	1132.0	0.10	0.0
valio hedtarha luomumustmehu 1l	Juomat:994	0.0	9.2	0.0	0.0	158.0	0.00	0.0
maustettu rahka 200g appels lton	Maito:351	7.0	16.0	7.2	0.0	650.0	0.10	0.0
alpro 400g mango pbay soijavalmiste	Maito:351	2.0	5.8	3.7	1.0	255.0	0.22	0.0

and (4) reinforcement learning. In the supervised learning both features and corresponding labels are known and the algorithm is trained using them. During the training the algorithm compares its outputs to correct outputs and modifies the model according to the found error. [3]

Unlike the supervised learning, the unsupervised learning does not know the corresponding labels. Therefore, the algorithm takes features, explores them and tries to find out a structure within them. The semisupervised learning is a mixture of the supervised and unsupervised learning. The algorithm trains on both labeled and unlabeled data. In the reinforcement learning, the algorithm discovers through trial and error which actions yield the greatest rewards. The goal is to learn the best policy to be followed such that reward of taking an action is maximized. [3]

This work utilizes decision tree and random forest algorithms as supervised learning methods and K-means++ clustering as the unsupervised learning methods. All of these algorithms are described in section III-A. Then continues to overview general machine learning concepts: distance metrics (III-B) and feature selection (III-C). As well, how the models are trained (III-D) and evaluated (III-E).

A. Models

This section introduces three ML models: 1) Decision Tree, 2) Random Forest and 3) K-means++. The first two models are supervised models which require both features and labels of the dataset (II) for training. The last model is unsupervised and needs only features for training.

1) Decision Tree: is a powerful ML model where the knowledge learned in training is converted into hierarchical structure (=tree). The creation of the model starts with building a tree that uses hierarchical decision boundaries based on the features of the data. This process is also called induction. Like in any other ML task, the goal is to predict the value of the target variable (label) based on the input variables (features). After induction, a process called pruning takes control and removes the unnecessary structure (less important branches) from a tree. This procedure simplifies the structure of the tree, reduces overfitting and increases the performance. [4]

There are two types of decision trees: classification tree and regression tree. The former classifies features to discrete set of values (pre-defined labels) and the latter predicts continuous values, usually real numbers. [5]

This work uses classification based decision tree model from Scikit Learn library. The model is trained (section III-D) on the features and labels described in section II and predicted labels are computed for evaluation (III-E).

2) Random Forest: is even more powerful ML model than decision tree. The model consists of multiple decision trees which are merged together to get better accuracy in predictions. Unlike decision trees where the most important feature is searched and set of rules are constructed, this model relies on randomness. The decision trees of the random forest are built on the top of the best feature among a random subset of features. Once the trees are built, the results are averaged. Because of the randomness and enough trees in the forest, the results have more diversity yielding to better model and preventing overfitting. [6]

This work uses classification based random forest model from Scikit Learn library. The model is trained (section III-D) on the features and labels described in section II and predicted labels are computed for evaluation (III-E).

3) K-means++: is an extension of k-means. Both of these methods aim to group points which are similar to each other in the same group while dissimilar points are assigned to a different group. In other words, these are algorithms to cluster data points. The difference of the algorithms appears in the initialization of the cluster center points. With k-means all initial center points are chosen uniformly at random, which can lead to a non-optimal solution. To prevent this problem k-means can be run several times where the best solution is chosen from those different runs. In k-means++ the first center point is chosen randomly and the next ones with a probability proportional to the distance, denoted $D(v)^2$, between a point v and the closest center point chosen so far. [7]

In a particular problem, given integer k, choose k clusters and their center points which minimizes intra-cluster distances [7]

$$\sum_{j=1}^{k} \sum_{v \in C_j} ||v - c_j||_2^2 \tag{1}$$

where C_j is the set of points chosen for each cluster $j \in \{1, \cdots, k\}$, c_j is the center point of the cluster C_j and $||v - c_j||_2^2$ is the Euclidean distance (III-B) between two points.

Then, the clusters C_j and their center points c_j can be computed using the following algorithm [8]:

1) Choose center point c_1 uniformly at random from set of points V
2) Choose center points c_i ($i \in \{2, \cdots, k\}$) from V with probability $\frac{D(v)^2}{\sum_{v \in V} D(v)^2}$ where $D(v)$ is the shortest

distance from v to its closest center

3) For each cluster center c_i ($i \in \{1, \cdots, k\}$), update the center of mass of all points in $C_i : c_i = \frac{1}{|C_i|} \sum_{v \in C_i} v$
4) Repeat steps 3 and 4 until convergence

K-means++ is used to cluster the feature vector of the ingredients (features are described in section II) to k clusters where k is the total amount of the labels in Table I. The trained clusters are converted to labels and evaluated in section III-E.

B. Distance metrics

The distance metric is a mathematical formula which is used to recognize similarities (or dis-similarities) between contents. There are several distance functions and most known ones are: Minkowski, Cosine and Mahalanobis distance. Given following equation of the general distance metrics, Minkowski distance

$$\Big(\sum_{i=1}^{n} |x_i - y_i|^p\Big)^{1/p} \begin{cases} p = 1, \text{ Manhattan distance} \\ p = 2, \text{ Euclidean distance} \\ p = \infty, \text{ Chebychev distance} \end{cases} \quad (2)$$

the equation extends to Manhattan, Euclidean and Chebychev distance. The Manhattan distance is used to calculate the distance between two points in a grid like path whereas Euclidean distance is used in a plane. The last one computes the distance between two points as the sum of the absolute differences of their coordinates. [15] This work is already utilizing Euclidean distance in computing clusters using K-means++ algorithm in section III-A.3.

The Cosine distance is a measure which finds out the degree of an angle between two vectors and is formulated as follows

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\overrightarrow{||a||}\overrightarrow{||b||}} \begin{cases} \text{if } -1 \text{ then no similarity} \\ \text{if } 0 \text{ then some similarity} \\ \text{if } 1 \text{ then similar} \\ \text{else intermediate similarity} \end{cases} \quad (3)$$

The distance is especially useful when the orientation of the vectors plays a bigger role than the magnitude. [15] Thus, exploring the corpus of the ingredients and finding out which ingredients are similar and what are their labels (more details in the section III-D).

The Mahalanobis distance measures the distance between two data points in a multivariate space. This distance is used to examine the strength/similarity between two different data points and is formulated as follows

$$\sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})} \quad (4)$$

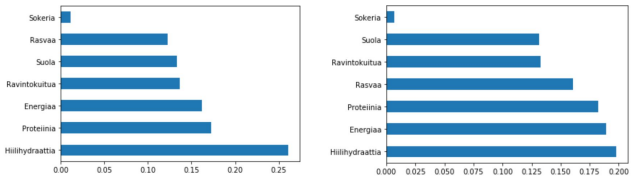
where x is the observation vector, μ is the mean of the observation vector and S is the covariance matrix. [15]

C. Feature selection

Feature selection is one of the core concepts in ML. The intention is to identify and select relevant features which promote best towards output variable (label). The performance of the model is hugely impacted by those features. The training of the model is faster and less complex if the data points have been reduced. The same reason improves the accuracy and avoids the overfitting of the model. In the overfitting the model makes decisions based on noise and those decisions increase the more redundant data points are included. [9]

There are a bunch of techniques to select features, among the others are univariate selection, feature importance and correlation matrix. In the first technique, the features are selected according to the strongest relationship with the label using statistical tests (e.g. chi-squared). The second technique gives a score value for each feature. The higher the value is the more important the feature is. The last technique explores the relationship between the other features and label. A positive value increases the value of the target feature/label and a negative value decreases it. [9]

This work utilizes the last two techniques to select relevant features in Figures 1 and 2. The importance scores in Figure 1 demonstrate that the first place is dedicated to feature Carbohydrate (Hiilihydraattia) followed by Protein (Proteiinia) and Energy (Energiaa). The correlation matrix in Figure 2 exposes that positive values are obtained by features Protein, Fat (Rasvaa) and Energy, thus, they are best describing the output label (last row named 0).



(a) Decision Tree (b) Random Forest

Fig. 1: The importance scores of the features in the dataset. The higher the score is, the more relevant the feature is.

The selected features in this work are the ones that both figures agree on, hence Protein and Energy. In addition, a third feature Carbohydrate is selected instead of Fat because it is placed as first in the Figure 1. These features are used to train (III-D) the models described in the previous section (III-A).

D. Training

Before the training, the dataset is reduced with one more ingredient whereas all the ingredients which include word 'cucumber' are removed. This information is used later to evaluate the model performance on ingredient which has not been seen in training data (III-E).

The training of the model is implemented by using k-fold cross-validation which is a method to compare and select the best performing model. In the cross-validation the training

TABLE III: The dataset used in the evaluation including ingredient's name, label and features.

Name	Group 1	Rasvaa (Fat)	Hiilihydraattia (Carbohydrate)	Proteiinia (Protein)	Ravintokuitua (Dietary Fiber)	Energiaa (Energy)	Suolaa (Salt)	Sokeria (Sugar)
tomaatti (tomato)	Hedelmät ja vihannekset:100	0.1614	3.45	0.56	1.4	95	0	0
paprika	Hedelmät ja vihannekset:100	0.21	2.5	0.94	1.9	89	0	0
kurkku (cucumber)	Hedelmät ja vihannekset:100	0.0641	1.4	0.56	0.7	46	0.01	0
retiisi (radish)	Hedelmät ja vihannekset:100	0.0902	1.95	1.44	1.6	74	0.04	0

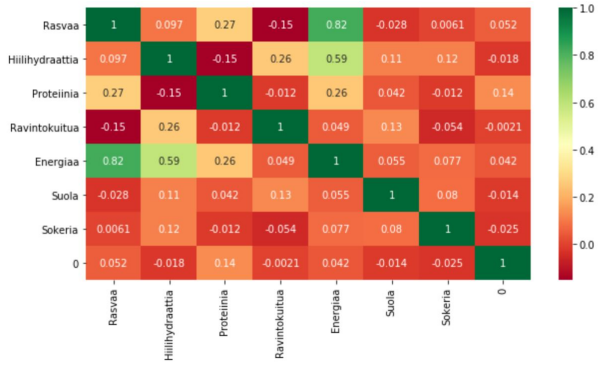


Fig. 2: The correlation of the features and label (presented with value '0') in the dataset. The higher the value is, the better it describes the corresponding feature/label.

data is partitioned into j equal parts. Then the model is trained with $j-1$ parts and validated with the part which has not been used for training. This is carried on until every part of the split has been used for validation. The intention is to validate the model's ability to predict the unseen data and to avoid problems such as overfitting and high biases. [10]

In each iteration of k-fold cross-validation, the model is trained twice: on all features and on the features which have been selected in III-C. The model which performs better in training and validation accuracy is selected. The performance is measured according to following equation

$$training_{score} * 0.8 + validation_{score} * 0.2 \qquad (5)$$

where $training_{score}$ is weighted to be more relevant than $validation_{score}$.

The above training is executed on all the models introduced in section III-A: Decision Tree, Random Forest and K-means++. Random Forest is trained using 100 trees in the forest and K-means++ predicted clusters are converted to labels before measuring the performance of the model. The transformation has following procedure: (1) take predefined labels in random order for i times, (2) at each iteration assign labels to clusters with majority elements, (3) at each iteration assign also labels that cannot be found in pre-defined labels to -1, (4) compute accuracy score and (5) choose the predictions with best accuracy score. The label order, in step (1), which gives the best accuracy score is later used in the evaluation.

E. Evaluating

The evaluation is done by several metrics: accuracy score, confusion matrix and prediction results. The first one measures the ratio of correctly predicted labels to the total labels. The higher the value is, the better the model performs. The second metric, confusion matrix, is an error matrix in a table layout. The matrix visualizes true and predicted labels and captures which labels are classified under wrong label. The last metric is used to evaluate the performance of the model on the predictions of similar (first two rows in Table III) and unseen ingredients (last two rows in Table III).

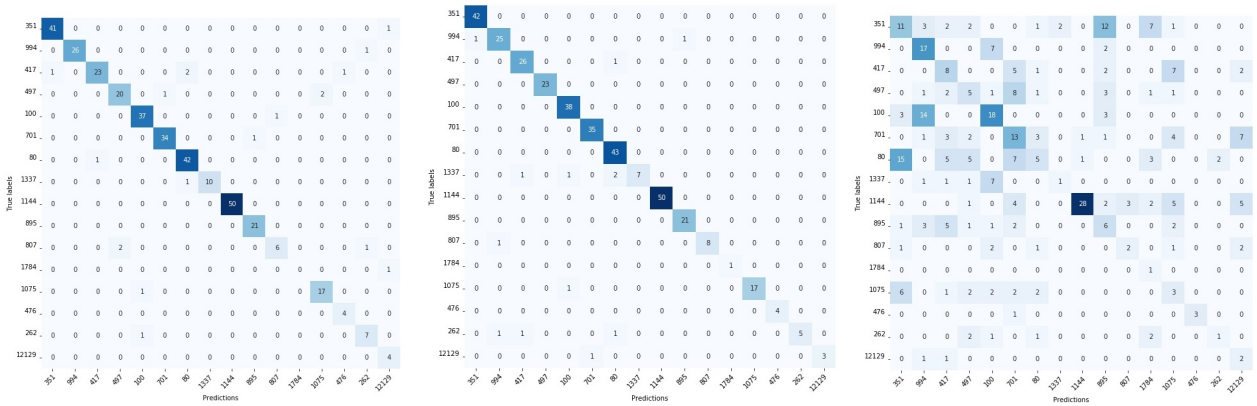
IV. NATURAL LANGUAGE PROCESSING

Natural Language Processing (NLP) is a subfield of Artificial Intelligence which aims to interact between computers and human using human's natural language. The comprehensive intention is to understand both words and the meaning of the context using the natural language in a manner that is valuable. This task is considered quite difficult, since the nature of the natural language. Especially, what rules should be processed to pass the information from natural language to computers. [11]

There are algorithms which are able to recognize and extract the rules of the natural language and convert unstructured text into a format (vectors) that is understandable for computers. Moreover, these algorithms can capture the meaning behind the sentences and thus, collect the relevant data from them. The main techniques for this task are syntactic and semantic analysis. In the syntactic analysis, grammatical rules (e.g. lemmatization, word segmentation, part-of-speech tagging, parsing, stemming) are used to a group of words and the meaning of the context is obtained. In the semantic analysis, the structure of the sentence and the meaning/interpretation of the words (e.g. named entity recognition, word sense disambiguation, natural language generation) are processed and understood. [11]

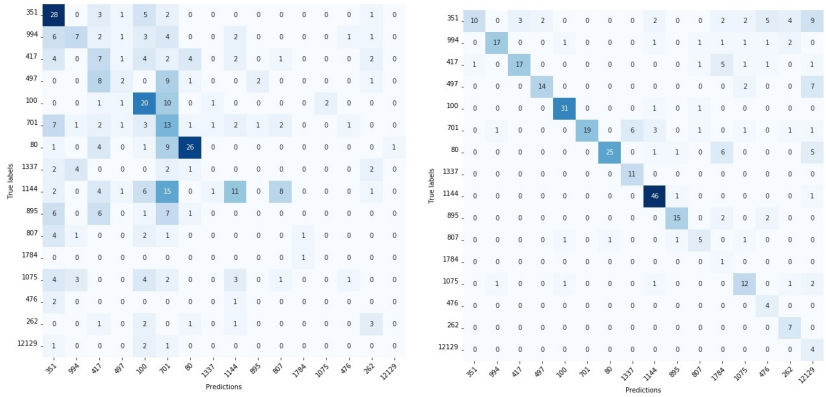
In this work a word embedding technique, called Word2Vec, is used to convert the corpus (dataset in Table II) into vectors (one for each word) that computers are capable to understand. This technique is competent to capture the context of a word in a text, semantic/syntactic similarities and association with other words. The Word2Vec model is a shallow, two-layer neural network that is trained to reconstruct the linguistic contexts of the words. Furthermore, the model can be implemented either using Skip Gram (predicting context from the words) or Common Bag of Words (CBOW, predicting words from the context). [12]

The subsections below describe the NLP models (IV-A), their training (IV-B if necessary) and evaluating (IV-C).



(a) Decision Tree (b) Random Forest (c) K-means++

Fig. 3: Confusion matrix of Machine Learning methods.



(a) Finnish Internet ParseBank (b) Gensim Word2Vec

Fig. 4: Confusion matrix of Natural Language Processing methods.

A. *Models*

This section introduces two different NLP approaches: 1) Finnish Internet ParseBank and 2) Gensim Word2Vec. The former model is a pretrained model on parsebank data and the latter model is trained using the dataset in section II. Both models use Cosine distance (III-B) to compute the similarity between two words.

1) Finnish Internet ParseBank: is a research project of Turku University which aimed to create universal web parsebanks. In other words, a web-scale corpora in many languages utilizing a cross-linguistically consistent Universal Dependencies (UD) scheme. Moreover, the intention is to have a corpora that is automatically syntactically analyzed using the method above. [14]

The outcome of the project was the first UD parsebank which included the largest Finnish Internet web corpus of over 3 billion tokens with size of 4 GB. Their intention is to extend the parsebank to support all the languages of the UD release 1.0: Czech, English, Finnish, French, German, Hungarian, Irish, Italian, Spanish, and Swedish. [14]

They provide an online demo tool[1] to query semantically similarly words. The tool uses a Word2Vec model which has been trained on the parsebank data[2]. In addition, they share a light version of a library[3] written in Python which includes the same functionalities.

This work uses the Python library to create a Word2Vec model from the parsebank data and to compute similarities between ingredients and labels for evaluation (IV-C). In other words, the model is pretrained.

2) Gensim Word2Vec: is an open source library for NLP maintained by Czech NLP researcher Radim Řehůřek and his company RaRe Technologies. The library is an efficient tool for topic modeling (discovering abstract topics from a collection of documents). Among the other, it provides functionalities for handling word embeddings: training a Word2Vec model, saving the model to file, loading it from file and making queries e.g. similarity search between words in the corpus of the model. [13]

[1]http://bionlp-www.utu.fi/wv_demo/
[2]http://dl.turkunlp.org/finnish-embeddings/
[3]https://github.com/fginter/wvlib_light

TABLE IV: Prediction results.

	True Label	Predicted Labels				
Name	Group 1	Decision Tree	Random Forest	K-means++	Finnish Internet ParseBank	Gensim Word2Vec
tomaatti (tomato)	Hedelmät ja vihannekset:100	100	100	100	100	100
paprika	Hedelmät ja vihannekset:100	100	100	100	100	100
kurkku (cucumber)	Hedelmät ja vihannekset:100	994	994	100	80	-1
retiisi (radish)	Hedelmät ja vihannekset:100	100	100	100	100	-1

This work uses Gensim library to create a Word2Vec model. Then the model is trained with the dataset in Table II (next section IV-B) and the similarities between ingredients and labels are computed for evaluation (IV-C).

B. Training

To train a NLP model, a corpus in Finnish is required. To find a big corpus in a specific language turned out to be quite difficult. Therefore, this work assembled its own corpus and combined columns *Name* and *Group 1* from Table II.

The training is done similarly as in section III-D. The only difference is that here the training is executed on NLP models introduced in section IV-A. Because one of the models is already pretrained, only Gensim Word2Vec is needed to train.

Once the model is trained (or pretrained), the ingredients are assigned to labels as following procedure: (1) split ingredient text to words (recall that ingredient may consist of 1-5 Finnish words), (2) compute Cosine similarity (III-B) between ingredient and labels in Table I and (3) select the label with highest similarity score.

C. Evaluating

The evaluation is completed exactly as in section III-E.

V. RESULTS

Once the models were trained according to III-D and IV-B, the accuracy scores of the training, validation and whole data were compared between the models described in III-A and IV-A. The leader models which learned over 90 % of the content of the dataset are random forest and decision tree (*Full* column in Table V). This result is expected since these models are both powerful algorithms in ML. As well, there is a strong belief that random forest should perform slightly better than decision tree. An interesting point of view is that the model of the decision tree achieves its best score in training on all features. The belief is that selected features (III-C) should give better results since those clearly are relevant features to describe the dataset. Although, the pruning might have removed less important features later. Because random forest takes the best feature from a subset of random features for each decision tree before merging them, it does not have such a big impact if selected features are used instead of all features.

The middle place of the leader board is reserved for Gensim Word2Vec model which covers over 60 % of the dataset. Compared to the other NLP model, Finnish Internet ParseBank which results over 30 % of the knowledge of dataset, the precision indicates that the model which is trained with domain specific data accomplishes better score.

TABLE V: Accuracy results

Model	Features	Scores		
		Training	Validation	Full
Random Forest	all	0.99	0.85	0.96
Decision Tree	all	1.00	0.75	0.95
Gensim Word2Vec	none	0.73	0.38	0.66
Finnish Internet ParseBank	none	0.30	0.42	0.33
K-means++	III-C	0.30	0.51	0.31

Moreover, if the pretrained model would be trained further with the domain specific data, the level of the accuracy would definitely increase. Recall that ingredients are assigned to labels according to the best similarity score between each word in the ingredient text and labels. If some of the words are referring to wrong labels (since human's natural language) with higher similarity score, then the ingredient is mislabeled. This could be prevented by fine tuning the ingredient text even further.

The last place is taken over by K-means++ algorithm which has also labeled over 30 % of the ingredients correctly. In order to make K-means++ clusters comparable to other methods, the clusters were converted to labels using specific procedure (more details in section III-A.3). This could explain the reason for the low accuracy score.

In addition, when digging deeper to model specific confusion matrices (Figures 3 and 4) where the true labels are predicted under wrong labels, the K-means++ and Finnish Internet ParseBank models are popping out. The former model is unable to cluster the labels which include more ingredients correctly. Thus, this model captured the knowledge of the labels on the right better than the latter model. Recall, that the first model is trained on features (=nutrients) and the second one is pretrained on the parsebank corpus. Furthermore, the result of the confusion matrix in the case of the latter model, where the predictions are missing for the labels on the right, confirms that this model could be clearly improved with additional domain specific data. Because the accuracy of these models is low, it does not make sense to compare which labels are predicted under wrong labels (most of the labels are confused with each other).

After the models were trained, predictions on similar and unseen ingredients from Table III were performed and collected to Table IV. The best performing model is the K-means++ which predicted all the labels correctly. Although, this is expected since it used the selected features to cluster the ingredients. Interestingly, the tree models, which used all the features in training, predicted cucumber to be a drink instead of the vegetable. Here the results indicate that selected features may capture the structure of some

ingredients better than all features.

The poorest model is the Gensim Word2Vec which is not able to predict ingredients that are missing on the corpus. As well as the models above, also the Finnish Internet ParseBank predicts cucumber under wrong label, meat! The explanation for this is simple; in Finnish language cucumber has another meaning: throat.

VI. CONCLUSION

In this work several models from the field of the ML (decision tree, random forest, K-means++) and NLP (Finnish Internet ParseBank, Gensim Word2Vec) were examined. The ML models were trained using either the labels and features of the ingredients or only the features of the ingredients. These models were influenced by feature selection to examine what role it plays in the evaluation. The NLP models were either not trained at all (in the case of pretrained model) or trained using the name and label of the ingredients. All the models were evaluated on the following metrics: accuracy score, confusion matrix and prediction results.

In the accuracy score the decision tree and random forest models were strong ones to predict the whole data correctly, more than 90 % of the dataset was covered. Unlike K-means++ which performed poorest, covering approx 30 % of the dataset. The NLP models landed between the best and worst models.

In the confusion matrix there were two models which did not perform well: K-means++ and Finnish Internet ParseBank. Although, K-means++ seemed to capture the structure of the dataset for the labels with small quantity much better. In here, the feature selection played a huge role. In the prediction results, the worst model so far, called K-means++, surprisingly performed best and predicted all the labels correctly. Unlike the other models, which were struggling with one or two wrong predictions.

According to the results of the predictions in Table IV and the requirements introduced in the section I, the suitable model to predict a label of the ingredient would be a combination of two models: Finnish Internet ParseBank and K-means++. Although, if the predictions rely on the meaning of the ingredient, then none of the NLP models is the best model to be used. But in the nature of simplicity requirement and human's natural language, a NLP model combined with a model that clearly notices mis-labeling based on the features and does not need additional information (such as labels) in the training, would be an interesting choice to explore further in the future works. Furthermore, K-means++ is not skimpy if some of the ingredients are missing from the training.

Other improvements that the future work could consider are:

- other alternatives to convert clusters to labels in K-means++
- training pretrained Finnish Internet Bank with additional domain specific data
- translating the ingredients to English and trying out NLP models using English corpus
- evaluating the work with more data and labels

REFERENCES

[1] S-Kanava. S-Kanava: S-ryhmn asiakasomistajan palvelu. 2019. [Online] Available: https://www.s-kanava.fi/asiakasomistaja

[2] Foodie.fi. S-ryhmn verkkokauppa. 2017. [Online] Available: https://www.foodie.fi/

[3] SAS Institute Inc.. Machine Learning. What it is and why it matters. 2019. Online [Available] https://www.sas.com/en_us/insights/analytics/machine-learning.html

[4] Seif, George. A Guide to Decision Trees for Machine Learning and Data Science. 2018. Online [Available] https://towardsdatascience.com/a-guide-to-decision-trees-for-machine-learning-and-data-science-fe2607241956

[5] Gupta, Prashant. Decision Trees in Machine Learning. 2017. Online [Available] https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052

[6] Donges, Niklas. 2018. The Random Forest Algorithm. Online [Available] https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd

[7] Gionis, Aristides. Slide set 9: Data clustering. 2018. [Online] Available: https://mycourses.aalto.fi/pluginfile.php/843350/mod_resource/content/1/09-clustering.pdf

[8] Arthur, David and Vassilvitskii, Sergei. k-means++: The Advantages of Careful Seeding. 2006. [Online] Available: http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf

[9] Shaikh, Raheel. Feature Selection Techniques in Machine Learning with Python. October, 2018. [Online] Available: https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e

[10] Wikipedia. Cross-validation (statistics). 2019. [Online] Available: https://en.wikipedia.org/wiki/Cross-validation_(statistics)

[11] Garbade, M.J.. A Simple Introduction to Natural Language Processing. 2018. Online [Available] https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32

[12] Athithya, Vijay. Stepping into NLP Word2Vec with Gensim. 2019. Online [Available] https://hackernoon.com/stepping-into-nlp-word2vec-with-gensim-e7c54d9a450a

[13] Brownlee, Jason. How to Develop Word Embeddings in Python with Gensim. 2017. Online [Available] https://machinelearningmastery.com/develop-word-embeddings-python-gensim/

[14] Luotolahti, J., Kanerva, J., Laippala, V., Pyysalo, S. and Ginter, F.. Towards Universal Web Parsebanks. Proceedings of the International Conference on Dependency Linguistics (Depling'15). 2015. Online [Available] https://research.utu.fi/converis/getfile?id=5572857portal=true

[15] Sharma, Natasha. Importance of Distance Metrics in Machine Learning Modelling. 2019. [Online] Available: https://towardsdatascience.com/importance-of-distance-metrics-in-machine-learning-modelling-e51395ffe60d