Department of Informatics
M.Sc in Data science

M.Sc. Thesis

Text classification for the detection of food recalls

Alexandros Chasapis
F3351914

Supervisor: John Pavlopoulos

October, 2020


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Abstract

In this, thesis we experimented with machine and deep learning models in order to apply text classification on food recalls from online announcements. A food recall is defined as:*"Action taken to remove from sale, distribution and consumption foods which may pose a safety risk to consumers"*.[1] Each food recall is a text, which includes the kind of the product that have been removed, the specific product, the kind of the hazard of the product and the specific hazard. Initially, we classified food recalls on the kind of hazard and product type they belong to, using two machine learning models, the Logistic Regression and the Random Forests Classifier. Then, we had to classify each recall on all the different specific products and hazards. For that purpose, we developed deep neural networks, like Recurrent Neural Networks (RNNs) with Long short-term memory (LSTM) architecture and LSTMs with bidirectional strategy, and we compared their performance with two baseline models, the SVM Classifier and a Random Forests Classifier. In every experiment, deep neural networks outperformed the baseline models, especially on the classification on the hazard types.

Περίληψη

Σην παρούσα διπλωματική εργασία, έγινε χρήση μοντέλων μηχανικής και βαθιάς μάθησης με σκόπο την ταξινόμηση κειμένων που σχετίζονται με ανακλήσεις τροφίμων. Η ανάκληση τροφίμων ορίζεται ως: "Δράσεις που λήφθηκαν για την απομάκρυνση από την αγορά, τη διανομή και την κατανάλωση, τροφίμων που ενδέχεται να αποτελούν κίνδυνο για τους καταναλωτές". Κάθε ανάκληση τροφίμου είναι ένα κείμενο που περιέχει το είδος του προϊόντος που ανακλήθηκε, το συγκεκριμένο προϊόν, την κατηγορία του κινδύνου που ανήκει και τον συγκεκριμένο κίνδυνο. Τα πειράματα βασίστηκαν στην ταξινόμηση κάθε ανάκλησης τροφίμου στις προηγούμενες κατηγορίες. Αρχικα, ταξινομήσαμε τις ανακλήσεις στις κυριότερες κατηγορίες κινδύνου και προϊόντος χρησιμοποιώντας δύο μοντέλα μηχανικής μάθησης, ένα Logistic Regression και έναν Random Forests Classifier . Έπειτα, έπρεπε να ταξινομήσουμε τις ανακλήσεις στην ακριβή κατηγορία κινδύνου και προϊόντος. Καθώς το πρόβλημα αυτό είναι αυξημένης δυσκολίας, εκπαιδεύσαμε μοντέλα βαθιάς μάθησης, όπως Ανατροφοδοτούμενα Νευρωνικά Δίκτυα αμφίδρομης ή απλής κατεύθυνσης (LSTM και BiLSTM), και συγκρίναμε την επίδοση τους με μοντέλα μηχανικής μάθησης, έναν SVM και έναν Random Forests Classifier . Σε κάθε πείραμα, τα μοντέλα βαθιάς μάθησης είχαν καλύτερη επίδοση από τα υπόλοιπα, ειδικά στην ταξινόμηση ανακλήσεων στις κατηγορίες κινδύνου.

[1] https://www.foodstandards.gov.au/industry/foodrecalls/recalls/Pages/whatisafoodrecall.aspx


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Acknowledgments

I would like to express my sincere thanks to my supervisor Ioannis Pavlopoulos for his constant support and for his guidance throughout the development of this thesis. Foremost, I would like to thank him for giving me the opportunity to explore the area of Natural Language Processing during my studies. Furthermore, I would like to thank Agroknow and especially Ioannis Stoitsis, as well as Mihalis Papakonstantinou, for their advice and their help, during our video conferences and during my visits to the offices of the company. In addition, I would like to thank my friend Vanessa for the interesting discussions and the support during this year. Last but not least, I would like to thank my parents and every person close to me for their support in every moment of my life.


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Contents

Chapter 1

Introduction

Food recalls are the actions taken to remove from sale, distribution and consumption foods, which are not safe for the consumers.[1] A food recall may be initiated as a result of a report or complaint from a variety of sources, like manufacturers, wholesalers, retailers, government agencies and consumers. It may also occur as a result of a food business's internal testing and/or auditing. Recalls are conducted by food businesses to ensure that potentially hazardous or unsafe foods are not consumed. In our case, we collected food recalls via FOODAKAI platform.[2] FOODAKAI is an intelligent online system that minimizes the food safety risk in the supply chain by offering insights about the hazards that are present in ingredients and products. It collects food recalls from all the international organisations which are related to the food industry and they process the collected of information. FOODAKAI is a cloud subscription-based solution, developed, owned and licensed by Agroknow.[3] Agroknow collects, translates and enriches global food safety data from official and trusted sources in real-time. They harness emerging technologies to achieve greater visibility across the supply chain and they enable companies to make food safety decisions with confidence.

Our task is to find the kind of the product type, the specific product type, the kind of the hazard type and the specific hazard behind each food recall. Figure 1.1 presents an example of a food recall.

Title: Recall of Several Frozen Bakery Products
due to the Presence of The Unauthorised Pesticide Ethylene Oxide.
Date: Friday, 30 October 2020
Country Of Origin: France
Description: Several batches of frozen bakery products are being recalled due to the presence of the pesticide ethylene oxide in sesame seeds used in the product. This pesticide is not authorised for use in foods sold in the EU. The implicated batches were not sold directly to consumers; they have been supplied to food services such as bakeries only.

Figure 1.1: An example of a food recall.

The red highlighted words are the attributes that we need to detect for each

[1] https://www.foodstandards.gov.au/industry/foodrecalls/recalls/pages/whatisafoodrecall.aspx

[2] https://www.foodakai.com/

[3] https://agroknow.com/

food recall. *Bakery products* is the kind of the product, *sesame seeds* is the specific product, *chemical* is the kind of the hazard, which is not written on the recall, but is annotated by FOODAKAI, and *unauthorised substance ethylene oxide* is the specific hazard.

Until recently, analyses presented in food/chemistry related papers were mostly based on classical statistical approaches (Birmpa et al. [14], Bouzembrak et al. [15]). However, today machine and deep learning methods are applied (Eftinov et al. [1], Chin et al. [3], Mezgec et al. [4]).

Initially, we applied classification for finding the kind of the product and the kind of the hazard for each food recall. There are 9 different kinds of hazards and 30 kinds of products. For that task, we trained two machine learning (ML) models, a Logistic Regression and a Random Forests Classifier. Both of them performed really well, but the Logistic Regression brought higher results. As the specific hazard types and the specific product types are numerous, we developed deep learning models, one RNN-LSTM for the classification on all the different hazard types and for the classification on all the different product types a RNN-BiLSTM, because it performed better with a huge number of classes.

There are 1.714 specific hazards and 14.622 specific products. As the different products and hazards are numerous, we decided to apply an experiment where we kept some of the most frequent categories for both product and hazard types. In every experiment, Deep Neural Networks outperformed the baselines and they brought high results, especially at the experiment with the most frequent classes.

1.1 Thesis Outline

The main parts of this thesis are described below:

- **Chapter 2**: We investigate and we discuss the related works to our task. These works are about the food classification and how food classification has been achieved with different methods.
- **Chapter 3**: Here, we present all the different models that we use for the food classification task.
- **Chapter 4**: We describe the experiments that we apply during the thesis.
- **Chapter 5**; We present the results of the experiments.
- **Chapter 6**: We summarise the experiments and the results of them. Furthermore, we discuss the future work related to our task.

Chapter 2

Related work

The problem of food classification has drawn a lot of attention by the scientific community and food industry. Scientists have access to food data from databases related to food, such as FoodEx2, FoodBase and FOODAKAI, which will be described next.. FoodEx2 is a standardised food classification and description system made by the European Food Safety Authority (EFSA).[1] The system consists of descriptions of a large number of individual food items aggregated into food groups and broader food categories in a hierarchical parent-child relationship. FoodBase corpus is a database, which includes annotated food entities. They collected data from "Allrecipes" website, which is the largest food focused social network where everyone plays a part in helping cooks discover and share the home cooking.[2] Below, we present some recent works, that are related to food classification with ML and Deep Learning (DL) methods and techniques.

Eftinov et al. created a semi-automatic system to classify and describe foods according to FoodEx2 [1]. For the classification part, they cleaned the data by applying tokenization and then they created TFIDF features by removing the terms that only appear in at most 1% of the documents. Then, they applied SVM, Random Forests and Max Entropy classifiers to predict the category of each food. After clarifying the category of each food, they described it in terms of FoodEx2. To describe it using FoodEx2 code, Part-Of-Speech (POS) tagging, also called grammatical tagging, was used to identify nouns, adjectives, and verbs. For example, if the food item, "dried vine fruits" (currants, raisins and sultanas), needed to be described with a FoodEx2 code, first POS tagging was applied to extract the nouns, adjectives, and verbs. These sets were then processed by applying lemmatization. The resultant sets were the noun set, which was "vine, fruit, currant, raisin, sultana", the adjective set, which was null for this food item, and the verb set, which was "dry". For each word, that they described before, they tried to find the same term in FoodEx2. For example, the word "dry", returned the terms "fruit soup dry" and "dried vine fruits". The next step was to define the similarities between the food item name and each of the FoodEx2 food item names that belong to the subset. That achieved with the Jaccard similarity index.

[1] https://www.efsa.europa.eu/en/data/data-standardisation
[2] https://www.allrecipes.com/

Eftinov et al. found a vector representation for each food concept [6]. After gathering the data from FoodEx2, they created word embeddings with the method of Poincaré embeddings (Nickel et al. [20]) and they applied text classification with ML models. They compared the results of Support Vector Machines (SVM), Random Forests, Classification Decision Trees (TREE) and Dense Neural Networks (NNET) performed in different dimensionality of Poincaré embeddings.

Popovski et al. focused on Information Extraction (IE) on food entities and they created FoodIE [9]. FoodIE is rule-based Name-entity recognition (NER) system, which works with unstructured textual data. These data are coming from FoodBase, which we introduced before. To develop that, they applied named-entity recognition (NER), which addresses the problem of identification and classification of predefined concepts. They used UCREL Semantic Analysis System (Piao et al. [21]) and coreNLP (Manning et al. [22]) in order to provide word tokens associated with their POS tags and define phrases in the text related to food entities. Furthermore, they created food chunks with a combination of tokens from the two previous methods. They evaluated manually the results of their model, as there was not a method to evaluate such a text corpus. That indicates the lack of NLP tools that can be used for IE of food entities, but also the lack of annotated corpora related to food data.

Dunnmon et al. applied agricultural sentiment analysis on Twitter's data [10]. The term *agricultural sentiment analysis* is used for the predictions of the sentiment of Twitter feeds from farming communities. The main question before the experiment was whether models achieve better performance predicting agricultural sentiment when those models were trained on a smaller, domain-relevant dataset or a richer dataset from one or more unrelated domains. To obtain the smaller target dataset, they designed Twitter queries with specific words (like wheat, lettuce, soybeans etc.) and they extracted tweets of that interest. For unrelated domains, they used publicly available, fully-labeled datasets on sentiment related to movie reviews, the first 2016 GOP debate, and self-driving cars. In order to find the best neural network for this application, they built a pipeline for extensive hyperparameter search across models, ultimately running over 720 distinct models. Their analysis revealed that CNNs outperform and train faster than RNN variants across datasets. They also found that training on smaller, relevant datasets outperformed training on larger, unrelated datasets.

Elizabeth L. Chin et al. developed 9 ML models in order to predict the amount of lactose in a given Automated Self-Administered 24-Hour Dietary Assessment (ASA24) food dataset [3]. For data preprocessing, data were standardized to zero mean and unit variance. The models that were used for that task were LASSO, Bounded-LASSO, Combined LASSO, Ridge, Bounded Ridge, Combined Ridge, Feed Forward Neural Networks (FFNN), XGB-Regressor and Combined XGB. For the evaluation, they used the R2 metric. The FFNN and XGB-Regressor performed better than the rest of the classifiers.

Chapter 3

Methods

3.1 Baseline Models

In this section, we will describe the baseline models that we used for our experiments. Most of them were used in order to compare the results with our main deep neural networks. These are the Logistic Regression, the Random Forests Classifier and the Support Vector Machines (SVMs).

3.1.1 Logistic Regression

`Logistic Regression` is a supervised learning classification algorithm used to predict the probability of a target variable [25]. A type of `Logistic Regression` is the binary `Logistic Regression`. The *binary* classification happens when the dependent variable has only two different types (e.g. 0 or 1, positive or neutral). At *multinomial* classification, the variable could take more than 3 unordered types and at the *ordinal*, the variable takes 3 or more ordered types.

In our task, we had to develop `Multinomial Logistic Regression` classifiers, as the hazard and the product types are more than 1.000. In order to map the predicted values to probabilities, `LR` uses the sigmoid function. The sigmoid function maps any real value into another value between 0 and 1.

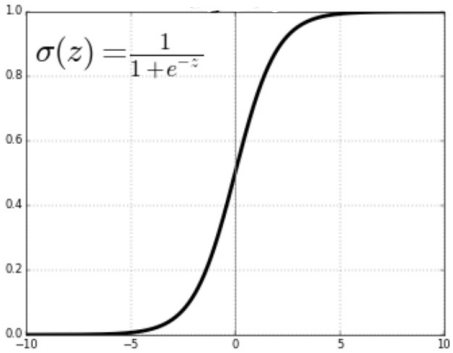

Figure 3.1: The sigmoid function.

For the *Multinomial Logistic Regression*, let's assume that we have $c_1, c_2, c_3, ..., c_k$ different classes and the input vector $\vec{x}$. Also, for each class c_i, we learn a linear separator with its own weights vector $\vec{w}_j$. In order to find in which class $\vec{x}$ belongs to, we have to calculate the below formula:

$$P(c_i \mid \vec{x}) = \frac{e^{\vec{w}_j \cdot \vec{x}}}{\sum_{i=0}^{k} e^{\vec{w}_i \cdot \vec{x}}} \tag{3.1}$$

for all the classes c_i. We then assign the class that corresponds to the highest probability.

3.1.2 Random Forest Classifier

Random forest is a supervised learning algorithm [26]. The "forest" it builds, is an ensemble of decision trees, usually trained with the bagging method. The general idea of the bagging method is that a combination of learning models increases the overall result.

The training algorithm applies the technique of bagging. Let's assume that we have a training set $X = \vec{x_1}, \vec{x_2}, \vec{x_3}, ..., \vec{x_n}$ and the responses of X are $Y = y_1, y_2, ..., y_n$. Then, it is bagging repeatedly (B times) and it selects a random sample with replacement of the training set and fits trees to these samples.

After training, predictions for unseen samples $\vec{x}$ can be made by averaging the predictions from all the individual regression trees on $\vec{x}$ with the below formula:

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(\vec{x}) \tag{3.2}$$

3.1.3 Support Vector Machines (SVMs)

Support Vector Machines is a supervised learning algorithm mostly used for classification [27]. The main idea is that based on the labeled data the algorithm tries to find the optimal hyperplane which can be used to classify new data. SVMs can be a linear classifier, when the classification task in binary, but it can be nonlinear classifier by applying kernel tricks.

3.2 Recurrent Neural Networks

Each Neural Network (NN) consists of *neurons*, *connections*, *weights* and *propagation functions*. There are different kinds of NNs based on the structure of the previous elements. Three of the most common NNs are the Multi-layer Perceptrons (MLPs), the Recurrent Neural Networks (RNNs) and the Convolutional Neural Networks (CNNs).

RNNs are NNs and they process the input data with sequences [28]. RNNs take one input at a time and they keep a type of short-term "memory cell", which contains all the information from the previous actions. It means that in time t, that cell

contains information from the previous time periods $(t-1, t-2, t-3, ...)$. Moreover, they are called recurrent because they perform the same task for every element of a sequence, with the output being dependent on the previous computations. In Figure 3.2 you can see an unfolded and a folded architecture of RNN.

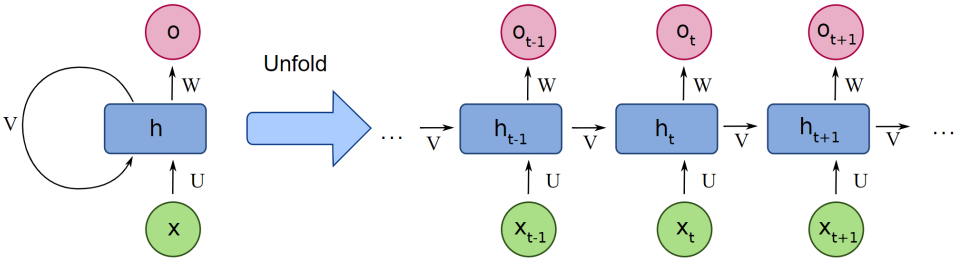

Figure 3.2: The RNN's architecture.[1]

Figure 3.2 shows an RNN being unfolded into a full network. Let assume that the input of the RNN is a sentence with 20 tokens. According to the diagram, the RNN would be unfolded into 20-layer NN, one layer for each token. The expressions written on the above diagram have the shape of matrices and below is the explanation of them.

1. x_t is the input at step t. For example, x_3 could be a TFIDF vector corresponding to the third token of a sentence.
2. h_t is the hidden state at time step t. It's the "memory" cell of the RNN. h_t is calculated based on the previous hidden state and the input at the current step.
3. o_t is the output of the RNN at the time t.
4. U is the transformation matrix of input x.
5. V is the weight matrix that transform h_t to h_{t+1}.
6. W is the matrix which transforms the computed state h_t to the output o_t.

3.2.1 Bidirectional RNNs

Figure 3.3 depicts a Bidirectional RNN model.

Two RNNs are stacked on top of each other, one going from the beginning to the end and the other from the end to the beginning. The output is computed based on hidden states of both networks (e.g., concatenating the two hidden states that correspond to a single token).

[1] https://medium.com/deep-learning-recurrent-neural-networks
[2] https://missinglink.ai/guides/neural-network-concepts/recurrent-neural-network-glossary-uses-types-basic-structure/

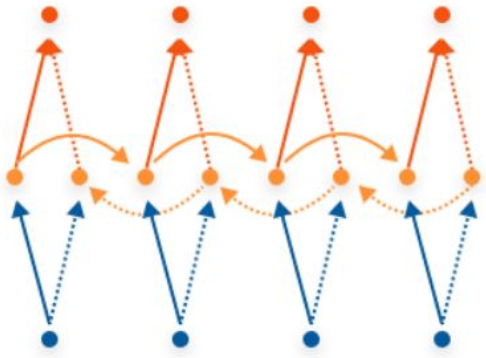

Figure 3.3: The Bidirectional RNN's architecture. [2]

3.2.2 RNN's main formulas

The main formula the RNN is based on is that of the hidden state h_t. This is :

$$h_t = f(h_{t-1}, x_t) \tag{3.3}$$

From the above equation it is clear that each RNN hidden state is related to the previous one. Function f can be linear or non-linear, such as a tanh, a sigmoid or a ReLU function. Formula 3.3 could be written more analytically:

$$h_t = f(V \cdot h_{t-1} + U \cdot x_t) \tag{3.4}$$

Furthermore, the output at time t is defined as the below formula:

$$o_t = g(W \cdot h_t) \tag{3.5}$$

where g is the output layer function, which can again be a a tanh, a sigmoid or a ReLU function.

3.2.3 Dropout

In machine learning, regularization is a way to prevent over-fitting. Regularization reduces over-fitting by adding a penalty to the loss function. Dropout is an approach to regularization in neural networks which helps reducing interdependent learning amongst the neurons (Strivastava et al. [23]). Dropout refers to "freezing" units during the training phase of certain sets of neurons which is chosen at random. By "freezing", we mean that these units are not valuable during the training at that specific moment. At each training step, some nodes are randomly selected and they dropped out with probability $1 - p$.

3.2.4 Long Short-Term Memory (LSTMs) models.

They were introduced by Hochreiter and Schmidhuber (1997), in order to avoid the long-term dependency problem [19]. We need to focus on the architecture of an LSTM network, as they are the main models for our classification task.

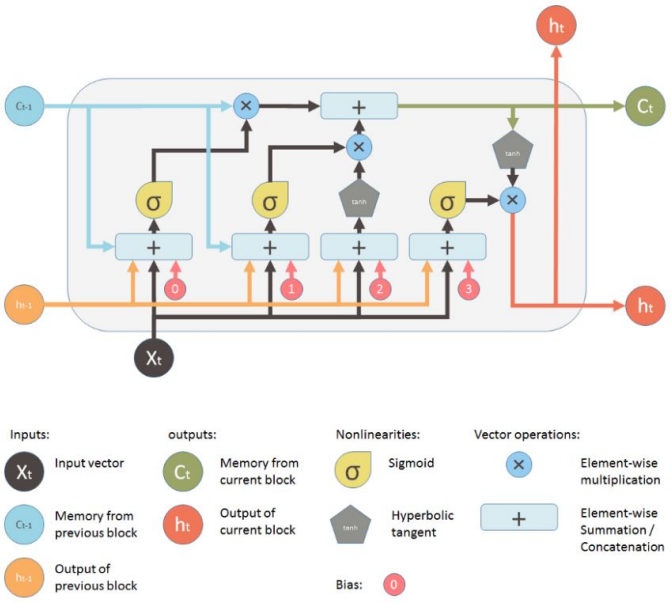

Figure 3.4: The LSTM architecture.[3]

Initially, we will discuss the inputs and the outputs of an LSTM. It takes three different inputs. The first one is the x_t (the current input at time t), the second is the h_{t-1} that is the output of the hidden state of the RNN at the time $t-1$ and the third one is the c_{t-1}, which is the long-term memory cell from the previous unit.

Figure 3.5 shows the first functionality of the LSTM. It takes as input the memory cell c_{t-1} from the previous unit and it filters that cell based on the memory that it wants to keep. For example, if the old memory c_{t-1} is multiplied with a vector that is close to 1, it means that it wants to keep almost all the memory from the previous unit. The second action applies a piece-wise summation. It is the part that merges the old memory cell with the new one. It answers the question: How much new memory should be added to the old memory when a new input is added. After these operations, LSTM produces the new output c_t.

[3] https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714

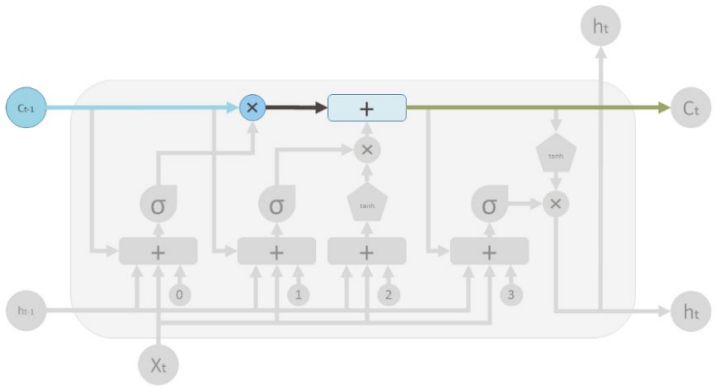

Figure 3.5: First action of LSTM.[4]

Now, we will explain how the "valves" (blocks with sign + from the figures) of LSTM work 3.6. The first one is the valve that applies the "forget" operation. It consists of a one-layer NN and it takes as input the h_{t-1} from the previous hidden state, the current input x_t and the old long-term memory c_{t-1} and a bias term b_0. The activation function of this NN is a sigmoid function and the output is the forget valve, that we explained before.

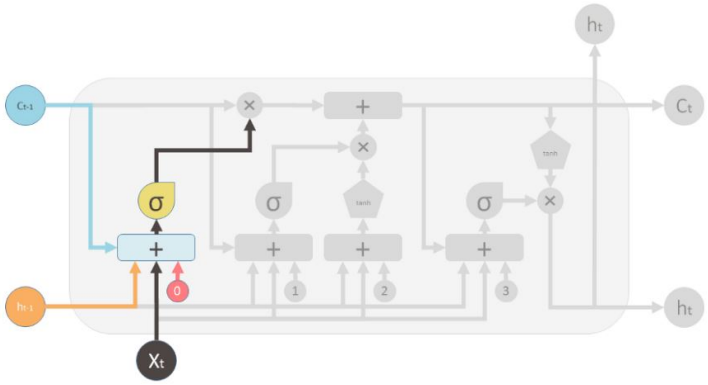

Figure 3.6: Second action of LSTM.[5]

The second valve is called the new memory valve (fig. 3.7). It is a one-layer NN that has the same inputs as previously. This gate controls how much old memory will be added to the new memory. But, the new memory is produced by another NN (third valve) with the tanh as an activation function. The output of this NN will element-wise multiple the new memory valve, and add it to the old memory in order to create the new memory.

[4] https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714
[5] See footnote 4.
[6] See footnote 4.

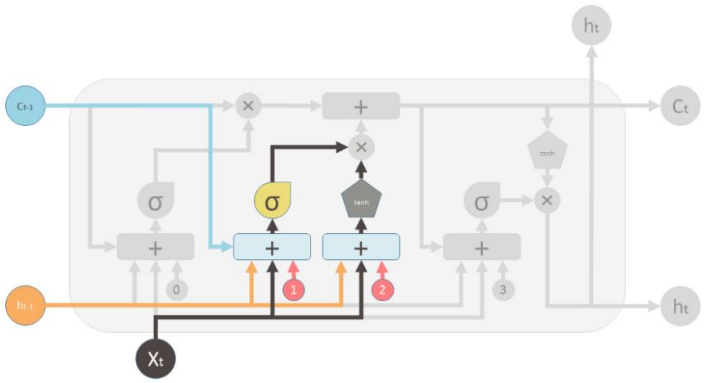

Figure 3.7: Third action of LSTM.[6]

For the last part 3.8, we have to produce the output of that LSTM unit. That valve is again a simple one-layer NN, with the same inputs as before (h_{t-1},x_t and a bias term). The activation function is a sigmoid function. The output is the valve that decides how much memory it should be transferred to the next LSTM unit.

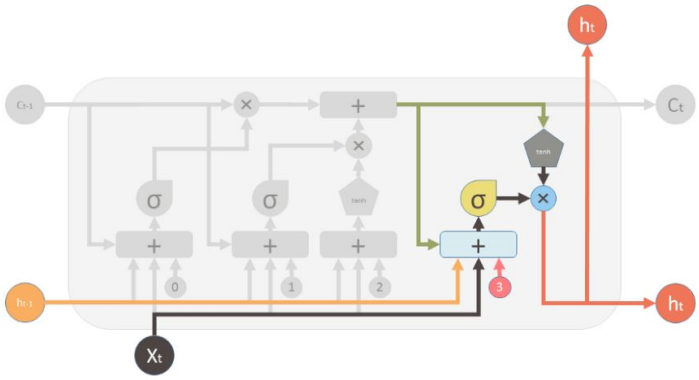

Figure 3.8: Last action of LSTM.[7]

[7] https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714

Chapter 4

Experiments

4.1 Dataset Exploration

In our research, data were retrieved from FOODAKAI. FOODAKAI gathers food recalls from the below organizations:

Organisations	
Abu Dhabi Food Control Authority	Australian Competition and Consumer Commission
Australian Department of Agriculture Imported Food Reports	Austrian Food Safety Authority
Brazilian Health Regulatory Agency	FDA Enforcement Reports
California Department of Public Health	Canadian Food Inspection Agency
Carrefour France	Centers for Disease Control and Prevention (CDC)
Centre for Food Safety Hong Kong	Competition, Consumer Affairs and Fraud Control Authority of France
Croatian Food Agency	Czech Agriculture and Food Inspection Authority
EU Knowledge centre for food fraud and quality	FDA
FDA Import Alerts	FDA Import Refusals
Federal Agency for the Safety of the Food Chain of Belgium	Food Safety and Standards Authority of India
Food Safety Authority of Ireland	Food Safety Authority of Luxembourg
Food Standards Australia New Zealand	Food Standards Scotland
FreshDirect USA	FSIS Import Refusals
FSIS USDA	German Federal Office of Consumer Protection and Food Safety
Greek National Drug Organization	Hellenic Food Authority (EFET)

Italian Ministry of Health	Ministry for Primary Industries New Zealand
Ministry of Agriculture, Livestock and Farming of Brazil	Ministry of Agriculture, Nature and Food Quality of the Netherlands
Ministry of Environment and Food of Denmark	Ministry of Food and Drug Safety of South Korea
Ministry of Health Israel	Ministry of Health, Labour and Welfare Japan
Nigerian National Agency for Food and Drug Administration Control	Polish State Sanitary Inspection
Public Health Agency of Canada	RASFF
Republic of China Import Refusals	Republic of Slovenia Ministry of Agriculture, Forestry and Food
Singapore Food Agency	UK Food Standards Agency

The dataset includes incidents (food recalls) since 1980 (the oldest incident on the platform) till today. FOODAKAI adds every day the daily incidents, so our dataset includes incidents till the 15th of July 2020. Currently, this dataset is owned by FOODAKAI and is not publicly available. However, we were given access to the data for academic purposes.

Each incident is described by the following features:

- **id**: The unique ID of the incident.
- **title**: The title of the incident.
- **description**: The proper description of the incident.
- **publication date**: The official date of the publication of the incident.
- **remote title**: A short brief of the description.
- **kind of product**: The kind of the product, which have been recalled.
- **specific product type**: The specific recalled product.
- **kind of hazard**: It describes the generic kind of the hazard.
- **specific hazard type**: The specific hazard type of the product that have been recalled.

Our initial goal is to apply classification on the description of each incident.

After removing English stopwords we found that our vocabulary consists of 52.685 different words and some of the most frequent are: *product*, *food*, *kg*, *recall* and *consumers*[1].

[1] https://gist.github.com/sebleier/554280

4.1.1 Hazard Types

Each incident is labeled with the kind of hazard, the specific hazard type, the kind of product and the specific product that belongs to. Our first task was to develop a text classifier to detect the kind of hazard of each incident. All incidents are classified into 9 kinds of hazard. These are: *Chemical*, *Biological*, *Fraud*, *Organoleptic Aspects (OA)*, *Allergens*, *Food Contact Materials (FCM)*, *Foreign Bodies (FB)*, *Food Additives and Flavourings (FAF)* and *Other Hazard (OH)*

- **Chemical**: Hazards as a result of contamination by chemical substances (e.g. fertilisers) and biological toxins (1. naturally present in food, 2. additives e.g. pesticides)
- **Biological**: They include hazards associated with the development of microorganisms, in particular: bacteria, yeasts/fungi, viruses and parasites.
- **Fraud**: Hazards as a result of adulteration (EMA), counterfeit, intentional distribution of products not fit for consumption, unauthorised ingredients, false labelling, misdescription, smuggling and other not classified fraud hazards (ex. too high content of water, absence of certified analytical report and others)
- **Organoleptic Aspects**: The organoleptic characteristics of food are perceived by vision, hearing, smell, taste and tactility. Any change which does not meet the consumer's expectations, as a result of these characteristics, is considered unacceptable.
- **Allergens**: Substances capable of inducing allergy or specific hypersensitivities, such as milk, eggs, wheat or nuts.
- **Food Contact Materials**: Actions taking part between the food and its package. In particular, the migration of a packaging material to the food, a defective packaging affecting the food or the incorrect product dosage in the package.
- **Foreign Bodies**: Substances or small foreign objects that can be found in food, due to poor processing, maintenance, treatment or storage of the product. (ex. a piece of glass/plastic, pesticides, insects).
- **Food Additives and Flavourings**: They are natural or synthetic substances, which are deliberately added to food or beverages, in order to give them specific functional properties or characteristics, such as maintenance, stability and/or improvement of the organoleptic characteristics of the food or drink. They are also substances that enhance the existing taste and/or smell of the food
- **Other Hazard**: Hazards such as feed additives, GMO's, insufficient controls, processing, tses, radiation, novel foods, suffocation risk, traceability systems and others.

We have in our disposal **99.000** incidents. Figure 4.1 shows the percentage and the absolute frequency of the incidents that allocated for each main hazard type.

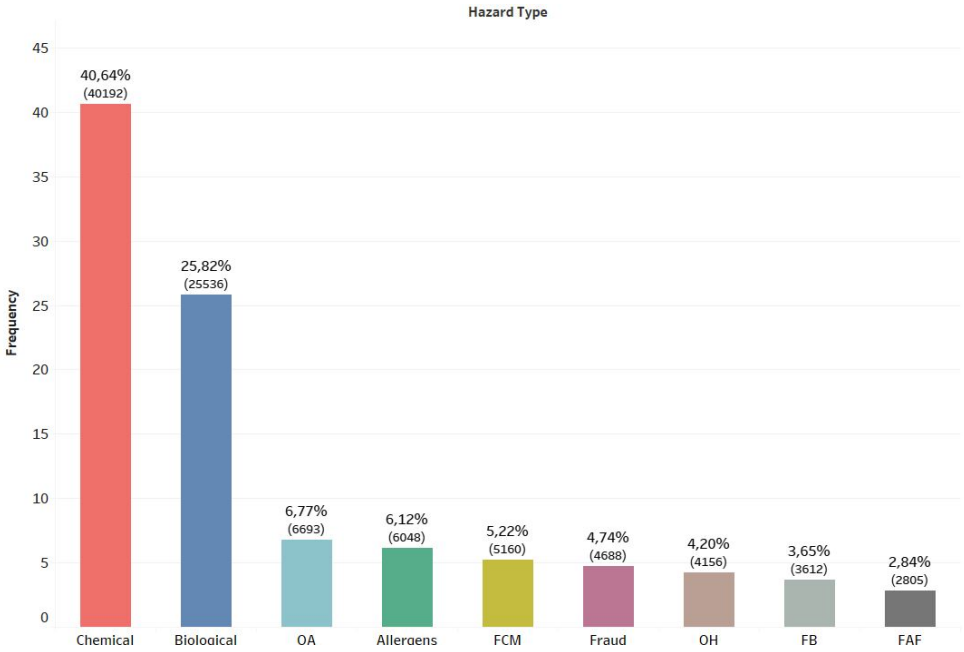

Figure 4.1: Dataset labels' Distribution

Now, we calculated the frequency of each word of the vocabulary for each main hazard type, e.g. *How frequent is the word aflatoxin on each hazard type?* That frequency was divided by the number of the total tokens of our corpus and it was multiplied by 100. Therefore, we created a dataframe that shows us the frequency of each word of the vocabulary on each main hazard type. For example the frequency of words *product* and *food* on each kind of hazard is:

	product	*food*
chemical	0.007	0.006
biological	**0.36**	**0.27**
OA	0.01	0.17
allergens	**0.33**	0.17
FCM	0.01	0.01
fraud	0.07	0.09
OH	0.05	0.06
FB	0.2	0.1
FAF	0.05	0.09

Table 4.2: Frequency of words *food* and *product* on each main hazard type.

As we have calculated the frequencies for all the words on each main hazard type, we can find the correlation between these kinds of hazard based on the frequency of the vocabulary words.

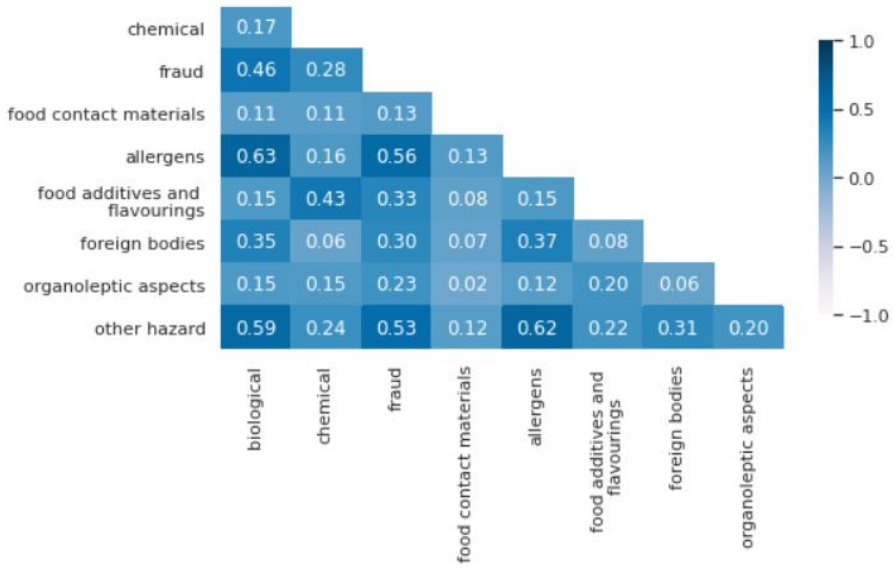

Figure 4.2: Correlation between hazard types based on the words frequency.

As you can see from Figure 4.2, the categories *allergens* and *biological* are highly correlated (correlation equal to 0.63) in terms of the relative frequency of the words of our vocabulary. On the other hand, the categories *organoleptic aspects* and *foreign bodies* are not highly and positively correlated (correlation equal to 0.06).

4.1.2 Product Types

In terms of products type, there is a hierarchy similarly to hazard types. At the first level of the hierarchy, there are 30 different kinds of products. Below, you can see these 30 different classes and the percentage of incidents that allocated for each main product type.

- fish and fish products (**11.7%**)
- cereals and bakery products (8.5%)
- milk and milk products (2.8%)
- herbs and spices (8.4%)
- fruits and vegetables (**15.4%**)
- feed additives (0.4%)
- cocoa and cocoa preparations, coffee and tea (2.6%)
- prepared dishes and snacks (2%)
- dietetic foods, food supplements, fortified foods (2.8%)
- nuts, nut products and seeds (**12.1%**)
- meat and (other than poultry) (9.4%)
- non-alcoholic beverages (1.7%)
- food additives and flavourings (0.01%)
- poultry meat and poultry meat products (3.9%)
- food contact materials (1.5%)
- alcoholic beverages (0.5%)
- confectionery (1.9%)
- honey and royal jelly (0.5%)
- soups, broths, sauces and condiments (1.4%)
- other food product / mixed (0.6%)
- eggs and egg products (0.6%)
- feed materials (0.1%)
- pet feed (0.1%)
- ices and desserts (0.7%)
- bivalve molluscs and products therefor (1.2%)
- fats and oils (0.9%)
- crustaceans and products thereof (5.4%)
- cephalopods and products thereof (1.3%)
- gastropods (0.06%)
- sugars and syrups (0.1%)

4.2 Experimental Settings

Before developing any type of classification, we need to split the dataset into three sets: *training set*, *development set* and *test set*.

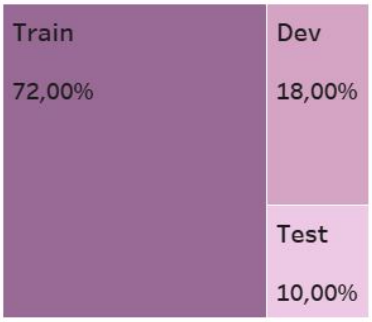

Figure 4.3: FOODAKAI Dataset Split

After removing all the numbers and the stopwords from our corpus, we could gain some descriptive information for the vocabulary of the FOODAKAI dataset.

	Train	Dev	Test	Corpus
avg length	37	35	38	36
# vocabulary size	41,806	11,732	6,855	56,165
# tokens	2,601,864	626,449	375,746	3,604,059

Table 4.3: Additional Information for FOODAKAI dataset and vocabulary.

4.2.1 HAZARD Classification

Our first task is to classify each food recall on the 9 main hazard types.

A) Classification on the top level of the hierarchy

For that task, we applied a Logistic Regression and a Random Forests classifier. Moreover, we needed to define the input of these classifiers. For that purpose, we decided to try three different vertical representations. The first was the term frequency–inverse document frequency (TFIDF) vectors. The second is the Bag of Words representation and the third is using the FastText word embeddings [29].

Inputs

1. **TFIDF vectors**
   These vectors contain the $TF_i \cdot DF_i$ score for each word w_i of the vocabulary. TF_i score indicates how frequent the word w_i is in the text. IDF_i shows how rare the word w_i is in the language. The mathematical expressions for the previous terms are:

$$TF_i = \frac{DF_i}{N_{doc}} \qquad \text{and} \qquad IDF_i = \log(\frac{N_{doc}}{DF_i}),$$

   where N_{doc} is the number of tokens in the corpus

   To create these vectors, we used TfidfVectorizer[2] from scikit-learn. We did hyper-parameter tuning in the size of the TFIDF vector and we found that each vector should have 20.000 features. Moreover, we used unigrams and bigrams as well for the vector representation.

2. **Bag of Words**
   In this method, a text (such as a sentence or a document) is represented as a bag (multiset) of its words, disregarding grammar and word order. To create these vectors, we used CountVectorizer[3] from scikit-learn. In our approach, we created boolean vectors with 50.000 features. We counted also unigrams at first and then the bigrams as well.

3. **FastText**
   FastText extends Word2Vec by using character n-gram embeddings and forming the vector of each word by summing its n-grams [30]. Hence, it also considers morphology and it can produce vectors for out of vocabulary words. As we downloaded the pre-trained (on Wikipedia) word vectors, we calculated the cendroids of each word and we created the vector representation.

Models

1. **Logistic Regression**
   For our case, we used solver `lbgs` for the optimization algorithm, as it is a multiclassification task.[4] Moreover, we chose the `multinomial loss`, because we have to deal with a multiclass classification task. After hyperparameter tuning, we set C = 2, where C is the inverse regularization parameter. Finally, we turned on the `"balanced"` mode, which adjusts the weights to class frequencies in the input data and that should be very efficient for our imbalanced dataset.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[3] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

[4] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

2. **Random Forest Classifier**
   We set equal to 2 the minimum number of samples that required to be at a leaf node. Also, the maximum depth of the tree is 46.

B) Classification on all the leafs of the hazard hierarchy

Next, we discuss the experimental settings when we investigated the classification on all the leafs of the hierarchy. The first level includes only 9 main hazard types. FOODAKAI has defined 1714 different hazard types. The first experiment was applied in all the 1714 different classes. As we were concerned that 1714 were too many for a classifier, we experimented also with less. We kept the classes with support greater than 10 (i.e., more than ten incidents assigned with that class). Figure 4.4 presents the cumulative distribution of the classes, based on their support.

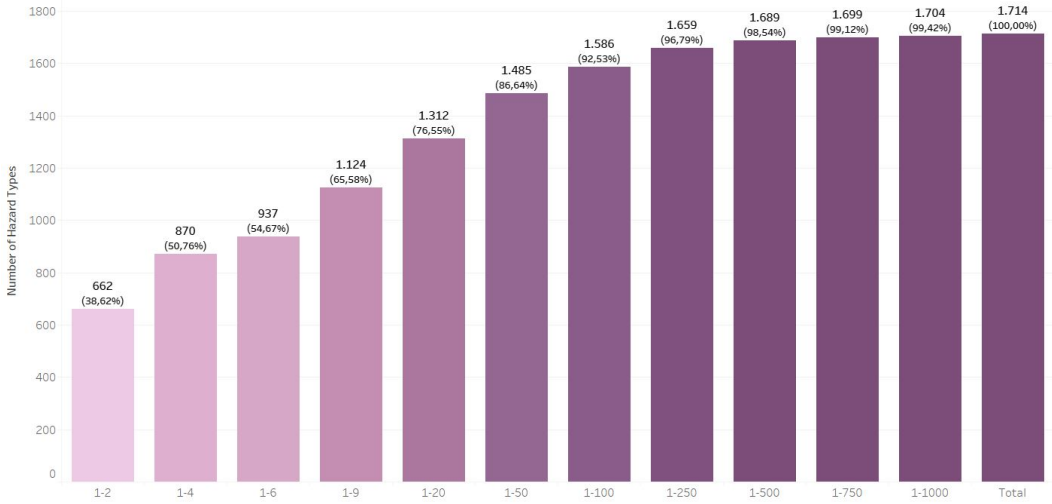

Figure 4.4: Cumulative distribution of the hazard types based on their support.

Figure 4.5 shows the 1124 hazard types with support less than 10. After the elimination of these 1124 hazard types, we had 590 different hazard types.

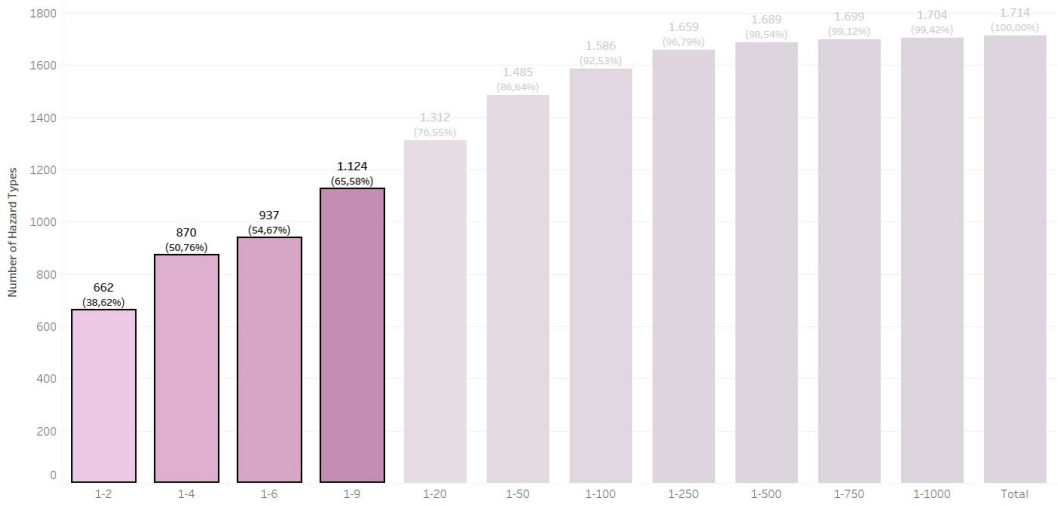

Figure 4.5: The hazard types with support less than 10.

	Train	Dev	Test	Total
Experiment C1714	80,190	8,910	9,900	99,000
Experiment C590	77,539	8,615	9,572	95,726

Table 4.4: Dataset's split in both experiments.

We need to mention that in our case, each incident belongs to only one hazard type, so, we had to deal with a multi-classification task. For these experiments we tried three different models, two baseline models and one deep neural network. Similarly to Section 4.2.1, the baseline models were: the *Random Forests Classifier* and the *Stochastic Gradient Descent Classifier (SGD Classifier)* implemented by Agroknow. The deep neural network was a RNN with an LSTM architecture.

Inputs

1. For the baseline models, we used as inputs the methods that we described before: the `TFIDF` vectors and the `Bag of Words` representations.

2. In order to define the input of the RNN-LSTM model, we needed to create pad sequences for each input `X`. The length of it was 1,000. After that, we set equal to 60,000 the number of the distinct words of the vocabulary and we enabled the Embedding Layer with dimension 300 at the first level of the architecture of our model.

Models

1. **Random Forest Classifier**
   We set the minimum number of samples required to be at a leaf node equal to 2. Also, the maximum depth of the tree was 80.

2. **SGD Classifier**
   The training of the model was implemented by Agroknow and we do not have in our disposal the parameters of it.

3. **RNN-LSTM**
   Initially, we needed to create an embedding layer that would transform the input `X`. We mentioned before that the dimension of the embedding layer is 300. After that layer, we applied a Spatial Dropout (1d) with a rate equal to 0.2. Now, the LSTM layer used as an activation function the hyperbolic tangent (`tanh`). Also, we set dropout equal to 0.2. The same value (0.2) was chosen for the recurrent dropout. The number of hidden units whose activation get sent forward to the next time step space is 100. We need to mention that the model was trained with 10 epochs.

4.2.2 PRODUCT Classification

The first experiment was about to classify each incident into the 30 main product types.

A) Classification on the top level of the hierarchy

For that task, we applied the techniques that we used before. We implemented two ML models, one *Logistic Regression* and one *Random Forest Classifier*. Furthermore, as inputs for these models, we used the same vertical representations as before. The first one is the *TFIDF* vectors and the second one is the *Bag of Words* representation.

B) Classification on all the leafs of the product hierarchy

We had to classify each incident on all the leafs of the hierarchy. FOODAKAI platform includes 14.266 different product leafs. For that purpose, we applied two experiments. The first experiment included product types with support more than 10 and the second with support more than 50.

Figure 4.6 show the cumulative distribution of the number of product types per their support.

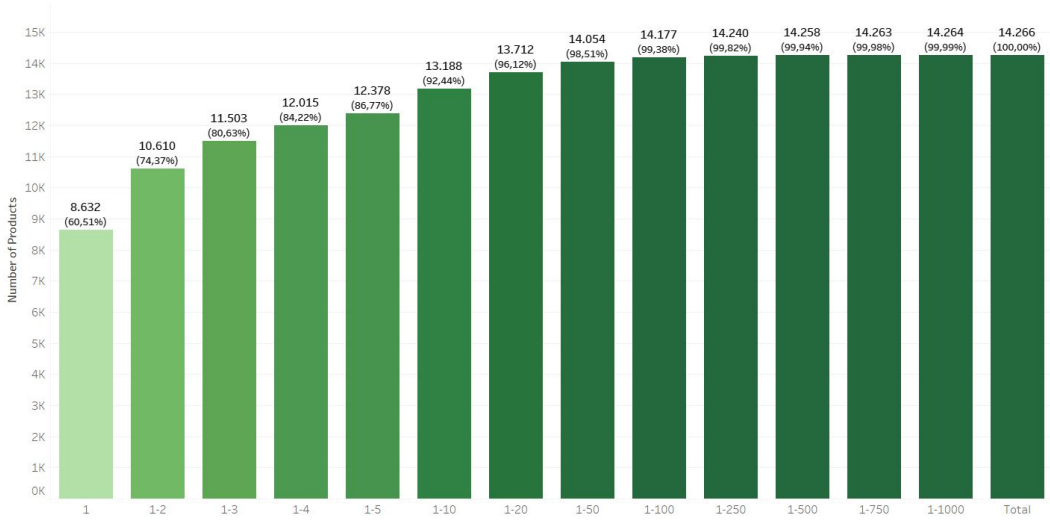

Figure 4.6: Cumulative distribution of the 14.266 product types per their support.

The first experiment includes the products with support more than 10.

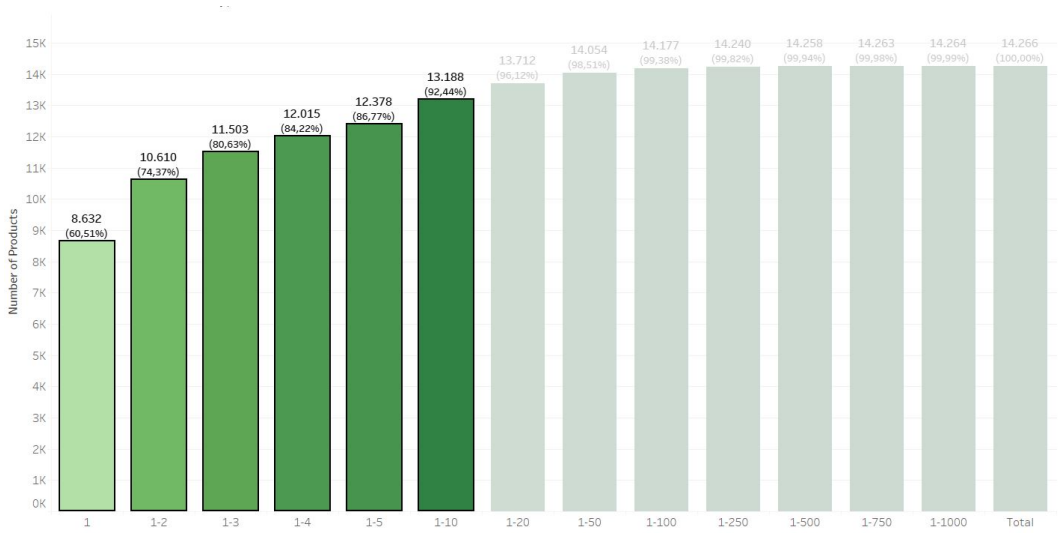

Figure 4.7: The product types with less than 10 incidents.

Figure 4.8 shows the 13,188 product types, which include less than 10 incidents. So, for the experiment, we did not take into consideration these product types and we kept the 1,078 most frequent classes.

The second experiment takes as classes the products with support more than 50.

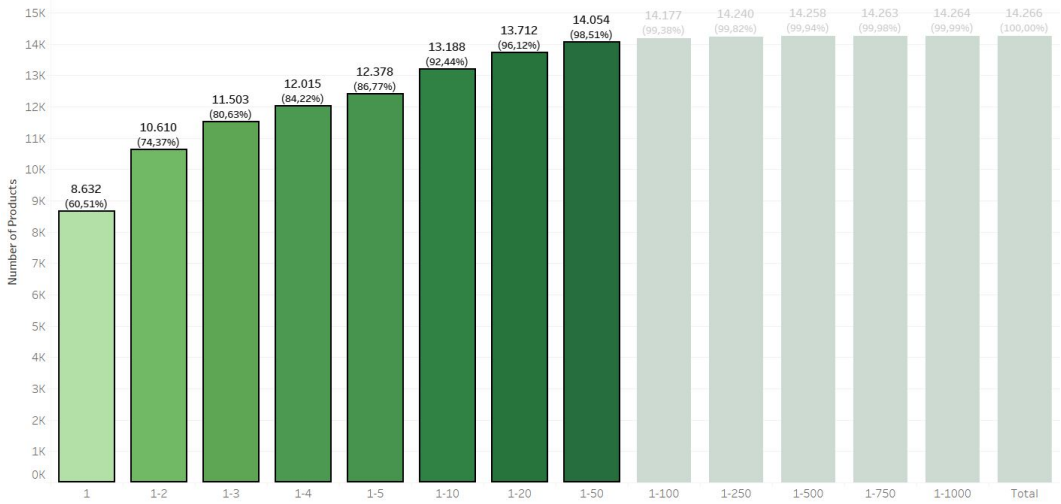

Figure 4.8: The product types with less than 50 incidents.

At this point, 14,054 products have support less than 50. Thus, for that experiment we took 212 different product types.

	Train	Dev	Test	Total
Experiment C1078	41,758	4,639	5,155	51,552
Experiment C212	26,805	2,978	3,309	33,092

Table 4.5: Dataset's split in both experiments.

For these experiments, we applied three different models, two baseline models and one deep neural network. The baseline models are: a *Random Forest Classifier* and a *Support Vector Machines (SVMs)* implemented by Agroknow. The deep neural network is a RNN with BiLSTM architecture.

Inputs

The inputs for the baseline models are the same as we did during the classification on the hazard types. In order to define the input of the RNN-BiLSTM model, we created pad sequences for each input *X*. The length of it was 700. We set equal to 50,000 the number of the distinct words of the vocabulary. We enabled the Embedding Layer with dimension 300 at the first level of the architecture of our model.

Models

1. The baseline models, the (*Random Forest Classifier* and the *SVM*) were described at the classification task on the hazard types.

2. **RNN-BiLSTM**
   Firstly, we created an embedding layer that would transform the input X. We

mentioned before that the dimension of the embedding layer is 300. After that layer, we applied a Spatial Dropout (1d) with a rate equal to 0.2. The BiLSTM layer used as an activation function the hyperbolic tangent (tanh). Also, we set the dropout [20] equal to 0.3. The same value (0.3) was chosen for the recurrent dropout. The number of hidden units whose activation gets sent forward to the next time step space is 100. We need to mention that the model was trained with 20 epochs.

Chapter 5

Results

5.1 Evaluation Metrics

In order to compare the results and evaluate our models, we used *accuracy score*, *f1-score* and *Under Area Precision-Recall Curves (AUPRC)*.[1]

- **Accuracy-score** is defined as the summation of true positives (T_p) and of true negatives (T_n) over the summation of true positives and of true negatives and of false positives (F_p) and of false negatives (F_n).

$$accuracy = \frac{T_p + T_n}{T_p + F_p + T_n + F_n} \tag{5.1}$$

- **Precision (P)** is defined as the number of true positives (T_p) over the number of true positives plus the number of false positives (F_p).

$$P = \frac{T_p}{T_p + F_p} \tag{5.2}$$

- **Recall (R)** is defined as the number of true positives (T_p) over the number of true positives plus the number of false negatives (F_n).

$$R = \frac{T_p}{T_p + F_n} \tag{5.3}$$

- **f1-score** is defined as the harmonic mean of precision and recall.

$$f1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{5.4}$$

AUPRC is typically used in binary classification to study the output of a classifier. In order to extend the precision-recall curve and average precision to multi-class or multi-label classification, it is necessary to binarize the output. Precision-Recall curves summarize the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds.

[1] https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html

5.2 Results on Hazard classification

5.2.1 Top level of hierarchy

Initially, we will present the results of our models at the highest level of the hierarchy, where we had to classify each recall on the 9 main hazard types. We need to find the model and the vector representation that performs better for that task. In table 5.1 you can see the results of the different models based on the accuracy score.

Model	**Accuracy**
LR with TFIDF	0.94
LR with BoW	**0.97**
LR with FastText	0.85
RF with TFIDF	0.88
RF with BoW	0.91
RF with FastText	0.79

Table 5.1: Accuracy on test set.

We see that Logistic Regression with BoW vectors as input performed better than the other classifiers. We noticed also, that FastText word embeddings did not perform that well for that task. One of the reasons is that our vocabulary is related to food and chemistry data, and the FastText embeddings were trained on a more wide dataset, the Wikipedia's dataset.

Now, we calculated the f1-score and the AUPRC for each main hazard category based on the best model that we found before (Logistic Regression).

Hazard Type	**f1-score**	**AUPRC**
Allergens	0.94	0.97
Chemical	0.97	0.98
Biological	**0.98**	0.98
Food additives and flavourings	0.95	0.97
Foreign Bodies	0.97	**0.99**
Fraud	0.92	0.96
Food contact materials	0.90	0.95
Other hazard	0.95	0.97
Organoleptic aspects	0.95	0.98

Table 5.2: F1-score and AUPRC on test set.

5.2.2 All the leafs of hazards

Now, we will see the performance of our models on all the specific hazards (1714 different hazards). It is very difficult to compare the results between 1714 categories, we grouped these classes based on the number of incidents that there are in each class.

Groups	**Support**
Group 1	0-1
Group 2	2-5
Group 3	6-0
Group 4	11-20
Group 5	21-50
Group 6	51-100
Group 7	101-300
Group 8	301-600
Group 9	600+

Table 5.3: Grouped hazard types based on the incidents.

We need to mention that we applied two experiments on that classification task. At the first one, we applied three different models: a Random Forest Classifier (`RF C1714`), a RNN-LSTM (`LSTM C1714`) and a SGD Classifier (`SVM-AGR C1714`) by Agroknow for all the 1714 classes. During the second experiment, we kept the 590 most frequent classes and we used the same models (`RF C590`, `LSTM C590` and `SVM-AGR C590`). Below, you will see the mean f1-score and the mean AUPRC for each of the groups that we described before.

Grouped Classes	**RNN-LSTM C1714**	**RF C1714**	**SVM-AGR C1714**	**RNN-BiLSTM C590**	**RF C590**	**SVM-AGR C590**
0-1	0.09	0.02	0.04	0.45	0.13	0.22
2-5	0.60	0.30	0.39	0.81	0.34	0.48
6-10	0.75	0.63	0.53	0.82	0.67	0.67
11-20	0.84	0.79	0.73	0.87	0.77	0.72
21-50	0.84	0.81	0.71	0.88	0.86	0.82
51-100	0.88	0.84	0.73	0.90	0.84	0.78
101-300	0.90	0.86	0.79	0.93	0.91	0.85
301-600	**0.98**	**0.96**	0.73	0.98	**0.96**	**0.95**
600+	0.97	0.88	**0.89**	**0.99**	0.93	0.93

Table 5.4: Mean f1-score for grouped hazard types.

Grouped Classes	**RNN-LSTM C1714**	**RF C1714**	**SVM-AGR C1714**	**RNN-LSTM C590**	**RF C590**	**SVM-AGR C590**
0-1	0.08	0.02	0.04	0.44	0.13	0.22
2-5	0.50	0.25	0.32	0.73	0.29	0.41
6-10	0.64	0.52	0.43	0.73	0.56	0.57
11-20	0.74	0.67	0.62	0.79	0.66	0.61
21-50	0.73	0.68	0.57	0.79	0.75	0.70
51-100	0.79	0.72	0.59	0.81	0.72	0.64
101-300	0.83	0.75	0.63	0.87	0.83	0.75
301-600	**0.97**	**0.93**	0.71	0.97	**0.93**	**0.90**
600+	0.95	0.80	**0.82**	**0.98**	0.87	0.88

Table 5.5: Mean AUPRC for grouped hazard types.

It is clear, that all the models achieved really high scores in Group 7, Group 8 and Group 9. These groups consist of the hazard types with the most incidents per hazard type. Furthermore, the RNN-LSTM outperformed the other two baseline models. Another interesting point is that all the models had better performance at the second experiment, where we kept the 590 most frequent classes.

5.3 Results on Product classification

5.3.1 Top level of hierarchy

For that part of our task, we decided to keep the model that outperformed before at the first level of the hierarchy of the hazard types. That was a *Logistic Regression* with input a BoW vertical representation. In table 5.6 you can see the f1-score and AUPRC for each main product type.

Product Type	**f1-score**	**AUPRC**
Fish and fish products	0.90	0.96
Cereals and bakery products	0.91	0.94
Milk and milk products	0.70	0.80
Herbs and spices	0.75	0.85
Fruits and vegetables	0.81	0.88
Feed additives	0.49	0.77
Cocoa and cocoa preparations, coffee and tea	0.76	0.83
Prepared dishes and snacks	0.87	0.94
Dietetic foods, food supplements, fortified foods	0.91	0.94
Nuts, nut products and seeds	0.76	0.86
Meat and meat products (other than poultry)	0.81	0.93
Non-alcoholic beverages	0.03	0.91
Food additives and flavourings	0.82	0.91
Poultry meat and poultry meat products	0.57	0.86
Food contact materials	0.94	0.98
Alcoholic beverages	0.82	0.86
Confectionery	0.45	0.99
Honey and royal jelly	0.72	0.85
Soups, broths, sauces and condiments	0.95	0.97
Other food product / mixed	0.91	0.95
Eggs and egg products	0.95	0.97
Feed materials	0.80	0.91
Pet feed	0.68	0.85
Ices and desserts	0.96	0.97
Bivalve molluscs and products therefor	0.60	0.82
Fats and oils	**1.00**	**1.00**
Crustaceans and products thereof	0.92	0.96
Cephalopods and products thereof	0.53	0.75
Gastropods	0.70	0.86
Sugars and syrups	0.48	0.89

Table 5.6: F1-score and AUPRC on test set.

5.3.2 Classification on all the leafs of products

The second task was to classify the incidents on all the different product types. As we mentioned before, there are 14.266 different product types. For that purpose, we applied two experiments. During the first experiment, we trained three models for the 1.078 most frequent products. These models are: a RNN-BiLSTM (`BiLSTM C1078`), a Random Forest Classifier (`RF C1078`) and a SGD Classifier by Agroknow
(`SGD-AGR C1078`). In the second experiment, we trained the previous models (`BiLSTM C212`, `RF C212`, `SGD-AGR C212`) on the 212 most frequent classes. As we said before, it is very inefficient to compare the results between 1.078 and 212 classes, so we grouped these classes based on the number of incidents that there are in each class. Table 5.7 presents these groups.

Groups	**Support**
Group 1	1
Group 2	2
Group 3	3
Group 4	4
Group 5	5
Group 6	6-10
Group 7	11-20
Group 8	21-50
Group 9	50+

Table 5.7: Grouped product types based on the incidents.

Now, figures 5.8 and 5.9 depicts the average f1-score and the average AUPRC of the previous groups.

It is clear, that all the models achieved really high scores for Group 8 and Group 9. These groups consist of the product types with the more incidents per product type. Furthermore, the RNN-BiLSTM outperformed the other two baseline models, except `RF C212`, which gave us better results for Group 2. Another interesting point is that all the models had better performance at the second experiment, where we kept the 212 most frequent classes.

Grouped Classes	RNN-BiLSTM C1078	RF C1078	SVM-AGR C1078	RNN-BiLSTM C212	RF C212	SVM-AGR C212
1	0.56	0.29	0.33	0.83	0.00	0.33
2	0.63	0.30	0.43	0.73	0.88	0.55
3	0.67	0.37	0.49	0.85	0.70	0.52
4	0.72	0.46	0.54	0.82	0.72	0.65
5	0.71	0.51	0.59	0.83	0.66	0.69
6-10	0.72	0.59	0.60	0.84	0.70	0.64
11-20	0.80	0.69	0.67	0.86	0.77	0.71
21-50	0.87	0.74	0.75	0.89	0.87	0.83
50+	**0.98**	**0.82**	**0.83**	**0.98**	**0.89**	**0.87**

Table 5.8: Mean f1-score for grouped product types.

Grouped Classes	RNN-BiLSTM C1078	RF C1078	SVM-AGR C1078	RNN-BiLSTM C212	RF C212	SVM-AGR C212
1	0.53	0.28	0.31	0.75	0.00	0.25
2	0.56	0.25	0.36	0.65	**0.81**	0.51
3	0.57	0.30	0.39	0.77	0.61	0.44
4	0.63	0.38	0.46	0.73	0.57	0.56
5	0.62	0.43	0.47	0.74	0.58	0.60
6-10	0.60	0.46	0.47	0.76	0.61	0.54
11-20	0.70	0.56	0.55	0.78	0.67	0.61
21-50	0.77	0.59	0.60	0.84	0.77	0.71
50+	**0.97**	**0.71**	**0.71**	**0.96**	0.81	**0.78**

Table 5.9: Mean AUPRC for grouped product types.

Chapter 6

Conclusion and Future work

6.1 Conclusion

This thesis addressed to the classification task of the food recalls on hazard and product types. As we explained, our data originate from FOODAKAI. At first, we tried to classify each food recall on the top level of the hierarchy, on the kinds of product and hazard types (30 main kinds of products and 9 main kinds of hazards). For that task, we applied two machine learning models, a `Logistic Regression` and a `Random Forest Classifier`. We tried three different inputs for these models. The first one was the `TFIDF` vector, the second was the `BoW` vector representation and the third was the `FastText` word embedding. We find out that `Logistic Regression` with `BoW` vectors as input outperformed the other two models and features.

The second task was the classification on all the hierarchy leaves of hazard and product types. There are 1.714 hazard types and 14.622 product types in total. For the hazard type classification, we developed a `RNN-LSTM` model and for the classification on the product types a `RNN-BiLSTM`. For both classification tasks, we developed two baseline models, a `Random Forest Classifier` and a `SVM Classifier` implemented by Agroknow. As there are many categories, we decided to apply some experiments, where we kept the most frequent classes from products and hazards. In order to compare the results of the previous models, we decided to group the categories based on the number of incidents that they included. In terms of the results, Deep Neural Networks outperformed all the baselines in both classification tasks. Also, all the models performed better when we reduced the number of the classes.

6.2 Future work

First, we will try to change the input of our models. Since `FastText` embeddings were not a perfect match for our task, we could create new embeddings based on our vocabulary. Furthermore, we could use `SciBERT`[1], a pre-train model for word embeddings, which is trained on the scientific text and it could be more related to our corpus.

As we saw before, there are many hazard and product types that have very low support. Our classifiers had not very satisfying results on these categories. For that reason, we could create an artificial dataset, in which there will be more incidents with these hazard and product types. These data could be retrieved from tweets or food reviews. After that, we will add the artificial dataset to our main dataset and we will train again our models. Now, it will be more possible for the classifiers to predict categories with low support.

[1] https://github.com/allenai/scibert

Bibliography

[1] Tome Eftinov, Peter Korošec and Barbara Koroušic (2017)
"StandFood: Standardization of Foods Using a Semi-Automatic System for Classifying and Describing Foods".
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5490521/

[2] Sophie Birot, Charlotte B. Madsen, Astrid G. Kruizing, Tue Christensen, Amélie Crépet, Per B.Brockhof (2018)
"Procedure for grouping food consumption data for use in food allergen risk assessment".
https://www.sciencedirect.com/science/article/pii/S0889157517300182

[3] Elizabeth L. Chin, Gabriel Simmons, Yasmine Y. Bouzid, Annie Kan, Dustin J. Burnett, Ilias Tagkopoulos and Danielle G. Lemay (2019)
"Nutrient Estimation from 24-Hour Food Recalls Using Machine Learning and Database Mapping: A Case Study with Lactose".
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6950225/B22-nutrients-11-03045

[4] Simon Mezgec, Tome Eftimov, Tamara Bucher and Barbara Koroušić Seljak (2018)
"Mixed deep learning and natural language processing method for fake-food image recognition and standardization to help automated dietary assessment".
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6536832/

[5] Maria Kapsokefalou, Mark Roe, Aida Turrini, Helena S. Costa, Emilio Martinez-Victoria, Luisa Marletta, Rachel Berry, and Paul Finglas (2019)
"Food Composition at Present: New Challenges".
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6723776/

[6] Tome Eftimov , Gorjan Popovski , Eva Valenčič , Barbara Koroušić Seljak (2020)
"FoodEx2vec: New foods' representation for advanced food data analysis".
https://pubmed.ncbi.nlm.nih.gov/32088249/

[7] Ya Lu, Thomai Stathopoulou, Maria F Vasiloglou, Stergios Christodoulidis, Beat Blum, Thomas Walser, Vinzenz Meier, Zeno Stanga, Stavroula G Mougiakakou (2019)
"An Artificial Intelligence-Based System for Nutrient Intake Assessment of Hospitalised Patient".
https://ieeexplore.ieee.org/document/8856889/authorsauthor

[8] J D Ireland , A Mølle (2010)
"LanguaL food description: a learning process".
https://www.nature.com/articles/ejcn2010209

[9] Gorjan Popovski, Barbara Koroušić Seljak , Tome Eftimov (2019)
"FoodBase corpus: a new resource of annotated food entities".

[10] Jared Dunnmon, Swetava Ganguli, Darren Hau, Brooke Husi (2019)
"Predicting State-Level Agricultural Sentiment with tweets from Farming Communities".
https://arxiv.org/pdf/1902.07087.pdf

[11] Manokamma Singh, Dr. Katarina Domijan (2019)
"Comparison of Machine Learning Models in Food Authentication Studies".
https://arxiv.org/pdf/1905.07302.pdf

[12] Gorjan Popovski , Stefan Kochev , Barbara Korousi c Seljak and Tome Eftimov (2019)
"A Rule-based Named-entity Recognition Method for Food Information Extractio".

[13] Alberto Nogales , Rodrigo Díaz Morón and Álvaro J. García-Tejedor (2020)
"Food safety risk prediction with Deep Learning models using categorical embeddings on European Union Portal data".
https://arxiv.org/ftp/arxiv/papers/2009/2009.06704.pdf

[14] Birmpa A., Vantarakis A., Anninou A., Bellou M., Kokkinos P. and Groumpos P. (2015)
"A user-friendly theoretical mathematical model for the prediction of food safety in a food production chain".
https://www.longdom.org/open-access/a-userfriendly-theoretical-mathematical-model-2157-7110.1000421.pd

[15] Yamine Bouzembrak, Hans J.P.Marvin (2016)
"Prediction of food fraud type using data from Rapid Alert System for Food and Feed (RASFF) and Bayesian network modelling ".
https://www.sciencedirect.com/science/article/pii/S095671351530205X

[16] Goldberg, Yoav et al. (2017)
"Neural Network Methods in Natural Language Processing ".

[17] Huang, Zhiheng (2015)
"Bidirectional LSTM-CRF Models for Sequence Tagging".
http://arxiv.org/abs/1508.01991

[18] Hochreiter, Sepp and Schmidhuber, Jürgen (1997)
"Long Short-Term Memory. Neural Comput".

[19] Gal, Yarin and Zoubin Ghahramani (2016)
"A Theoretically Grounded Application of Dropout in Recurrent Neural Networks".

[20] Nickel, Kiela (2017)
"Poincaré Embeddings for Learning Hierarchical Representations".
https://arxiv.org/pdf/1705.08039.pdf

[21] Piao, Bianchi, Dayrell, D'Egidio and Rayson (2015)
"Development of the Multilingual Semantic Annotation System".
https://www.aclweb.org/anthology/N15-1137.pdf

[22] Manning, Surdeanu, Bauer, Finkel, Bethard and McClosky (2014)
"The Stanford CoreNLP Natural Language Processing Toolkit".
https://nlp.stanford.edu/pubs/StanfordCoreNlp2014.pdf

[23] Srivastava, Hinton, Krizhevsky, Sutskever and Salakhutdinov (2014)
""Dropout: A Simple Way to Prevent Neural Networks from Overfitting".

[24] Stanislau, Aliaksei and Erhardt (2016)
"Recurrent Dropout without Memory Loss".
https://arxiv.org/pdf/1603.05118v2.pdf

[25] T. Mitchell, McGraw-Hill (1997)
"Machine Learning".

[26] A. Liaw, M. Wiener (2002)
"Classification and regression by random forest".

[27] Thorster Joackims (2002)
"Learning to Classify Text Using Support Vector Machines".

[28] Yujun Zhou, Bo Xu, Jiaming Xu, Lei Yang, Changliang Li, Bo Xu (2016)
"LSTM Recurrent Neural Networks for Short Text and Sentiment Classification".

[29] Ali Alessa, Miad Faezipour, Zakhriya Alhassan (2018)
"Text Classification of Flu-Related Tweets Using FastText with Sentiment and Keyword Features".
https://ieeexplore.ieee.org/abstract/document/8419391

[30] Ali Alessa, Miad Faezipour, Zakhriya Alhassan (2018)
"Text Classification of Flu-Related Tweets Using FastText with Sentiment and Keyword Features".
https://ieeexplore.ieee.org/abstract/document/8419391

[31] Pradeep Vashisth, Kevin Meehan (2020)
"Gender classification using twitter data".
https://ieeexplore.ieee.org/abstract/document/9180161

Appendix

Table 6.1 describes the hyper-parameters used for the RNN-LSTM and the RNN-BiLSTM models.

Parameter name	**Description**
`batch_size`	The batch size
`patience`	Patience for early stopping
`lr`	The learning rate of Adam optimizer
`SpatialDropout1D`	The 1D Dropout
`dropout`	The Dropout used for the models
`reccurent_dropout`	The Dropout for the RNN models
`units`	Dimensionality of the output space
`emb_dim`	The dimension of embeddings

Table 6.1: Parameters used for the RNN models.

Table 6.2 shows the hyper-parameter tuning achieved for the previous hyper-parameters.

Parameter name	**Description**
`batch_size`	[64, 128, 256]
`patience`	[3, 5, 8]
`lr`	[0.0001, 0.001]
`SpatialDropout1D`	[0.2, 0.4, 0.6]
`dropout`	[0.2, 0.4, 0.6]
`reccurent_dropout`	[0.2, 0.4, 0.6]
`units`	[100, 150, 200]
`emb_dim`	[250, 300, 350]

Table 6.2: Hyper-parameter tuning.