FSA - Allergies and Social Media

The four Seasons

Contents

Preface

The team is composed of Alejandro, Andrew, Ozge and Toufik.

Chapter 1

Introduction

1.1 Context

Using social media data, FSA wants to be able to track endorcement of allergen based legislations in the UK. There are two streams of analysis involved in this project :

1.1.1 Stream 1

Using social media data, we would like to determine whether some Local Authorities face more allergy-relate issues than others by tracking:

- Allergy enquiries: Do consumers experience negative or positive reactions from staff when enquiring about allergen information in restaurants/ food outlets?
- Food labelling: Do consumers face issues with incorrect or incomplete food labelling?
- Reporting reactions: Do consumers report allergic reactions to food or near-misses?

1.1.2 Stream 2

Based on a list of 14 common allergens provided in the EU legislation, we will focus on these 2 areas:

- How much are each of the 14 allergens being talked about, and are they usually mentioned in isolation or in combination with others? What insight can we gain by analysis of posts relating to these 14 allergens?
- Are other allergens outside of this list of 14 being talked about? We will explore this by: (a) using a list of other likely allergenic foods provided by subject matter experts, and (b) for posts that do not contain any of the main 14, are there any common food themes. Are any of these mentioned more than any of the 14 allergens?

Chapter 2

Data Cleaning and Preprocessing

2.1 Cleaning

First step of our code is too import the data provided by FSA in Excel Format and to load it in a DataFrame. This will allow us to perform the data cleaning, which is composed of the following steps:

- Remove of undesiered columns
- Ordering of data per dates
- Lowercasing of content
- Removal of undesired characters from including Emoticons, Hashtags, URLs, HTML tags and symbols and punctuation
- Removal of duplicates
- Removal of spaces around conten (Trimming)

For further future analysis, some information like Username and Hashtags are extracted in inserted in a new column.

Lots of content is composed of abbraviations. In order to be the more relevant possible in the future analysis, we convert those to there expander meaning for e.g. “asap” becomes “as soon as possible”

2.2 Normalization

The normalization refers to the transformation of words into a more uniform form.

We perform stop words removals where words like “the” “an” and other articles are removed from the content. This allows compuation to be more efficiant, reducing content size.

We apply the stemming process, an algorithm that converts inflected forms of words into their base forms (stems). This allows us to discard variations of words (like singular, plural).

The words are treated like normalized elements that we call tokens.

We need to convert this set of tokens to a corpus in order to perform matricial analysis.

2.3 Data Term Matrix

After all our data is cleaned and preprocessed, our corpus is converted into a Data Term Matrix. A DTM is a matrix in which rows are documents, columns are terms, and cells indicate how often each term occurred in each document. In our case each line represents a tweet, forum entry or news.

This is the base element that we will use for Stream 1 and Stream 2 analyis.

Chapter 3

Data Analysis

3.1 Supporting local authorities

3.1.1 Approach

The idea is to use provided data in order to find out information about: 1. Allergy enquiries 2. Food labelling 3. Reporting reactions

Our approach was to use each of these elements and to find words related to the subject.

Let's take Food labelling as an example. The idea is to find cases where consumers face issues due to incorrect food labelling.

We divided it into the following list:

```
c("consumer", "issue", "labelling", "allergy")
```

We used a Google library called Word2Vec an efficient implementation of the continuous bag-of-words and skip-gram architectures for computing vector representations of words.

For each of the words consumer, issue, labelling and allergen, we used word2vec to generate dictionaries of words with a strong semantic proximity.

If we talk the word allergy, applying word2vec functions to it provided us with a list of words related to the allergen subject. With some data cleaning and manual rework, we obtain a dictionary with and values.

Example for allergy : `allergy allergy,allergen,allergenic,allergens,allergic,allergies,allergy,`

This processed is applied to all the words that we will use to identify issues with food labelling.

We will generate a DFM composed of the list of all the entries (tweets, news, forum) by line, the column will be the key of our dictionary (in that case "allergy") Because some of these may be repeted several times in a same document tweet or forum posts, the DFM is **normalized** containing 1 for any occurence of the word >= 1. The matrix is composed of 0 and 1, corresponding to the occurrences of the context words.

3.1.2 Labelling

Using the previous generated matrix, we will apply the following logic to filter and label the tweets as per their corresponding context.

Staying in the example of food labelling, we apply the following logical expression :

[(consurmer AND issue AND labelling) OR (incorrect AND allergy AND labelling) OR (consumer AND allergy AND labelling)]

When one of the above conditions is verified, we flag the corresponding document (tweet, or post forum) with the food_labelling value to 1

The same process is applyied for the allergies inqueries and reactions reports.

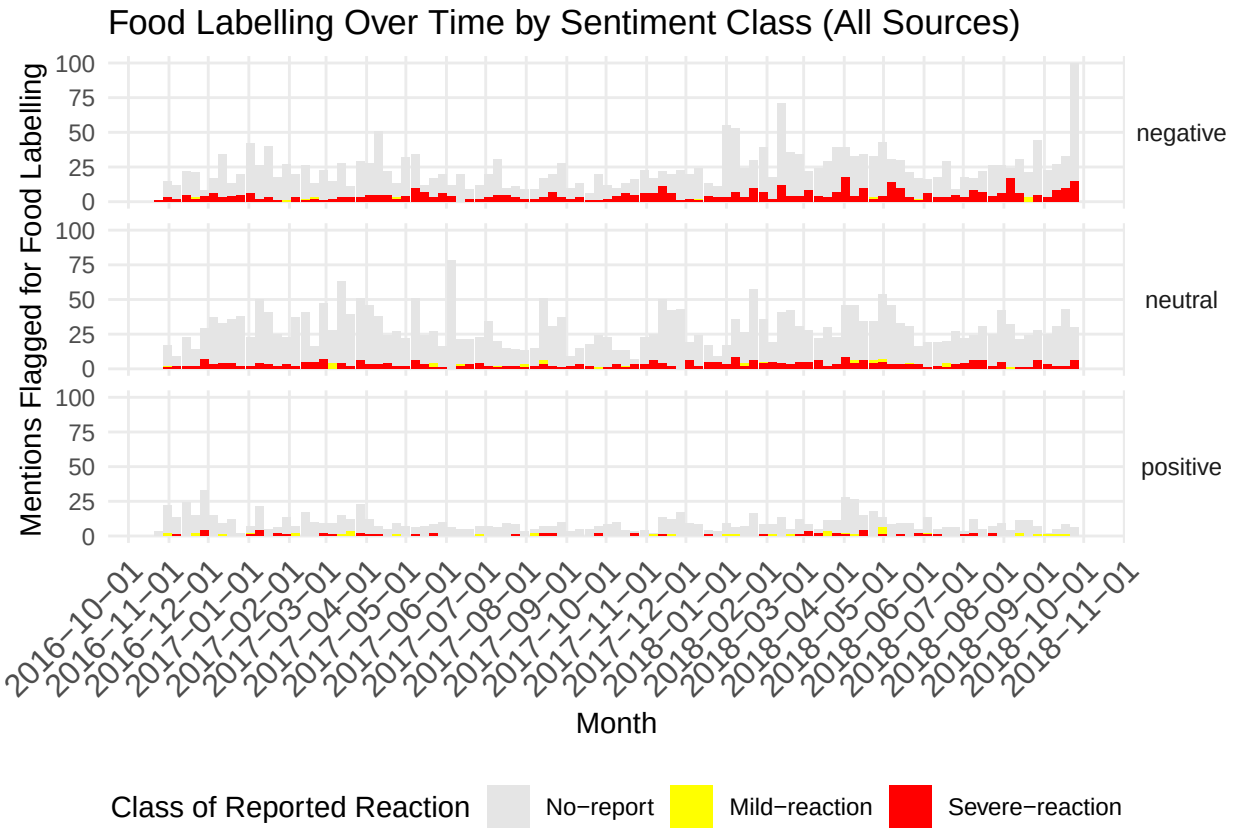
3.2 Identifying common allergens

A similar approach is used to identify common allergens. The difference is that we already know the predefined list of keywords we are looking for.

In that case, we built dictionaries for the 14 allergens provided, and for the other list of allergens using the same principal as in first stream.

```
"celery" c("celery","celery","celeri")
```

```
"cereals containing gluten" c("cereals_contain_gluten","gluten","wheat","rye","barley","barli","oat")
```

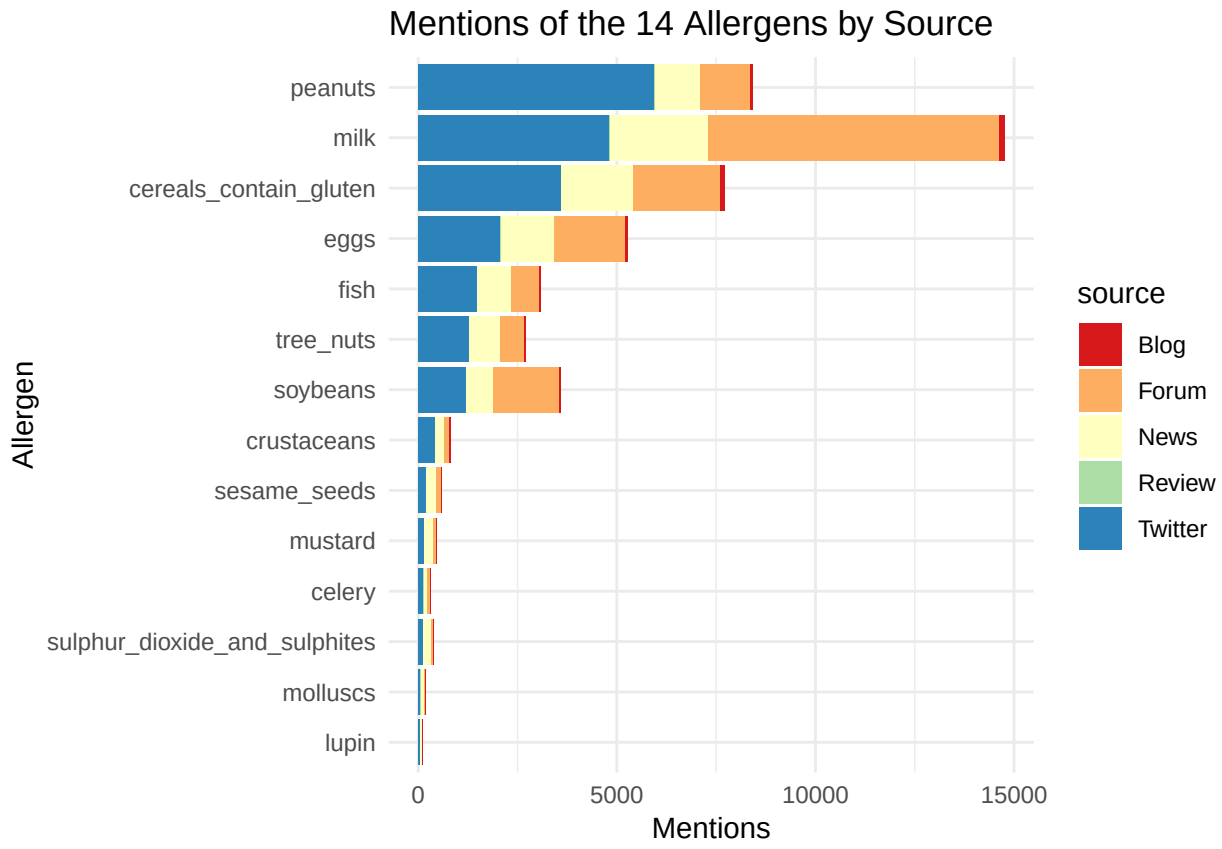


Chapter 4

Results and Plots

4.1 14 allergens

In the representation below, we display the number of mentions of predefined list of 14 allergens, whether coming from Blog, News, Review, Forum or Twitter



4.2 Other Allergens

```
other.bysource <- ggplot(subset(allergen.bysource.df,
                                Allergen %in% other.allergen.names),
                         aes(x = fct_reorder2(Allergen, source, count, .desc = FALSE), y= count, f
  geom_bar(stat = "identity") +
  theme_minimal() +
  scale_fill_brewer(palette="Spectral") +
  xlab("Allergen") +
  ylab("Mentions") +
  ggtitle("Mentions of Other Allergens by Source") +
  coord_flip()
other.bysource
```