Packaged foods with pulse ingredients in Europe: A dataset of text-mined product formulations

Tristan Salord, Marie-Benoît Magrini, Guillaume Cabanac

Contents lists available at ScienceDirect

Data in Brief

journal homepage: www.elsevier.com/locate/dib

Data Article

Packaged foods with pulse ingredients in Europe: A dataset of text-mined product formulations

Tristan Salord[a,*], Marie-Benoît Magrini[a], Guillaume Cabanac[b]

[a] AGIR, INRAE, University Toulouse, Castanet-Tolosan, France
[b] CNRS, IRIT, University Toulouse, Toulouse, France

ARTICLE INFO

Article history:
Received 12 January 2022
Revised 1 April 2022
Accepted 7 April 2022
Available online 14 April 2022

Keywords:
Food ingredient parsing
Text mining
Pulses - legumes
Market biodiversity
Processed food
Food science and technology

ABSTRACT

There is a lack of methods and tools to reveal robust information on the ingredients used in packaged foods. To tackle this challenge, we developed an original method to parse ingredient lists of packaged foods. We built a dataset of food product innovations with their parsed ingredient lists. We explain the parser algorithm used to provide this dataset; and a benchmark method assessing the performance of the parsing techniques applied on those food ingredient lists. The primary data we used to test and apply this method were retrieved from MINTEL-GNPD. These data cover new food products containing pulse ingredients launched on European markets over the last decade. This work brings original results informing on the diversity of pulse species used in food products, and on the technological features of these ingredients from whole-grain to ultra-processed uses (such as protein isolates). The parsing techniques we developed can be reused to analyse other ingredient lists. This method also makes it possible to assess marketed crop biodiversity in relation to how species diversity is represented in food products, as well as the level of complexity of food formulations. Hence, this work contributes towards providing more complete information on the characteristics of foodstuffs supplied on markets for both private and public stakeholders.

[*] Corresponding author.
E-mail address: tristan.salord@inrae.fr (T. Salord).
Social media: @TristanSalord (T. Salord)

https://doi.org/10.1016/j.dib.2022.108173

Specifications Table

Subject	Food Science
Specific subject area	Text-mining on food ingredient lists
Type of data	Tabular data in csv files, Parsing algorithm written in Python, Benchmark method
How data were acquired	The primary data were retrieved from MINTEL-GNPD Database as a csv file thanks to their database query engine. This access was allowed through a subscription to MINTEL-GNPD. The secondary dataset was built by applying our parsing algorithm to the primary data.
Data format	Filtered, csv file, Python module
Description of data collection	Primary data were retrieved thanks to the MINTEL Boolean query search engine. Secondary data were obtained by running our parsing algorithm and are presented in a csv file.
Data source location	INRAE Auzeville-Tolosane France Primary data sources: Excerpt of the MINTEL – GNPD (Global New Products Database), gathering food product innovations containing pulses during the last decade (2009–2019) on a European scale.
Data accessibility	Resulting Dataset: Repository name: https://data.inrae.fr/ Data identification number: doi:10.15454/KRPEI2 Direct URL to data: https://doi.org/10.15454/KRPEI2 Algorithm: Repository name: GitHub Repository URL: https://github.com/Pythrix/FOODCOP Digital Object Identifier:https://doi.org/10.5281/zenodo.6397383

Value of the Data

- The study of food ingredient lists reveals how the foodstuff supply is changing and developing. Here, we focus on the uses of pulse ingredients, which is currently debated at the global scale (see the 2016 FAO International Year of Pulses).
- Food platforms or any scientific community can use those methods to provide clear information on ingredients used in food products.
- This type of data informs public authorities on the ways foodstuffs are developing, and can help defining sound public food policies, in particular here for promoting pulses.
- This parsing technique facilitates the identification of the species used in foodstuffs. Parsed ingredient lists provide data for studying species biodiversity on food markets; from this we can study lock-in on food markets resulting from the unbalanced development between major species and minor ones such as pulses.

1. Data Description

There is a general consensus that increasing pulse consumption will contribute to more sustainable and healthy diets [1–4]. However, pulses are facing a strong lock-in situation as their consumption has been very low for years now in western countries, and globally neglected compared to major crops [5,6]. Therefore, numerous innovations are expected to develop their consumption, but debates exist on the different food pathways that could foster this development on food markets [1] and how various pulses are used to promote stronger food biodiversity [2].

Analysing the ‘promising narratives’ of pulses in food literature highlights two main pathways for development. On one hand, pulses contribute to unprocessed, traditional and whole foods, whilst on the other hand they are the fungible base materials for high-tech food processors [1]. But there is a vaster array of food product innovations between these two opposing main food paradigms. In addition, several market studies suggest that current development on pulses is mainly focused on the pea, thus neglecting other pulses [7]. But no dataset exists for analysing product innovations containing pulses to further the debate and assess how pulses are developing in food products.

To tackle this challenge, we retrieved data from the Mintel GNPD[1] (“Global New Products Database”), a commercial food database tracking food products launches, by those containing pulse ingredients. Our dataset [8] [dataset] provides an organised data table of detailed information extracted by parsing the primary data printed on the packaging of food products launched on the market over the last decade. It reflects the ingredients that compose those food products. Primary data are described in Figs. 2 and 5, and the tools used to harvest them in Figs. 3 and 4 (presenting the search engine and query used).

Working on the food products ingredients list was challenging. For the moment, no common presentation system exists for food ingredients [9], which means ingredient listing is idiosyncratic and depends very much on different national regulations and each firms’ own habits and methods. Numerous difficulties arise from this heterogeneity when it comes to automating textual processing of food ingredient lists. This may explain why very few tools exist in scientific literature for such tasks.

The most well-known food databases - the USDA’s[2] and OpenFoodFact[3] - have developed their own specific web service or framework [10] for retrieving data gathered by those platforms. This is powered by specific tools to perform the necessary pre-processing tasks, such as eliminating redundant information, correcting synonyms, normalising spelling and parsing (i.e., breaking-down) the list of ingredients. But these tools are dependent on the platforms for which they have been designed, i.e. for databases not specifically dedicated to food innovation; and have not been benchmarked.

Some scripts were released on well-known web-based version-control and collaboration platforms such as GitHub[4], but some fell short because (i) they had not been benchmarked; (ii) they did not catch the attention of the scientific community (no reuse in research projects). Finally, INRAE (the French National Research Institute for Agriculture, Food, and Environment) has also developed a web-platform dedicated to the study of food supply, ODALIM[5], from which an API offers a similar service. However, it is only available to researchers belonging to the institute and has not been benchmarked.

In this context, our work has aimed at developing an algorithm [11] that is able to parse ingredient lists of food products into organised data structures called dictionaries. Table 1 presents the types of information related to an ingredient (extracted from ingredient lists) that constitute the different main entries in dictionaries.

In short, the standard structure of the parsed ingredient list (i.e. data dictionary) is as follows (Fig. 1):

[1] https://www.mintel.com/about-mintel. Data were retrieved from the Mintel GNPD database thanks to a subscription license. The filtered data presented in the paper are subject to a publishing agreement with Mintel.

[2] https://fdc.nal.usda.gov/

[3] https://openfoodfacts.org/

[4] See for example https://github.com/q-m/food-ingredient-parser-ruby, or https://github.com/irockel/ingredients_parser

[5] https://odalim.inrae.fr

Table 1
Types of ingredient information and code names.

Types of ingredient information – code names	Description
Ingredient Name – 'rawing'	Name of the ingredient as mentioned in the ingredient list.
Ingredient level of appearance – 'level'	Depth at which the ingredient appears in the ingredient list. For example, if the ingredient appears inside parentheses, (ingredient), it has a depth of 1. If the ingredient appears inside parentheses which are themselves inside parentheses, ((ingredient)), ingredient has a depth of 2, and so on.
Proportion of the ingredient – 'prop'	Proportion of the ingredient as mentioned in the ingredient list.
Comment related to the ingredient – 'comment'	Extra-information related to the ingredient. Only concerns information marked by a specific character sign.
Additive Category – 'additives'	General category or function of the additive mentioned in the ingredient list.

```
{'ProdId':
        {'ProdId Nb.1':
             {'Ingredient Name': 'XXX', ' Ingredient level of appearance ': 'XXX',
          'Proportion of the ingredient': 'XXX', 'Comment related to the
          ingredient': 'XXX', 'Additive Category': 'XXX'},
        'ProdId. Nb.2':
             {…},
        …}
}
```

Fig. 1. The data structure of the parsed Ingredients list.

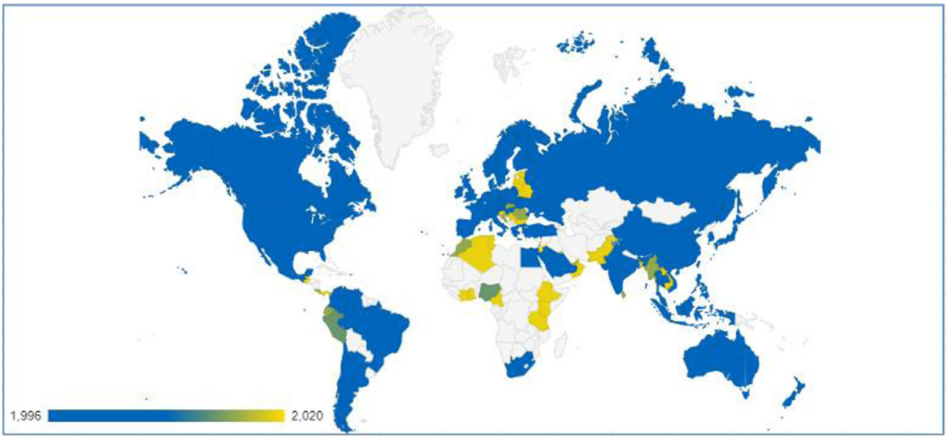


Fig. 2. Coverage map of mystery shoppers per country by MINTEL from 1996 to 2020 (source: personal communication from Mintel).

The general workflow of the algorithm is presented in Fig. 5. Results are illustrated in Excerpts throughout the article. Excerpts 1, 2, and 5 illustrate the different sorts of ingredient list notation styles we had to deal with, while Excerpt 3 underlines how ingredients lists are normalised before being transformed into enriched ingredient dictionaries (Excerpts 4 and 6).

We argue that the method used and its algorithmic implementation can be reused by the scientific community, and contributes in 4 major ways:

- We have provided a benchmarking method, for assessing the performance of ingredient list parsing. It includes a set of metrics, adapted to the type of output expected. These metrics are compared in Fig. 8. Excerpts 7 and 8 illustrate how ingredient dictionaries are transformed to compute those metrics.
- We have released an algorithm to the scientific community that parses ingredient lists into structured dictionaries. It is licensed under the Creative Commons Attribution-NonCommercial Licence 4 and is fully accessible [11].
- We have presented a benchmark of the existing algorithms making it possible to decompose food ingredient lists. These results are presented in the table named “Benchmark Table Results”. This table gathers the score obtained by our algorithm, and a similar tool available in the ODALIM web platform. Fig. 7 presents a general overview of the resulting differences between these two algorithms.
- We have reported the results obtained by our algorithm. Results are structured as a database table to facilitate further data processing and analysis.

2. Experimental Design, Materials and Methods

2.1. Presentation of materials

Materials used to develop this algorithm and the benchmark method were sourced from the Mintel GNPD for the Global New Products Database. Mintel is an international company that has been curating a large food database since 1972. It indexes all food products launched (i.e., food production innovations) on every market covered by the company. The database now surveys product innovations in 86 countries and the European market has been covered entirely since 1996 (Fig. 1).

The Mintel database records visual observations made by hired “Mystery Shoppers”[6]. Information on product packaging is recorded through an extensive network of expert shoppers in each country covered. 80 different data fields are considered. The Mintel-GNPD currently listed 3,666,585 products as of July 2021.

We chose to use this database because (i), it is an innovation-oriented database - only new launches are registered; and (ii), for the high level of details available for the products included [12]. Indeed, other well-known accessible food databases such as the USDA database or OpenFoodFacts do not make this distinction between product innovations and well-established food products (i.e., most consumed products). Therefore, with other databases it is not possible to assess how market supply is trying to change (whether or not those food innovations are a success). They also register less entries than the Mintel-GNPD. The OpenFoodFacts database listed 1,887,966 products and the USDA branded food database listed 1,142,610 as of July 2021.

The source corpus was obtained thanks to the Boolean query search engine proposed by the GNPD platform (Fig. 3).

As seen in Fig. 3, we retrieved all the products containing at least one pulse ingredient launched on European markets during the last decade – from 1st January 2010 to 31st December 2019 - based on the Mintel ingredient ontology. The search was limited to food products and a set of selected beverage categories (‘Carbonated Soft Drinks’, ‘Hot Beverages’, ‘Juice Drinks’…) in order to exclude pet food products, beauty products, and alcohols. Hence, we gathered an initial corpus of 58,649 products.

Data were downloaded from the Mintel GNPD as a csv file where each row represents a product, and each product is characterised by various information[7]. Using a filtering step on the geographical area where the product was distributed, we selected a final corpus (the filtered

[6] https://shopper.mintel.com

[7] For more information on the description of food products in MINTEL-GNPD see the Mintel Glossary: https://www.gnpd.com/gnpd/about/GNPD_Glossary_2016.1.pdf

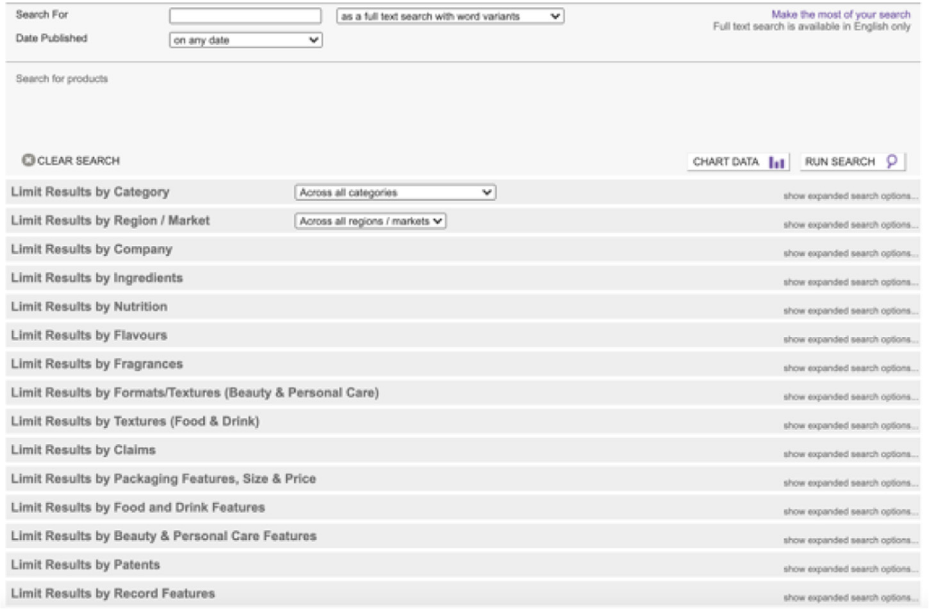
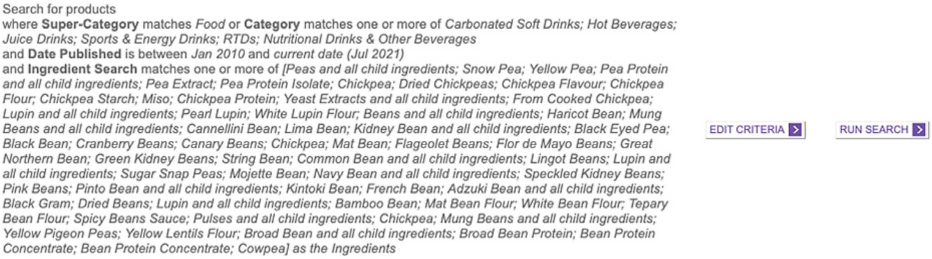

Search For
as a full text search with word variants
Make the most of your search
Full text search is available in English only
Date Published
on any date
Search for products
CLEAR SEARCH
CHART DATA
RUN SEARCH
Limit Results by Category
Across all categories
Limit Results by Region / Market
Across all regions / markets
Limit Results by Company
Limit Results by Ingredients
Limit Results by Nutrition
Limit Results by Flavours
Limit Results by Fragrances
Limit Results by Formats/Textures (Beauty & Personal Care)
Limit Results by Textures (Food & Drink)
Limit Results by Claims
Limit Results by Packaging Features, Size & Price
Limit Results by Food and Drink Features
Limit Results by Beauty & Personal Care Features
Limit Results by Patents
Limit Results by Record Features
show expanded search options...

Fig. 3. Screenshot of the Mintel GNPD Query Tool.



Search for products
where **Super-Category** matches *Food* or **Category** matches one or more of *Carbonated Soft Drinks; Hot Beverages; Juice Drinks; Sports & Energy Drinks; RTDs; Nutritional Drinks & Other Beverages*
and **Date Published** is between *Jan 2010* and *current date (Jul 2021)*
and **Ingredient Search** matches one or more of *[Peas and all child ingredients; Snow Pea; Yellow Pea; Pea Protein and all child ingredients; Pea Extract; Pea Protein Isolate; Chickpea; Dried Chickpeas; Chickpea Flavour; Chickpea Flour; Chickpea Starch; Miso; Chickpea Protein; Yeast Extracts and all child ingredients; From Cooked Chickpea; Lupin and all child ingredients; Pearl Lupin; White Lupin Flour; Beans and all child ingredients; Haricot Bean; Mung Beans and all child ingredients; Cannellini Bean; Lima Bean; Kidney Bean and all child ingredients; Black Eyed Pea; Black Bean; Cranberry Beans; Canary Beans; Chickpea; Mat Bean; Flageolet Beans; Flor de Mayo Beans; Great Northern Bean; Green Kidney Beans; String Bean; Common Bean and all child ingredients; Lingot Beans; Lupin and all child ingredients; Sugar Snap Peas; Mojette Bean; Navy Bean and all child ingredients; Speckled Kidney Beans; Pink Beans; Pinto Bean and all child ingredients; Kintoki Bean; French Bean; Adzuki Bean and all child ingredients; Black Gram; Dried Beans; Lupin and all child ingredients; Bamboo Bean; Mat Bean Flour; White Bean Flour; Tepary Bean Flour; Spicy Beans Sauce; Pulses and all child ingredients; Chickpea; Mung Beans and all child ingredients; Yellow Pigeon Peas; Yellow Lentils Flour; Broad Bean and all child ingredients; Broad Bean Protein; Bean Protein Concentrate; Bean Protein Concentrate; Cowpea] as the Ingredients*

Fig. 4. The original Query of the source corpus.

corpus In Fig. 5) of 21,519 food innovations launched on European markets during the targeted time-frame. This corpus was then used to (i) test and modify the parsing algorithm; (ii) produce a gold standard based on manual annotations for benchmarking purposes; (iii) test existing parsing methods and improve our own parser. Fig. 5 shows an overview of this process.

2.2. Ingredient parsing algorithm

The parsing algorithm takes as input the raw food ingredient lists retrieved from the primary source (Excerpt 1) in order to transform them into *dictionaries*[8] (see Fig. 1) - a data structure commonly used in computer science. This is a useful data structure for food ingredient lists as

[8] "A dictionary is a general-purpose data structure for storing a group of objects. A dictionary has a set of keys and each key has a single associated value. When presented with a key, the dictionary will return the associated value" (https://en.wikibooks.org/wiki/A-level_Computing/AQA/Paper_1/Fundamentals_of_data_structures/Dictionaries). A value can be a dictionary itself.This general-purpose type of data-structure is present in the most widely-used programming languages.

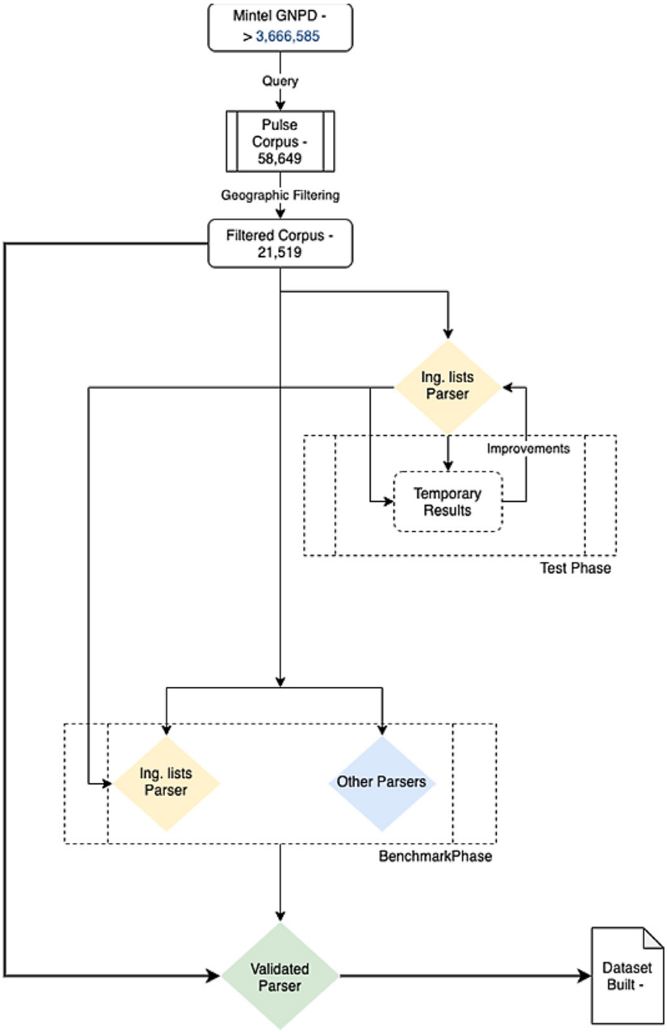


Fig. 5. The process flow describing the dataset built on food ingredients and the benchmark method applied to assess the parsing algorithm used.

it helps organising all the information associated to ingredients (see Table 1 as an illustration) such as proportions, size, labels, origins, etc. of certain ingredients (as not all ingredients are described with the same level of detail). We have explained hereafter the main steps of the parsing algorithm through various Excerpts taken as illustrations.

The data transformation from the parsing algorithm runs through 3 main stages:

- The pre-processing stage cleans ingredient lists from potentially ambiguous punctuation and normalises spelling variations for problematic expressions and additive name substitutions.
- The analytical stage identifies the different grammatical forms of the ingredient lists and normalises them, i.e., it reduces them to a common and more simple data form.
- The building stage embeds a parsing step and a dictionary building step. Each ingredient list is split into its different elements according to the general organisation of the list (that is,

the ingredients' position in the list). Any other information that can be associated to certain ingredients (such as origin, additive category, mention of a label, etc.) is preserved.

This parsing algorithm also features:

- A display function for users to show the result of the analytical step.
- A conversion option to convert the resulting dictionaries into nested lists (list of lists).
- Error-reporting during parsing is reported on a log file for further improvements.

Hereafter is a short description of each stage.

The pre-processing stage is a cleaning phase, necessary to avoid misunderstandings when parsing ingredient lists. One of its main tasks consists in normalising delimiters used to distinguish elements in ingredients lists. For example, in some cases, manufacturers use semicolons to separate each ingredient in ingredient lists. In other cases, brackets ([]), braces ({}) or colons (:) are used instead of parentheses to identify the sub-ingredients of a given ingredient (that could be considered as a complex ingredient).

We chose to standardise all these *varia* into a unique ingredient list form where each ingredient is separated from the others by a comma, and each sub-ingredient of a given ingredient is in parentheses. The pre-processing stage then uses a homemade dictionary to substitute potentially problematic annotations for less ambiguous ones, and additives names for their European code. For example, expressions such as 'l(+)-tartaric acid' or 'disodium 5'-ribonucleotide', that might impede the parsing stage because of the punctuation used ('+' and '-'), are replaced by the corresponding European code, in this case 'E334' and 'E635'. This mapping was based on the analysis of the error log during the algorithm test phase with the help of food science experts. It shall be updated and improved.

The analytical stage consists on a set of functions designed to analyse the way ingredient lists are composed. This is a crucial phase as it serves as input for the final parsing and building stage. During the analytical stage, the program determines if the raw food ingredient lists (i) include additional information, such as comments or additive categories (see Excerpt 1); (ii) are presented in a complex form or not, that is to say if an ingredient list includes ingredients presented with a sub-ingredient list (see Excerpt 2).

For instance, Excerpt 1 shows that some ingredients are marked by a specific character that refers to additional information. In this example, all ingredients identified with '°' come from organic agriculture. We can also observe that additives are categorised; 'soy lecithin' is used as an 'emulsifier'. The second illustration (Excerpt 2) presents a complex ingredient list containing three main compounds ('Noodles', 'Soup', 'Flakes') for each of which the list of sub-ingredients is given between a colon ':' and a carriage return.

The *building stage* consists of two main actions. Firstly, the normalisation of the ingredient lists removes all comments and additive categories, which are kept in memory by the program for building a dictionary. A similar process is applied to proportions. Finally, the complex ingredient lists are reduced to a simplified and standardised form, as illustrated in Excerpt 3: sub-ingredient lists are placed between parentheses and all colons and carriage returns are removed.

The second and final action consists in constructing the ingredient dictionary itself. Each element from the ingredient list is parsed and assigned to a given number corresponding to its rank in the ingredient list. As illustrated in Fig. 1, the concatenation of the product identifier (ProdId) and the rank of the considered ingredient becomes a key that identifies each ingredient, and any characteristic or information related directly to this ingredient. Is the ingredient linked to a comment or a proportion? Is the ingredient an additive? Has this additive been classified in a category? At what level does the ingredient appear? Etc. Excerpt 4 shows an overview of the parsing algorithm result. In this case, dictionary entries are the concatenation of the product identification number (here, '75') and the ingredient rank. The value of each entry is a sub-dictionary describing the ingredient targeted. Keys of that dictionary are constituted by the characteristics identified in Table 1. Only the characteristics available in the raw ingredient lists are mentioned. If an ingredient is not characterised by a proportion or a comment in the basic ingredient list, there will be no "proportion" or "comment" key in the output dictionary.

$$\mathrm{RBO}(S,T,p) = (1-p)\sum_{d=1}^{\infty} p^{d-1} \cdot A_d$$

Fig. 6. RBO Formula from Webber et al. [13].

2.3. Benchmark method

As mentioned in the section “Data Description”, we propose a benchmark method to assess the effectiveness of parsing algorithms having to deal with that sort of data structure. We give, as additional materials[9](see “Benchmark Table Results”), the results of this method applied both for our parsing algorithm and the ODALIM webservice dedicated to decoding ingredients (“Simple Decoding Ingredients”[10] – now called “SDI”) on a random dataset of 100 ingredient lists extracted from the “Filtered Corpus”.

Special attention was paid to this proximity calculation method, as in our particular case we had to take into account that:

- Ingredient lists obtained thanks to the different parsers could be of varying lengths.
- The order of ingredients (i.e., their rank in the ingredient list) and depth (whether the ingredient is a sub-ingredient of a complex ingredient) is extremely important when computing similarity or dissemblance scoring between ingredient lists.

These considerations greatly reduce choice among the available proximity computation methods. The most relevant ones are intended for lists of identical lengths according to Webber et al. [13]. We chose one designed for lists of varying lengths: the Rank-Biased Overlap (RBO) method, developed by the same above-mentioned authors. RBO is an established metric; largely commented upon [14] and assessed, easy to implement and already available in different programming languages[11].

The RBO method observes element ranking in a given list, with weighted elements of higher ranks (i.e., the first to appear in the list) compared to lower ranks. The metric is based on a calculation of the overlap between two indefinite rankings (S and T) at incrementally increasing depths, given that a probabilistic setting representing the user’s patience, the “*p*” parameter in the next figure, i.e. "(...) *the probability that the user, having examined the overlap at one rank, continues on to consider the overlap at the next*" (p20:3, and Fig. 6). A_d refers to the match between the two lists at rank *d.*

But ingredient lists should not only be looked at horizontally, i.e. only taking into account the ingredients’ order of appearance. They also need to be looked at vertically, as certain ingredients can be parts of other ingredients and cannot be considered as ingredients of the same level. As RBO does not handle such situations, our benchmark method decomposes RBO scores for each ingredient list of a given level, as illustrated in the following excerpts (Excerpts 5, 6).

The parsing algorithm translates a basic ingredient list into a dictionary (Excerpt 6).

The above illustrations demonstrate how ingredients that are part of another ingredient (for example, ‘canola oil’ is a ‘vegetable oil’, Excerpt 5) appear in the ingredient dictionary as ingredients from level 1 (Excerpt 6). Note that if the algorithm finds ingredients that are part of another ingredient that is already part of a higher-level ingredient, the level gets increased by 1 unit, and so on.

In this case, the ingredient list can be broken down into ingredient lists established at different levels. Thus, the above illustrated ingredient list can be split into a list of ingredients at level 0 (`‘’’water, milk protein concentrate, soy protein isolate,`

[9] For description of the results see next section “Benchmark Results”

[10] https://odalim.inrae.fr/fr/tools

[11] For Python implementation see for example https://github.com/changyaochen/rbo or https://github.com/dlukes/rbo. We also found an existing implementation in R https://github.com/neverfox/rbo, but we did not test it.

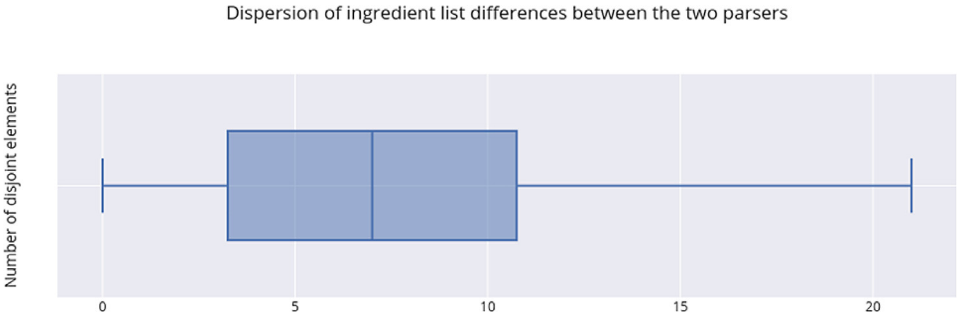


Fig. 7. Boxplot, Dispersion of the differences between ingredient lists parsed by the two parsers ($N = 75$).

`calcium caseinate,...'''`) and an ingredient list at level 1 (`''' canola oil, high oleic sunflower oil, corn oil,...'''`).

The benchmark method computes RBO[12] scores for every ingredient list belonging to the same level, and then sums-up each score following a fractional count method. In this way, the ingredients' order of appearance is included in the computation of the proximity score, as well as how far down the list these ingredients appear.

In addition to this benchmark method, we have also computed two other proximity scores on flattened ingredient lists for comparison purposes (Cosine Similarity and Average Overlap[13]). As such, the table presented in "supplementary material" (see table Parser Benchmark Results) gives the score obtained by our parsing algorithm and the ODALIM webservice mentioned above (SDI), computing Cosine Similarity and Average Overlap - necessary to compute RBO.

2.4. Benchmark results

The most significant differences between the parsers lie in the way they deal with punctuation and complex grammar - in particular those referring to additives and complex ingredient lists.

Tested on 100 ingredient lists, the SDI webservice failed 25 times (25%, 'NA' value in the table Parser Benchmark Results), returning a punctuation issue. These failures mostly occur when the algorithm has to deal with complex ingredient lists containing multiple lines (e.g., Excerpt 2), or when complex ingredient names contain specific characters such as '-', as in "mono- and diglycerides of fatty acids". Considering that our parser algorithm was specifically designed to deal with this kind of heterogeneity in terms of spelling or ingredient list grammar, it is not surprising we observed less failures. Tested on the same 100 ingredient lists, our parsing algorithm failed only 3 times, and each of these failures were due to a gap in the data[14].

Even when SDI succeeds in analysing the ingredient lists, some differences persisted among results. We can observe this in the box plot (Fig. 7) that shows the dispersion of the number of disjointed elements between the ingredient lists parsed by the two compared algorithms ("DisjointsEls", i.e. the number of ingredients that are absent from both parsed ingredient lists). The box plot is centred to the left, showing that when both parsers succeed in parsing ingredient lists, results are not entirely divergent.

[12] Our Benchmark Method is accessible as a Python script available in https://github.com/Pythrix/Food_Ingredient_Parser.

[13] For metric explanation see next section, "Benchmark Results"

[14] The issue is due to the presence of a mark in the ingredient list referring to a "comment" not listed in the ingredient list.

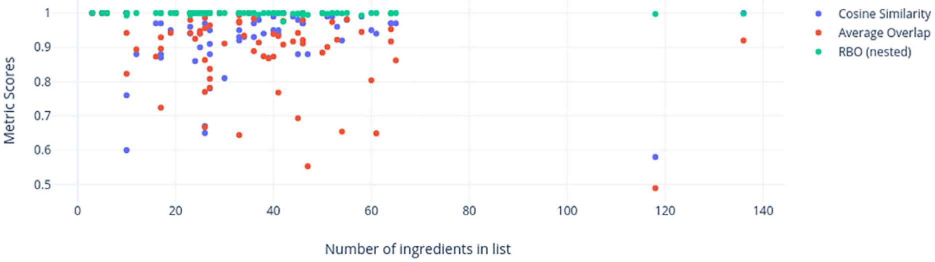


Fig. 8. Score obtained for each metric given the number of ingredients in ingredient list ($N = 100$).

Finally, the main differences between the two algorithms lie in the way additive categories are handled. In our case, we chose to consider additive categories as additional information related to a given ingredient (see for detailed information Table 1). Therefore, these elements are removed from the final output of the parser, while they are retained as ingredients in SDI. This explains particularly the values in the right side of the box plot, i.e. values outside the central 50%. When dealing with large and complex ingredient lists, the number of differences between parsers will increase as the number of additives increases in the product formulation. These differences are reflected in the proximity scores computed (see table of Parser Benchmark Results and columns “Parsers_SimScore”, Parsers_FlatAverageOverlap”, ”Parsers_NestedRBO”). The first two proximity scores were computed between the flat[15] ingredient lists obtained after parsing, both for our parser and the SDI.

The first proximity score consists in computing the cosine similarity (“Parsers_SimScore”) between flat ingredient lists obtained by the parsers. The second proximity score computed on flat ingredient lists is the Average Overlap [14]. This metric is computed on the set intersection of two ranked lists of indefinite lengths. This is a pertinent metric as it considers, in an incremental way, the depth by which each list is compared. The general equation is as follows:

$$AO(L1, L2, k) = \frac{1}{k}\sum_{d=1}^{k} A_d$$

In this equation, L1 and L2 represent two indefinite ranked lists, k is the maximum depth taken into account for evaluation, and A_d the match between the two lists at rank d.

Finally, we computed RBO on “nested ingredient lists (see columns “NestedIngList”, “OdalimNested” and “Parsers_NestedRBO” in the Parser Benchmark Results table). Nested ingredient lists are obtained by transforming ingredient dictionaries into a list where each ingredient of a given level is grouped into a sub-list at a rank equal to the level informed into the dictionary (Excerpt 8).

Both for flat ingredient lists and nested ingredient lists, RBO were computed with a p[16] parameter equal to 0.6 in order to take into account elements from higher ranks in the ingredient lists. In other words, our intention was to take an in-depth look at the proximity results for each parser.

The following figure (Fig. 8) represents the results obtained for each metric considering the complexity of the ingredient list investigated, i.e. the number of ingredients contained in the ingredient list.

As observed, the results of the two parsers are generally very similar. This being said, among the three metrics used, the Average Overlap seems to maximize the difference between each parser’s results, reflecting the differences of interpretation of the ingredient lists. It is clear that the differences observed between parser results are mainly due to the fact that in our parser

[15] Obtained by “flattening” ingredient lists dictionaries (Excerpt 7)

[16] See above, section Benchmark Method for signification of p parameter.

```
'''Raw cane sugar ° *, cocoa butter ° *, rice powder ° (rice °, sunflower oil °,
salt), grated coconut ° *, cocoa mass ° *, soy powder ° (soya °, maltodextrin °,
corn syrup °), walnuts °, cocoa powder ° (coconut milk °, maltodextrin °), green
tea leaves °, rose petals °, raspberry powder °, cocoa bean pieces ° *, sunflower
oil °, lemon powder ° (lemon juice concentrate °, corn starch °, sugar °), vanilla
bean powder °, matcha green tea powder °, rock salt, emulsifier (lecithin
(sunflower °)), emulsifier (soy lecithin °), spirulina concentrate, safflower
concentrate, star anise °, cinnamon °
* fair trade
° from certified organic agriculture'''
```

Excerpt 1. A raw food ingredient list.

```
'''Noodles: wheat flour, modified tapioca starch (contains INS 1420), refined palm
oil, modified potato starch (contains INS 1420), refined salt, defatted sesame
powder, glycerol (INS 422), soybean oil, acidity regulator (potassium carbonate
anhydrous (E501i), sodium carbonate (E500i), sodium phosphate dibasic (E339ii)),
citric acid (E330), onion extract, thickener (guar gum (E412)), purified water,
emulsifier (soy lecithin (E322i)), nature identical green tea flavour

soup: yeast extract, salt, flavour enhancer (monosodium L-glutamate (E621),
disodium 5'-ribonucleotide (E635)), soy sauce powder, white sugar, whole peanut,
spicy powder (chilli powder, garlic powder, onion powder), black pepper powder,
roasted wheat flour, chilli extract powder, red pepper powder, red pepper seed
oil, garlic flavour, palm oil, paprika

flake: rehydrated chives, dried carrot flake, dried bok choy, dried oak mushroom,
dehydrated broccoli flakes'''
```

Excerpt 2. A complex ingredient list.

```
'''noodles(wheat flour, modified tapioca starch (contains INS 1420), refined palm
oil, modified potato starch (contains INS 1420), refined salt, defatted sesame
powder, glycerol (INS 422), soybean oil, potassium carbonate anhydrous (E501i),
sodium carbonate (E500i), sodium phosphate dibasic (E339ii), citric acid (E330),
onion extract, guar gum (E412), purified water, soy lecithin (E322i), nature
identical green tea flavour),soup (yeast extract, salt, monosodium L-glutamate
(E621), disodium 5'-ribonucleotide (E635), soy sauce powder, white sugar, whole
peanut, spicy powder (chilli powder, garlic powder, onion powder), black pepper
powder, roasted wheat flour, chilli extract powder, red pepper powder, red pepper
seed oil, garlic flavour, palm oil, paprika),flake(rehydrated chives, dried carrot
flake, dried bok choy, dried oak mushroom, dehydrated broccoli flakes)'''
```

Excerpt 3. A normalised ingredient list.

additive categories are removed from the ingredient lists and retained as additional-information in ingredient dictionaries. As a consequence, there is an increase in differences for higher ranks in considered lists.

Conversely, the method we used to compute RBO on nested ingredient lists (fractional count for each RBO score computed at each level of a given ingredient list) seems to underestimate differences between parser results. This is partly due to the computation of the final score that is based on a simple fractional count method. Had we used other allocation methods we might have obtained significantly different results.

With this in mind, proximity scores computed on flat ingredient lists must be considered with caution, because the flattening transformation of ingredient lists remove information related to depth of appearance of ingredients. Nevertheless, more complex computational scores (such as RBO) that retain depth-related ingredient information, show convergent results.

Finally, the key advantages of our parsing algorithm are:

- that it can handle complex grammatical forms,
- that it can deal with heterogeneous punctuation systems in ingredient lists,

Raw Input:

```
‘’tomato* (tomato pulp*, tomato concentrate*), water, rehydrated red kidney beans* (11%),
courgettes* (8%), carrots* (7%), onions* (6%), peppers* (6%), soy proteins* (5%), rehydrated
chickpeas*, cold extracted extra virgin olive oil*, spices*, sea salt, garlic*, basil*

*ingredients from organic farming’’
```

--------------- *** ---------------

Dictionary Output:

```
{'75_1': {'rawing': 'tomato', 'level': 0, 'comment': ['ingredients from organic farming']},
'75_2': {'rawing': 'tomato pulp', 'level': 1, 'comment': ['ingredients from organic
farming']}, '75_3': {'rawing': 'tomato concentrate', 'level': 1, 'comment': ['ingredients from
organic farming']}, '75_4': {'rawing': 'water', 'level': 0}, '75_5': {'rawing': 'rehydrated
red kidney beans', 'level': 0, 'prop': ['11%'], 'comment': ['ingredients from organic
farming']}, '75_6': {'rawing': 'courgettes', 'level': 0, 'prop': ['8%'], 'comment':
['ingredients from organic farming']}, '75_7': {'rawing': 'carrots', 'level': 0, 'prop':
['7%'], 'comment': ['ingredients from organic farming']}, '75_8': {'rawing': 'onions',
'level': 0, 'prop': ['6%'], 'comment': ['ingredients from organic farming']}, '75_9':
{'rawing': 'peppers', 'level': 0, 'prop': ['6%'], 'comment': ['ingredients from organic
farming']}, '75_10': {'rawing': 'soy proteins', 'level': 0, 'prop': ['5%'], 'comment':
['ingredients from organic farming']}, '75_11': {'rawing': 'rehydrated chickpeas', 'level': 0,
'comment': ['ingredients from organic farming']}, '75_12': {'rawing': 'cold extracted extra
virgin olive oil', 'level': 0, 'comment': ['ingredients from organic farming']}, '75_13':
{'rawing': 'spices', 'level': 0, 'comment': ['ingredients from organic farming']}, '75_14':
{'rawing': 'sea salt', 'level': 0}, '75_15': {'rawing': 'garlic', 'level': 0, 'comment':
['ingredients from organic farming']}, '75_16': {'rawing': 'basil', 'level': 0, 'comment':
['ingredients from organic farming']}}
```

Excerpt 4. Parsing algorithm output. Raw ingredients list turned into a structured dictionary.

```
'''Water,milk protein concentrate, vegetable oil (canola oil, high oleic
sunflower oil, corn oil), soy protein isolate (2%), calcium caseinate(2%),
sodium(2%) caseinate(2%), vitamins and minerals (potassium citrate,
magnesium phosphate, vitamin B12)'''
```

Excerpt 5. Ingredient lists of multiple level.

```
{'15_1': {'rawing': "'water", 'level': 0},
 '15_2': {'rawing': 'milk protein concentrate', 'level': 0},
 '15_3': {'rawing': 'vegetable oil', 'level': 0},
 '15_4': {'rawing': 'canola oil', 'level': 1},
 '15_5': {'rawing': 'high oleic sunflower oil', 'level': 1},
 '15_6': {'rawing': 'corn oil', 'level': 1},
 '15_7': {'rawing': 'soy protein isolate', 'level': 0, 'prop': ['2%']},
 '15_8': {'rawing': 'calcium caseinate', 'level': 0, 'prop': ['2%']},
 '15_9': {'rawing': 'sodium', 'level': 0, 'prop': ['2%']},
 '15_10': {'rawing': 'caseinate', 'level': 0, 'prop': ['2%']},
 '15_11': {'rawing': 'vitamins and minerals', 'level': 0},
 '15_12': {'rawing': 'potassium citrate', 'level': 1},
 '15_13': {'rawing': 'magnesium phosphate', 'level': 1},
 '15_14': {'rawing': 'vitamin b12', 'level': 1}}
```

Excerpt 6. Dictionary of an ingredient list of multiple level.

- that it can reduce the complexity of ingredient lists while retaining the maximum information possible.

Consequently, we believe that this methodology (parsing algorithms and the benchmark method) will contribute to the work of any researchers, agencies or firms needing to analyse

```
{'6557555_1': {'rawing': 'garbanzo beans',
'level': 0}, '6557555_2': {'rawing': 'lemon
juice', 'level': 0}, '6557555_3': {'rawing':
'tahini paste', 'level': 0}, '6557555_4':
{'rawing': 'jalapeno peppers', 'level': 0},
'6557555_5': {'rawing': 'garlic', 'level':
0}, '6557555_6': {'rawing': 'olive oil',
'level': 0}, '6557555_7': {'rawing':
'parsley', 'level': 0}, '6557555_8':
{'rawing': 'cilantro', 'level': 0},
'6557555_9': {'rawing': 'tomato', 'level':
0}, '6557555_10': {'rawing': 'kosher salt',
'level': 0}}
```

```
['garbanzo beans', 'lemon juice', 'tahini
paste', 'jalapeno peppers', 'garlic',
'olive oil', 'parsley', 'cilantro',
'tomato', 'kosher salt']
```

Excerpt 7. Flattening operation.

```
{'4789937_1': {'rawing': 'enriched
bleached flour', 'level': 0},
'4789937_2': {'rawing': 'wheat flour',
'level': 1}, '4789937_3': {'rawing':
'malted barley flour', 'level': 1},
'4789937_4': {'rawing': 'niacin',
'level': 1}, '4789937_5': {'rawing':
'reduced iron', 'level': 1}, '4789937_6':
{'rawing': 'thiamin mononitrate',
'level': 1}, '4789937_7': {'rawing':
'riboflavin', 'level': 1}, '4789937_8':
{'rawing': 'folic acid', 'level': 1},…}
```

```
['enriched bleached flour', ['wheat
flour', 'malted barley flour', 'niacin',
'reduced iron', 'thiamin mononitrate',
'riboflavin', 'folic acid'], …]
```

Excerpt 8. Transformation of ingredient dictionaries into ingredient nested lists.

the formulation of food products; current debates on food classification [1] are provoking an increasing demand for analyses such as these. Moreover, another challenge in informing how the agrifood sector can engage a sustainable pathway is to measure species biodiversity used in food products. Our method provides the possibility to do exactly that.

Ethics Statement

Not concerned.

Declaration of Competing Interest

The authors declare they accessed the primary data used thanks to a subscription license with Mintel GNPD. Authors have no personal relationships or economic interest in the company that could have appeared to influence the work reported in this paper.

Data Availability

PulseFoodInnovation (Reference data) (https://data.inrae.fr).

CRediT Author Statement

Tristan Salord: Conceptualization, Visualization, Methodology, Investigation, Writing – review & editing, Project administration, Data curation, Writing – original draft; **Marie-Benoît**

Magrini: Conceptualization, Visualization, Methodology, Investigation, Writing – review & editing; **Guillaume Cabanac:** Conceptualization, Visualization, Methodology, Investigation, Writing – review & editing, Project administration, Supervision, Validation.

Acknowledgment

Thanks to Alice Thomson-Thibault for her English editing of the manuscript.
Thanks to the support of the Region Occitanie through the KING project

Supplementary Materials

Supplementary material associated with this article can be found in the online version at doi:10.1016/j.dib.2022.108173.

References

[1] G. Cusworth, T. Garnett, J. Lorimer, Legume dreams: the contested futures of sustainable plant-based food systems in Europe, Glob. Environ. Change 69 (2021) 102321, doi:10.1016/j.gloenvcha.2021.102321.

[2] R.D. Semba, R. Ramsing, N. Rahman, K. Kraemer, M.W. Bloem, Legumes as a sustainable source of protein in human diets, Glob. Food Secur. 28 (2021) 100520, doi:10.1016/j.gfs.2021.100520.

[3] B. Balázs, E. Kelemen, T. Centofanti, M.W. Vasconcelos, P.P.M. Iannetta, Integrated policy analysis to identify transformation paths to more sustainable legume-based food and feed value-chains in Europe, Agroecol. Sustain. Food Syst. 45 (2021) 931–953, doi:10.1080/21683565.2021.1884165.

[4] W. Willett, J. Rockström, B. Loken, M. Springmann, T. Lang, S. Vermeulen, T. Garnett, D. Tilman, F. DeClerck, A. Wood, M. Jonell, M. Clark, L.J. Gordon, J. Fanzo, C. Hawkes, R. Zurayk, J.A. Rivera, W. De Vries, L. Majele Sibanda, A. Afshin, A. Chaudhary, M. Herrero, R. Agustina, F. Branca, A. Lartey, S. Fan, B. Crona, E. Fox, V. Bignet, M. Troell, T. Lindahl, S. Singh, S.E. Cornell, K. Srinath Reddy, S. Narain, S. Nishtar, C.J.L. Murray, Food in the anthropocene: the EAT–lancet commission on healthy diets from sustainable food systems, Lancet 393 (2019) 447–492, doi:10.1016/S0140-6736(18)31788-4.

[5] M.B. Magrini, M. Anton, J.M. Chardigny, G. Duc, M. Duru, M.H. Jeuffroy, J.M. Meynard, V. Micard, S. Walrand, Pulses for sustainability: breaking agriculture and food sectors out of lock-in, Front. Sustain. Food Syst. 2 (2018) 64, doi:10.3389/fsufs.2018.00064.

[6] C.H. Foyer, H.M. Lam, H.T. Nguyen, K.H.M. Siddique, R.K. Varshney, T.D. Colmer, W. Cowling, H. Bramley, T.A. Mori, J.M. Hodgson, J.W. Cooper, A.J. Miller, K. Kunert, J. Vorster, C. Cullis, J.A. Ozga, M.L. Wahlqvist, Y. Liang, H. Shou, K. Shi, J. Yu, N. Fodor, B.N. Kaiser, F.L. Wong, B. Valliyodan, M.J. Considine, Neglecting legumes has compromised human health and sustainable food production, Nat. Plants 2 (2016) 16112, doi:10.1038/nplants.2016.112.

[7] M. Kaspers, Rabobank: Strong Growth Global Pulse Production Driven by Indian Demand, Rabobank, 2017.

[8] T. Salord, PulseFoodInnovation, (2021). 10.15454/KRPEI2.

[9] C.R. Sadler, T. Grassby, K. Hart, M. Raats, M. Sokolović, L. Timotijevic, Processed food classification: conceptualisation and challenges, Trends Food Sci. Technol. 112 (2021) 149–162, doi:10.1016/j.tifs.2021.02.059.

[10] J.K.C. Ahuja, Y. Li, R. Bahadur, Q. Nguyen, E. Haile, P.R. Pehrsson, IngID: a framework for parsing and systematic reporting of ingredients used in commercially packaged foods, J. Food Compos. Anal. 100 (2021) 103920, doi:10.1016/j.jfca.2021.103920.

[11] Pythrix, Pythrix/FOODCOP: FoodCop, Zenodo, 2022, doi:10.5281/ZENODO.6397383.

[12] E. Solis, Mintel global new products database (GNPD), J. Bus. Financ. Librariansh. 21 (2016) 79–82, doi:10.1080/08963568.2016.1112230.

[13] W. Webber, A. Moffat, J. Zobel, A similarity measure for indefinite rankings, ACM Trans. Inf. Syst. 28 (2010) 1–38, doi:10.1145/1852102.1852106.

[14] L. Ruback, C. Lucchese, A. Mera Caraballo, G. García, M. Casanova, C. Renso, Computing entity semantic similarity by features ranking, 2018.