The LScD (Leicester Scientific Dictionary)

April 2020 by Neslihan Suzen, PhD student at the University of Leicester (ns433@leicester.ac.uk)
Supervised by Prof Alexander Gorban and Dr Evgeny Mirkes

[Version 3] The third version of LScD (Leicester Scientific Dictionary) is created from the updated LSC (Leicester Scientific Corpus) - Version 2*. All pre-processing steps applied to build the new version of the dictionary are the same as in Version 2** and can be found in description of Version 2 below. We did not repeat the explanation. After pre-processing steps, the total number of unique words in the new version of the dictionary is 972,060. The files provided with this description are also same as described as for LScD Version 2 below.

* Suzen, Neslihan (2019): LSC (Leicester Scientific Corpus). figshare. Dataset. https://doi.org/10.25392/leicester.data.9449639.v2

** Suzen, Neslihan (2019): LScD (Leicester Scientific Dictionary). figshare. Dataset. https://doi.org/10.25392/leicester.data.9746900.v2

[Version 2]

Getting Started

This document provides the pre-processing steps for creating an ordered list of words from the LSC (Leicester Scientific Corpus) [1] and the description of LScD (Leicester Scientific Dictionary). This dictionary is created to be used in future work on the quantification of the meaning of research texts. R code for producing the dictionary from LSC and instructions for usage of the code are available in [2]. The code can be also used for list of texts from other sources. In this case, amendments to the code may be required. Further information can be found in next sections of this report.

LSC is a collection of abstracts of articles and proceeding papers published in 2014 and indexed by the Web of Science (WoS) database [3]. Each document contains title, list of authors, list of categories, list of research areas, and times cited. The corpus contains only documents in English. The corpus was collected in July 2018 and contains the number of citations from publication date to July 2018. The total number of documents in LSC is **1,673,824**.

All documents in LSC have nonempty abstract, title, categories, research areas and times cited in WoS databases. There are 119 documents with empty authors list, we did not exclude these documents

LScD is an ordered list of words from texts of abstracts in LSC. The dictionary is sorted by the number of documents containing the word in descending order. The dictionary stores **974,238 unique words**, where abbreviations of terminologies and words with number are contained in. All words in the dictionary are in stemmed form of words. The LScD contains the following information:

1. Unique words in abstracts in the LSC
2. Number of documents containing each word
3. Number of appearance of a word in the entire corpus.

'The number of documents containing a word' is the number of the documents with the corresponding word. A word that appears multiple times in a document is counted once (binary representation for existence). 'Number of appearance of a word in the entire corpus' is defined to be the total number of occurrences of a word in the LSC when the corpus is considered as one large document.

All words obtained after pre-processing steps are included in the LScD. The most frequent 20 words (calculated by the number of documents containing a word) are presented in Table 1.

Table 1. The most frequent 20 words (calculated by binary representation for existence) in the LScD

Word	Number of documents containing the word	Word	Number of documents containing the word
use	902,033	also	400,642
result	812,154	present	389,735
studi	723,827	increas	383,676
show	498,705	two	375,586
method	491,586	model	372,911
effect	476,757	signific	370,435
base	446,436	compar	355,381
differ	445,739	paper	346,514
can	441,512	time	344,817
high	402,737	perform	341,547

Processing the LSC

This section describes steps for creation of LScD from the LSC. The main challenge of using text data is that it is mess and not concretely structured. This means that a number of steps is needed to be taken to form the LScD. The initial step of building the dictionary is to convert unstructured text (raw corpus) into structured data. Structured data means highly organised and formatted in a way so the information contained can be easily used by data mining algorithms, mostly numerical data in relational databases. There are different ways to pre-process text data and pre-processing steps should be described for each corpus individually. Decision taken and steps of processing for creation of LScD are described below:

Step 1: Downloading the LSC Online

Use of the LSC is subject to acceptance of request of the link by email. To access the LSC for research purposes, please email to ns433@le.ac.uk. The data are extracted from Web of Science [3]. You may not copy or distribute these data in whole or in part without the written consent of Clarivate Analytics.

Step 2: Importing the Corpus to R

This is the step of reading all documents (CSV files) to R for processing the corpus. The full code including reading the LSC can be found in the GitHub [2].

All following steps can be applied for arbitrary list of texts from any source with changes of parameter. We note that the structure of the corpus such as file format and names (also the position) of fields should be taken into account to apply our code. The organisation of CSV files of LSC is described in data description file for LSC [1].

Step 3: Extracting Abstracts and Saving Metadata

After importing all CSV files to R, metadata that include all fields in a document excluding abstracts and the field of abstracts are separated. Metadata are then saved as MetaData.R to the output directory. Fields of metadata are: List_of_Authors, Title, Categories, Research_Areas, Total_Times_Cited and Times_cited_in_Core_Collection.

Step 4: Text Pre-processing Steps on the Collection of Abstracts

Text pre-processing means to bring the text into a form of analysable for the task. This step is highly important for transferring text from human language to machine analysable format by data mining algorithms. As each task requires different procedures to process the text based on aim of the study, ideal pre-processing procedure of each task should be developed individually. In this section, we presented our approaches to pre-process abstracts of the LSC.

1. **Removing punctuations and special characters:** This is the process of substitution of all non-alphanumeric characters by space. We did not substitute the character "-" in this step, because we need to keep words like "z-score", "non-payment" and "pre-processing" in order not to lose the actual meaning of such words. A processing of uniting prefixes with words are performed in later steps of pre-processing.
2. **Lowercasing the text data:** Lowercasing is one of the most effective pre-processing step in text mining problems to avoid considering same words like "Corpus", "corpus" and "CORPUS" differently. Entire collection of texts are converted to lowercase.
3. **Uniting prefixes of words:** Prefixes are letters placed before a word to create a new word with different meaning. Words containing prefixes joined with character "-" are united as a word. The list of prefixes united for this research are listed in the file "list_of_prefixes.csv". The most of prefixes are extracted from [4]. We also added commonly used prefixes: 'e', 'extra', 'per', 'self' and 'ultra'.
4. **Substitution of words:** Some of words joined with "-" in the abstracts of the LSC require an additional process of substitution to avoid losing the meaning of the word before removing the character "-". Some examples of such words are "z-test", "well-known" and "chi-square". These words have been substituted to "ztest", "wellknown" and "chisquare". Identification of such words is done by sampling of abstracts form LSC. The full list of such words and decision taken for substitution are presented in the file "list_of_substitution.csv".
5. **Removing the character "-":** All remaining character "-" are replaced by space.
6. **Removing numbers:** All digits which are not included in a word are replaced by space. All words that contain digits and letters are kept for this study because alphanumeric characters such as chemical formula might be important for our analysis. Some examples for words containing numbers are "co2", "h2o", "1990s", "zn2" and "21st".
7. **Stemming:** Stemming is the process of converting inflected words into their word stem. In this process, multiple forms of a specific word are eliminated and words that have the same base in different grammatical forms are mapped to the same stem. As stemming removes suffixes and reduces the number of words in corpus, this step results in uniting several forms of words with similar meaning into one form and also saving memory space and time [5]. For instance, the word "listen" is the word stem for "listens", "listened", and "listening". All words in the LScD are stemmed to their word stem.
8. **Stop words removal:** In natural language processing, *stop words* (function words) are defined as words that are extreme common but provide little value in a language. Some common stop words in English are 'I', 'the', 'a' etc. Such words appear to be of little informative in documents matching as all documents are likely to include them. In our research, we used 'tm' package in R to remove stop words [6]. There are 174 English stop words listed in the package.

Step 5: Writing the LScD into CSV Format

After pre-processing the abstracts of LSC, there are **1,673,824** plain processed texts for further analysis. All unique words in the corpus are extracted and written in the file "LScD.csv". This file also contains the number of documents containing each word (binary representation for existence) and the number of appearance of a word in the entire corpus.

The Organisation of the LScD and Fields in the CSV File

After pre-processing steps, all words are written in the file "LScD.csv". The total number of words in the file is **974,238**.

In the CSV file, unique words, the number of documents containing the word and the number of appearance of the word in the corpus are recorded on each line in separated fields. Each field is described below:

Word: It contains unique words from the corpus. All words are in lowercase and their stem forms. The field is sorted by the number of documents that contain words in descending order.

Number of Documents Containing the Word: This field contains the number of documents containing the corresponding word in 'Word' field. In this content, binary calculation is used: if a word exists in an abstract then there is a count of 1. If the word exits more than once in a document, the count is still 1. Total number of document containing the word is counted as the sum of 1s in the entire corpus.

Number of Appearance in Corpus: A word can appear many times in the same document. Number of appearance of a word is computed as the sum of appearance of the word in each document. The field contains how many times a word occurs in the corpus when the corpus is considered as one large document.

Instructions for R Code: Implementing the Code to LSC

This section discusses how to use LScD_Creation.R to implement data processing for building the dictionary from the LSC. The code and a detailed explanation can be accessed in [2].

LScD_Creation.R is an R script for processing the LSC to create an ordered list of words from the corpus. All outputs of the code are saved as RData file. LScD is also saved in CSV format. Outputs of the code are:

Metadata File: Metadata include all fields in a document from LSC excluding abstracts. Fields are List_of_Authors, Title, Categories, Research_Areas, Total_Times_Cited and Times_cited_in_Core_Collection.

File of Abstracts: It contains all abstracts in the corpus after pre-processing steps defined in the step 4 (see the code for the full code of pre-processing steps).

DTM: DTM is the Document Term Matrix constructed from the LSC [6]. In DTM, rows correspond to documents in the collection and columns correspond to terms (words). Each entry of the matrix is the number of times the word occurs in the corresponding document.

LScD: This file contains of ordered list of words from LSC as defined in the previous section.

The code can be used in the following way:

1. Download the folder 'LSC', 'list_of_prefixes.csv' and 'list_of_substitution.csv'
2. Open the LScD_Creation.R script

3. Change parameters in the script: replace with the full path of the directory with source files and the full path of the directory to which you write output files

```
# Directory with source files (.csv)
sourceDir = "/uol.le.ac.uk/LSC"
# Directory to write metadata files and abstracts (processed documents)
outDirectory = "/uol.le.ac.uk/LScD(processed)"
# File name of List of prefixes
prefFileName = "/uol.le.ac.uk/list_of_prefixes.csv"
# File name of List of substitution
substFileName = "/uol.le.ac.uk/list_of_substitution.csv"
```

4. Run the full code.

Warning

In pre-processing steps, all words including digits are kept for this study. One type of such words is scientific notation of numbers. Scientific notation of numbers are saved in the LScD. In scientific notation, the letter E (exponent) is used to mean ‘10 to the power of’. A number with E+n means the number multiplied by ’10 to the nth power’, in which n is a number. In EXCEL the number with ‘En’ is also considered as scientific notation of the number, so such words in the dictionary are displayed in scientific format in the CSV file. For instance, the word ‘0e6’ in LScD is displayed as ‘0.00E+00’ in the EXCEL. ‘LScD.RData’ file contains words as character and also R, Python, Matlab, Java and other programming languages can read the dictionary from CSV file correctly.

References

[1] N. Suzen. (2019). LSC (Leicester Scientific Corpus) [Dataset]. Available: https://doi.org/10.25392/leicester.data.9449639.v1

[2] N. Suzen. (2019). *LScD-LEICESTER SCIENTIFIC DICTIONARY CREATION*. Available: https://github.com/neslihansuzen/LScD-LEICESTER-SCIENTIFIC-DICTIONARY-CREATION

[3] Web of Science. (15 July). Available: https://apps.webofknowledge.com/

[4] A. Thomas, "Common Prefixes, Suffixes and Roots," *Center for Development and Learning*, 2013.

[5] C. Ramasubramanian and R. Ramya, "Effective pre-processing activities in text mining using improved porter’s stemming algorithm," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 12, pp. 4536-4538, 2013.

[6] I. Feinerer, "Introduction to the tm Package Text Mining in R," *Accessible en ligne: https://cran.r-project.org/web/packages/tm/vignettes/tm.pdf*, 2013.