Bibliometric Analysis of ICLS and CSCL research

Léonore Valentine Guillain, Nour Ghalia Abassi
EPFL

***Abstract*—**

-

I. TODO write some sentence about general motivation for bibliometric analysis

We explore ways to extract information from ICLS and CSCL papers to gain an understanding about the community formed by conference participants and what research is being conducted within the community. After describing how to extract data from the original dataset in suitable format for analysis we perform an exploratory data analysis after which we derive insights into the community using social network analysis and [Nour: NLP etc].

-

II. TODO write a few centences about the most central findings

III. Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua [?]. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat [?], [?], [?]. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

To illustrate how code, equations and figures are included in the notebook, I include a simple technical section (section IV) below.

A. Data Processing

To extract information from the dataset we first need to convert the data into a suitable format. We store the extracted data in csv format, as this makes it accessible to any scientist and can be read as is on the GitHub. Only the data frame containing emails is stored in pickle format, which is done for privacy reasons.

1) Metadata:

a) Extraction: The provided raw dataset consists of papers in pdf format as well as associated metadata in xml format. This associated metadata consists of : full-length author names, abstract of the paper, associated keywords, document type, publisher, title of the paper, time of issue and the citation string for the paper. We extract the xml metadata using the lxmltree python library into a csv file containing the above listed columns. The provided xml does not exactly follow the xml standard, hence we handle errors by iteratively parsing the file.

```
Number of papers is : 869
```

Some of the documents are not readable and have some strange format so we need to take them out of the analysis. Wwe find ourselves with roughly 834 documents

B. Data Cleaning

your part for the references universities parsing ...

1) Text preprocessing: From the pdf parser some of the documents were corrupted with some weird caracters such as , white space strings

- We used ***beautiful soup*** and ***string*** libraries to only keep the text that we needed without any special character.
- we applied ***regular expression*** to remove numbers because we thought that numbers were not that important.
- We put all the words to lower cases
- We removed the words that were connecting parts of a sentence rather than showing subjects, objects or intent. For us these words were not that important and were only going to increase the bag of words length
- In some methods we used ***Lemmatization*** and ***stemming*** and in other methods we didn't because we wanted to do a mapping with ***GloVe*** vocabulary which we will explain later on.

Lemmatization is a word variations to the root of the word (e.g. working, works, worked changed to work). ***Stemming*** is the process of reducing inflected and derived words to their word stem, base or root form—generally a written word form As the use of the stop words to reduce the BOW (bag of words size we also used lemmatization and stemming to be able to reduce the number of words without taking off any information)

Lemmatization and different form of ***Stemming*** were only used with ***TF-IDF*** while doing the Keywords extraction, otherwise we prefered keeping the words as they are because we were doing a mapping with a pretrained stanford word embedding and we did not wand to loose interpretability.

C. Keywords extraction

For the Keywords extraction we used three methods. The first one whe is a trivial one is to search in the pdf itself for keywords and extract them, this method is the most accurate but we found out that for 492 papers we had no keywords included. So we deciced to use two methods for keywords extraction. The first one is ***TF-IDF*** and the second one is ***Rake***.

- ***TF-IDF***: Tf–idf is one of the most popular term-weighting schemes today, in information retrieval, tf–idf or TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.

With TF-IDF we sorted for each document the words with the highest score and then selected the the 7 first words for each document.

- ***Rake***: Rake stands for Rapid Automatic Keyword Extraction. It is an existing python implementation that uses the NLTK toolkit for the score calculations.

How it works : * It splits the text into sentences and generates some words as candidates.
* Various punctuation signs will be treated as sentence boundaries. * All words listed in the stopwords file will be treated as phrase boundaries. This helps generate candidates that consist of one or more non-stopwords. However ,it won't work in cases where the stopword is part of the phrase. For example, 'new' is listed in RAKE's stopword list. This means that neither 'New York' nor 'New Zealand' can be ever a keyword. * For each keyword candidate generated, the algorithm computes the score of the pretended keyword, which is the sum of the scores for each of its words (if it's a composed word). The words are scored according to their frequency and the typical length of a candidate phrase in which they appear. * The last step is ranking the scores ansd selecting the ones with the highest score

Keywords in text: Select from the text the keywords already given by the authors of the paper.

Once we have all the keywords from the three methods, we tried to select the most accurate keywords. So we created a method that selects the intersection of TF-IDF keywords and the Rake keywords with the keywords of the authors (if there are any keywords in the text), if the intersection is empty and we have the keywords in the text we only take the keywords of the text. If we don't have any keywords we take the intersection of the TF-IDF and the Rake. If again, the intersection is empty then we only take the TF-IDF keywords because they were closer to the text keywords than the Rake Keywords.

Once we have our final keywords, we created a method that goes into the metadata and put assign to every document a publication year.

Since we don't have the same number of documents for each year we needed to scale over the years and use the percentage of appearence of a word. So after groupping the words by year and scaling them we found graph below

```
Out[33]:
```

We wanted then to know what are the trends of words for each specific year, so we selected distinct words for each year, it means that we selected for each year the words that were present for that year and never for the others. We get the graph below. As we can see from the graph, the most used word in the papers, more than 10%

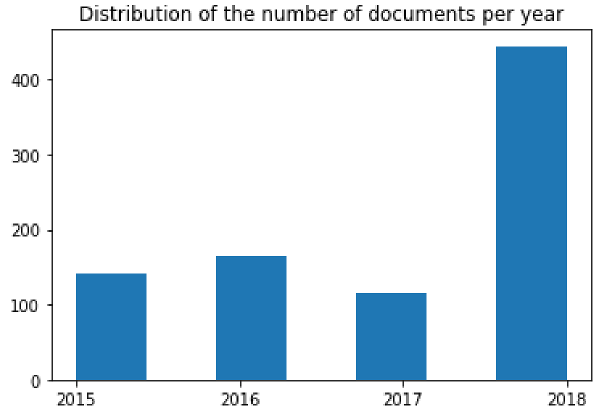

Figure 1

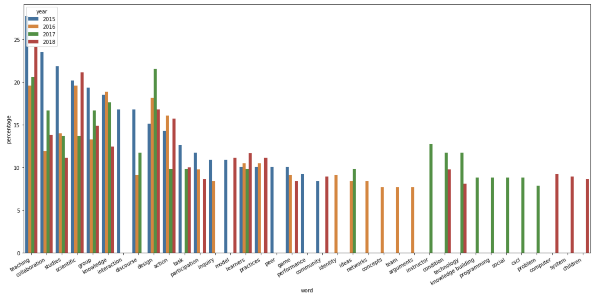

Figure 2

of 2015 papers uses the word ’peer’, more than 6% of the papers uses the words ’building’, ’embodied’ in 2016. In 2017 the most frequent keyword on is ’reponsive’ and in 2018 ’tools’ is present with more than 6%.

```
Out[37]:
```

D. Document Clustering

IV. Illustrative Example

For the purposes of illustration, consider a damped harmonic oscillator driven by a sinusoidal signal:

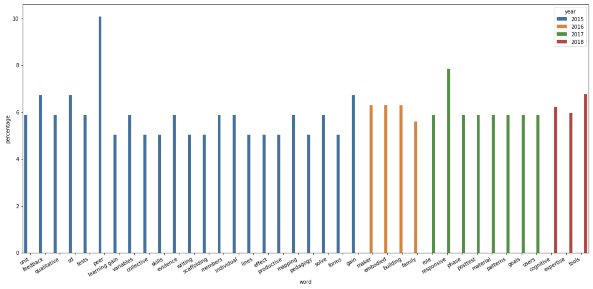

Figure 3

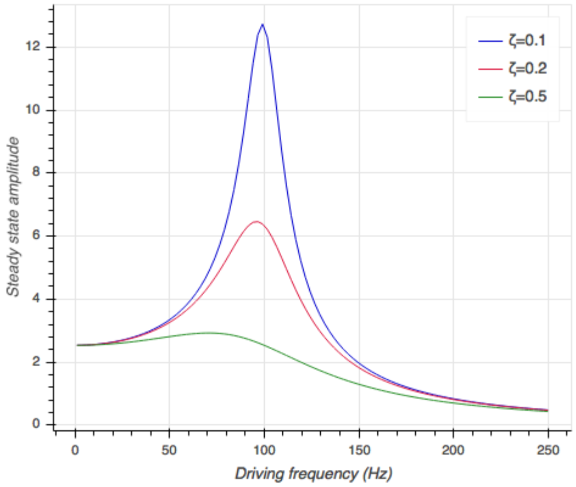

Figure 4: Variation of steady-state oscillation amplitude with driving frequency for an oscillator with undamped frequency of 100 Hz, a driving amplitude $A = 10^6$, and various damping constants.

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + 2\zeta\omega_0\frac{\mathrm{d}x}{\mathrm{d}t} + \omega_0^2 x = A\sin(\omega t), \tag{1}$$

where $x(t)$ is the quantity of interest, ζ is the damping constant, ω_0 is the undamped resonant frequency, and $A\sin(\omega t)$ is the driving sinusoidal signal. The steady state solution to (1) is given by:

$$\begin{aligned} x(t) &= \frac{A}{Z_m\omega}\sin(\omega t + \phi), \qquad (2) \\ Z_m &= \sqrt{(2\omega_0\zeta)^2 + \frac{(\omega^2-\omega_0^2)^2}{\omega^2}}, \\ \phi &= \arctan\left(\frac{2\omega\omega_0\zeta}{\omega^2-\omega_0^2}\right). \end{aligned}$$

In Figure 4 we plot the steady state amplitude of a 100 Hz oscillator as a function of driving frequency. We see that the oscillator exhibits a strong resonance for small values of ζ, and that the resonance frequency is slightly lower than the driving frequency of 100 Hz.

Algorithm 1 Calculate $y = x^n + z^m$

```
Require: $n \geq 0 \vee x \neq 0 \vee z \geq 0 \vee m \neq 0$
Ensure: $y = x^n + z^m$
  $y \leftarrow 1$
  if $n < 0$ then
    $N \leftarrow -n$
  else
    $N \leftarrow n$
  end if
```

V. Conclusions

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

References