

Topic Modeling of Course Content

Raphaël Morsomme

2018-12-28

Contents

1	Introduction	3
2	Data Preparation	3
2.1	Overview	3
2.2	Importing	3
2.3	Extracting Course Descriptions from Course Catalogues	4
2.4	Tidying	6
2.5	Stemming	6
3	Analysis	7
3.1	Overview	7
3.2	TF-IDF	7
3.2.1	Results	8
3.3	Topic Emergence	13
3.3.1	Results	13
3.4	LDA	16
3.4.1	Fitting Model	16
3.4.2	Results	16

```
library(tidyverse)
library(tidytext)

library(ggwordcloud) # Word Clouds
library(hunspell)    # Stemmer
library(topicmodels) # LDA
library(lemon)        # fine tune ggplot
library(tm)           # Corpus()
```

1 Introduction

University departments often have little knowledge of the actual content of the programs they offer. Yet, having a good understanding of what each course of a program covers is paramount to maintain the quality of the program.

In this script, I conduct a topic modeling exercise of the curriculum offered by the University College Maastricht (UCM), Maastricht University, the Netherlands. UCM offers a bachelor in Liberal Arts and Science. Its curriculum contains over two hundred courses on virtually every topic conceivable¹, making it a great subject for a topic modeling exercise. The analysis is exploratory in nature: instead of answering a specific research question, I explore the data to obtain a better understanding of the content of UCM's curriculum.

To accomplish this, I conduct three analyses. First, I use the tf-idf to identify the most distinctive terms of each course and cluster of courses. Then, I compare the content of the 2014-2015 and 2018-2019 course catalogues to identify the themes that have emerged and declined these last few years. Finally, I use the Latent Dirichlet Allocation algorithm to create a topic model of the 2018-2019 curriculum, both at the course- and at the cluster-level. The data I use in the analyses are the course descriptions present in the course catalogues.

2 Data Preparation

2.1 Overview

The data require some preparation before they can be analyzed. We start by importing a dataset on the courses and the course catalogues (saved as pdf) from the directory. We then extract the descriptions of the courses from the course catalogues. These descriptions are one to two pages long and form the textual data that we will analyze. Lastly, we transform the data into the *tidy text format* and stem the terms.

2.2 Importing

We import two datasets from the directory. `d_course` is a tibble indicating the code, name and cluster² of each course³. `corpus` is a corpus containing the five most recent course catalogues of UCM. Course catalogues are published every year and contain a description of each course of one or two pages.

```
d_course <- read_csv("Course.csv", col_type = cols())
```

¹Ranging from artificial intelligence to Shakespeare and terrorism.

²Courses are distributed among 17 clusters e.g. International Relation, Cultural Studies, Biomedical Science, etc.

³It also includes a variable with a shorter course title (`Title_short`) which we use in the plots for a better readability.

```
corpus <- Corpus(x = DirSource("Catalogues"),
  readerControl = list(reader = readPDF(control = list(text = "-layout"))))
```

2.3 Extracting Course Descriptions from Course Catalogues

We extract the description of each course from the course catalogues. The code is a little longish and does not add much to the script, so the reader can skip this section.

In the **Setup**, we create several objects that we use in the following two loops. We first loop (**Loop 1**) through each course catalogue present in `corpus`. In the loop, we first identify the section of the catalogue containing the course descriptions and assign it to `cat_description`. We then exclude from `cat_descriptions` the headers, appendix, etc, so that it only contains course descriptions. We take advantage of the fact that the first line of a course description always starts with the code of the course, meaning that the first three characters of the first page of a description are one of the following COR, HUM, SCI, SSC, SKI, PRO, UGR, CAP. Using `grep()`, we can therefore easily identify the first page of each description which we assign to `first_pages_description`.

We then loop (**Loop 2**) through the first pages of the course descriptions. In the loop, we check if the description is one- or two-page long: if the following page is the first page of a description, then the description is only one page long; otherwise, the description is two page long and consists of the current and the following page⁴. Once we have a course description, we use `substring(first = 1, last = 7)` to extract the code of the course (3 letters and 4 numbers) and save the description, with its code and the year of the catalogue in the tibble `d_description`.

```
# Setup
n_catalogue <- length(corpus)
calendar_years <- c("2014-2015", "2015-2016", "2016-2017", "2017-2018", "2018-2019")
d_description <- tibble(Code = character(0),
  `Calendar Year` = character(0),
  Description = character(0))

course_code <- c("COR", "HUM", "SCI", "SSC", "SKI", "PRO", "UGR", "CAP")
course_code <- paste(course_code, collapse = "|")

content_to_exclude <- c("^Core Courses \\\(COR\\)",
  "^Humanities \\\(HUM\\)",
  "^Sciences \\\(SCI\\)",
  "^Social Sciences \\\(SSC\\)",
  "^Skills Trainings \\\(SKI\\)",
  "^Skills Training \\\(SKI\\)",
  "^Project \\\(PRO\\)",
  "^Projects \\\(PRO\\)",
  "^Undergraduate Research \\\(UGR\\)",
  " UCM Undergraduate\\r\\nResearch",
  "UCM Undergraduate\\r\\n Research"
)

content_to_exclude <- paste(content_to_exclude, collapse = "|")

# Loop 1
for(n in 1 : n_catalogue){
```

⁴No course description exceeds two pages.

```

cat <- content(corpus[[n]])

# Description Section
page_description_start <- grep(pattern = "^Core Courses \\(COR\\)", x = cat) + 1
page_description_end <- grep(pattern = "Appendix", x = cat) - 1
if(length(page_description_end) == 0) page_description_end <- length(cat)
cat_description <- cat[page_description_start : page_description_end]

# Course Descriptions Only
pages_to_exclude <- grep(pattern = content_to_exclude, x = cat_description)
cat_description <- cat_description[!1:length(cat_description) %in% pages_to_exclude]

# First Pages of Course Descriptions
first_three_letters <- substr(cat_description, start = 1, stop = 3)
first_pages_description <- grep(pattern = course_code, x = first_three_letters)

# Loop 2
for(page in first_pages_description){

  # Extract Description
  following_page <- page + 1 # for convenience
  if(following_page %in% first_pages_description){
    description <- cat_description[page]
  }else{
    description <- paste(cat_description[page : following_page], collapse = " ")
  }

  # Save Description
  Code <- substring(description, first = 1, last = 7)
  year <- calendar_years[n]
  d_description <- d_description %>%
    add_row(Code = Code,
            `Calendar Year` = year,
            Description = description)

} # close for-loop 2 (page)
} # close for-loop 1 (n)

print(d_description)

```

```

## # A tibble: 831 x 3
##   Code   `Calendar Year` Description
##   <chr>   <chr>         <chr>
## 1 COR1002 2014-2015 "COR1002 - Philosophy of Science\r\nCourse coo~
## 2 COR1003 2014-2015 "COR1003 - Contemporary World History\r\nCours~
## 3 COR1004 2014-2015 "COR1004 - Political Philosophy\r\nCourse coor~
## 4 COR1005 2014-2015 "COR1005 - Modeling Nature\r\nCourse coordinat~
## 5 HUM1003 2014-2015 "HUM1003 - Cultural Studies I: Doing Cultural ~
## 6 HUM1007 2014-2015 "HUM1007 - Introduction to Philosophy\r\nCours~
## 7 HUM1010 2014-2015 "HUM1010 - Common Foundations of Law in Europe~
## 8 HUM1011 2014-2015 "HUM1011 - Introduction to Art; Representation~
## 9 HUM1012 2014-2015 "HUM1012 - Pop Songs and Poetry: Theory and An~
## 10 HUM1013 2014-2015 "HUM1013 - The Idea of Europe: The Intellectua~
## # ... with 821 more rows

```

2.4 Tidying

We save the course descriptions in the tidy text format with one row per course-year-term.

```
d_description_tidy <- unnest_tokens(d_description, output = word, input = Description)
print(d_description_tidy)
```

```
## # A tibble: 340,594 x 3
##   Code   `Calendar Year` word
##   <chr>   <chr>         <chr>
## 1 COR1002 2014-2015     cor1002
## 2 COR1002 2014-2015     philosophy
## 3 COR1002 2014-2015       of
## 4 COR1002 2014-2015     science
## 5 COR1002 2014-2015     course
## 6 COR1002 2014-2015   coordinator
## 7 COR1002 2014-2015     prof
## 8 COR1002 2014-2015     dr
## 9 COR1002 2014-2015     l
## 10 COR1002 2014-2015    boon
## # ... with 340,584 more rows
```

2.5 Stemming

Lastly, we stem the terms and filter out stop words. We use the stemmer from the `hunspell` package to build a stemming function `stem_hunspell()` which takes a term as input and returns its stem. We prefer the Hunspell stemmer over the usual Snowball stemmer because it offers a more precise stemming.

Trick: dictionary-based approach to stem a large number of terms.

Since it would take too much time to apply our stemming function to all 340.000 terms of `d_description_tidy`, we use a *dictionary-based approach*. We create a dictionary that provides the stem of the 8,500 unique terms present in the dataset and then `join` the newly created dictionary and `d_description_tidy` to stem all the terms at once. This way, we greatly reduce the number of times we use the stemming function.

```
# Stemming function
stem_hunspell <- function(term) {
  # look up the term in the dictionary
  stems <- hunspell_stem(term)[[1]]

  # identify the stem
  if (length(stems) == 0) { # if no stem in dictionary, use original term
    stem <- term
  } else { # if multiple stems, use last one (most basic)
    stem <- stems[[length(stems)]]
  }

  return(stem)
}

# Dictionary
my_dictionary <- d_description_tidy %>%
  distinct(word) %>%
  mutate(word_stem = purrr::map_chr(.x = word,
```

```

        .f = stem_hunspell))

# Full join
d_description_stem <- d_description_tidy %>%
  full_join(my_dictionary, by = "word") %>%
  rename(word_original = word,
         word          = word_stem) %>%
  filter(!word %in% stop_words$word,
         !word %in% as.character(1:1e3))
print(d_description_stem) # See humanities (original) - humanity (stem)

## # A tibble: 172,162 x 4
##   Code   `Calendar Year` word_original word
##   <chr>   <chr>         <chr>      <chr>
## 1 COR1002 2014-2015      cor1002     cor1002
## 2 COR1002 2014-2015    philosophy philosophy
## 3 COR1002 2014-2015     science     science
## 4 COR1002 2014-2015  coordinator coordinator
## 5 COR1002 2014-2015      prof        prof
## 6 COR1002 2014-2015      dr           dr
## 7 COR1002 2014-2015     boon         boon
## 8 COR1002 2014-2015    faculty      faculty
## 9 COR1002 2014-2015  humanities   humanity
## 10 COR1002 2014-2015   sciences     science
## # ... with 172,152 more rows

```

3 Analysis

3.1 Overview

Now that the textual data is stored in a tidy text format and is stemmed, we can model the content of the curriculum. We conduct three analyses. First, we identify the most important terms of each course and cluster with the tf-idf. Next, we identify terms that have emerged and declined in the curriculum. Finally, we use the LDA algorithm (a popular technique for topic modeling) to build a topic model of the 2018-2019 curriculum, both at the course- and cluster-level.

3.2 TF-IDF

The tf-idf is a popular measure to identify the most important terms of each document belonging to a corpus. By penalizing terms that occur in many documents, it allows us to focus on the terms that are specific to each document. Terms which appear in a large number of course descriptions such as “learn” or “student” tell us little about the content of the course and therefore have a low tf-idf. This way, we can identify the most *distinctive* terms of each course/cluster and get a feel of the topics that they cover.

We use the function `bind_tf_idf()` to obtain the tf-idf of each term for the year 2018-2019. We then identify the most distinctive terms of each cluster and course⁵ and display them both as barplots and word clouds. In the latter, the size and the color of a term indicates its tf-idf.

```

tdm_course <- d_description_stem %>%
  filter(`Calendar Year` == "2018-2019") %>%

```

⁵We only plot a selection of twelve courses to keep the plots readable.

```

count(Code, word, sort = T) %>%
bind_tf_idf(term = word, document = Code, n = n) %>%
left_join(d_course, by = "Code")

print(tdm_course)

## # A tibble: 24,033 x 9
##   Code word      n      tf      idf tf_idf `Course Title` Cluster Title_sho~
##   <chr> <chr> <int> <dbl> <dbl> <dbl> <chr>          <chr> <chr>
## 1 SSC3~ poli~    34 0.0757  1.82  0.137 Public Policy~ Govern~ Public Po~
## 2 UGR3~ sear~    30 0.101  1.02  0.103 Undergraduate~ Methods Undergrad~
## 3 PR01~ sear~    27 0.116  1.02  0.119 Research Proj~ Methods Res. Proj~
## 4 SKI3~ conf~    26 0.0716  4.05  0.290 Preparing Con~ Skills Preparing~
## 5 SSC2~ law     26 0.120  2.10  0.253 Law and Socie~ Sociol~ Law and S~
## 6 SSC2~ conf~    26 0.0992  2.26  0.224 Conflict Reso~ Int. R~ Conflict ~
## 7 SSC3~ trade   26 0.0533  3.54  0.189 International~ Intern~ Internati~
## 8 HUM2~ memo~    25 0.0595  3.36  0.200 Cultural Reme~ Cultur~ Cultural ~
## 9 SKI2~ argu~    25 0.0992  2.51  0.249 Argumentation~ Skills Argumenta~
## 10 SSC2~ publ~   25 0.112  1.97  0.220 Public Health~ Govern~ Public He~
## # ... with 24,023 more rows

tdm_cluster <- d_description_stem %>%
  filter(`Calendar Year` == "2018-2019") %>%
  left_join(d_course, by = "Code") %>%
  count(Cluster, word, sort = T) %>%
  bind_tf_idf(term = word, document = Cluster, n = n)

print(tdm_cluster)

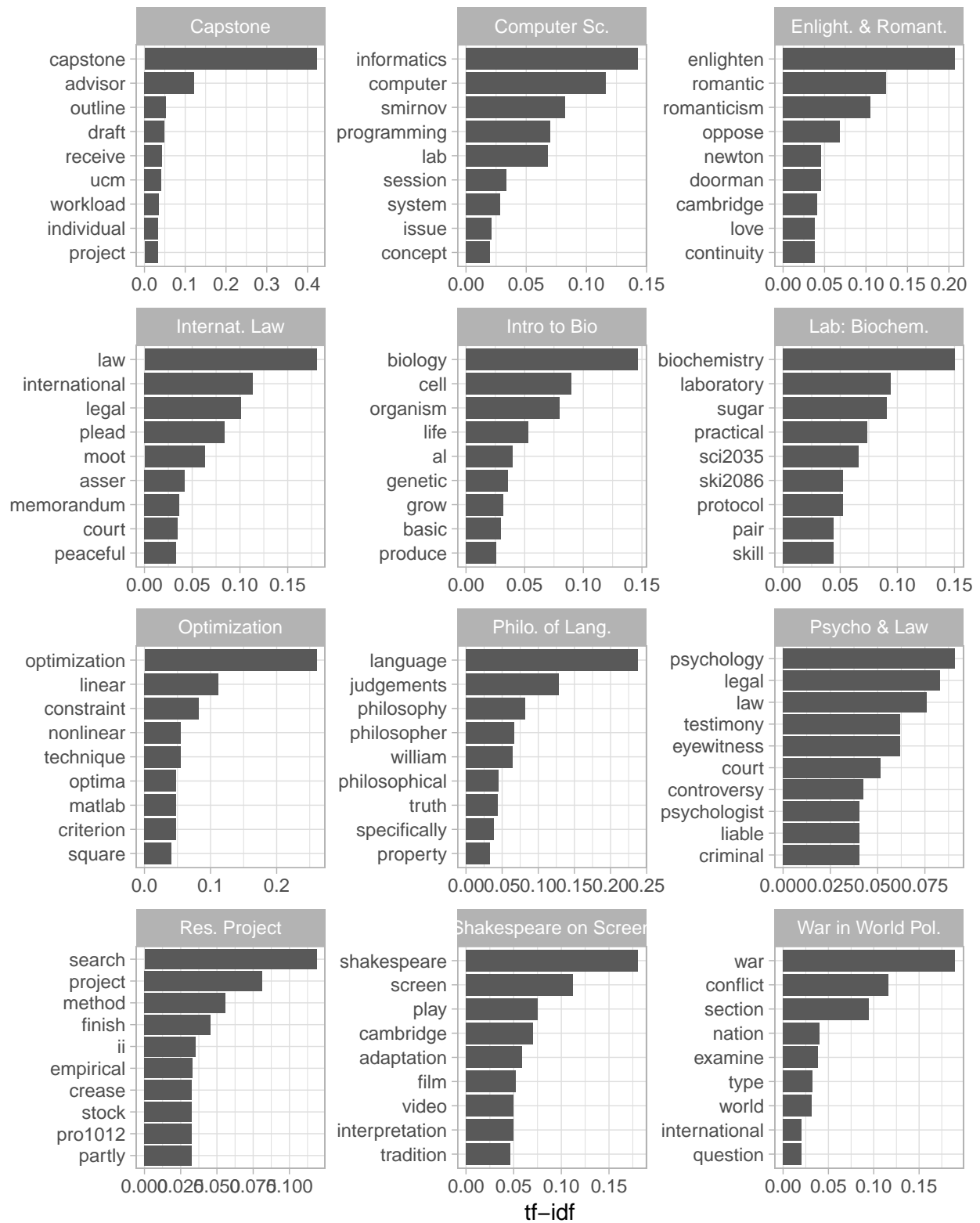
## # A tibble: 12,457 x 6
##   Cluster      word      n      tf      idf      tf_idf
##   <chr>      <chr> <int> <dbl> <dbl> <dbl>
## 1 Skills      student   195 0.0451 -0.0572 -0.00258
## 2 Methods      search   164 0.0472  0         0
## 3 Skills      skill    122 0.0282  0.125  0.00353
## 4 International Law law      102 0.0456  0.754  0.0344
## 5 Sociology    social    99 0.0326  0.0606  0.00198
## 6 Methods      student   86 0.0247 -0.0572 -0.00141
## 7 Economics    economic   81 0.0608  0.194  0.0118
## 8 Skills      project   79 0.0183  0.887  0.0162
## 9 Skills      academic   68 0.0157  0.194  0.00305
## 10 Int. Relations policy    64 0.0253  0.636  0.0161
## # ... with 12,447 more rows

```

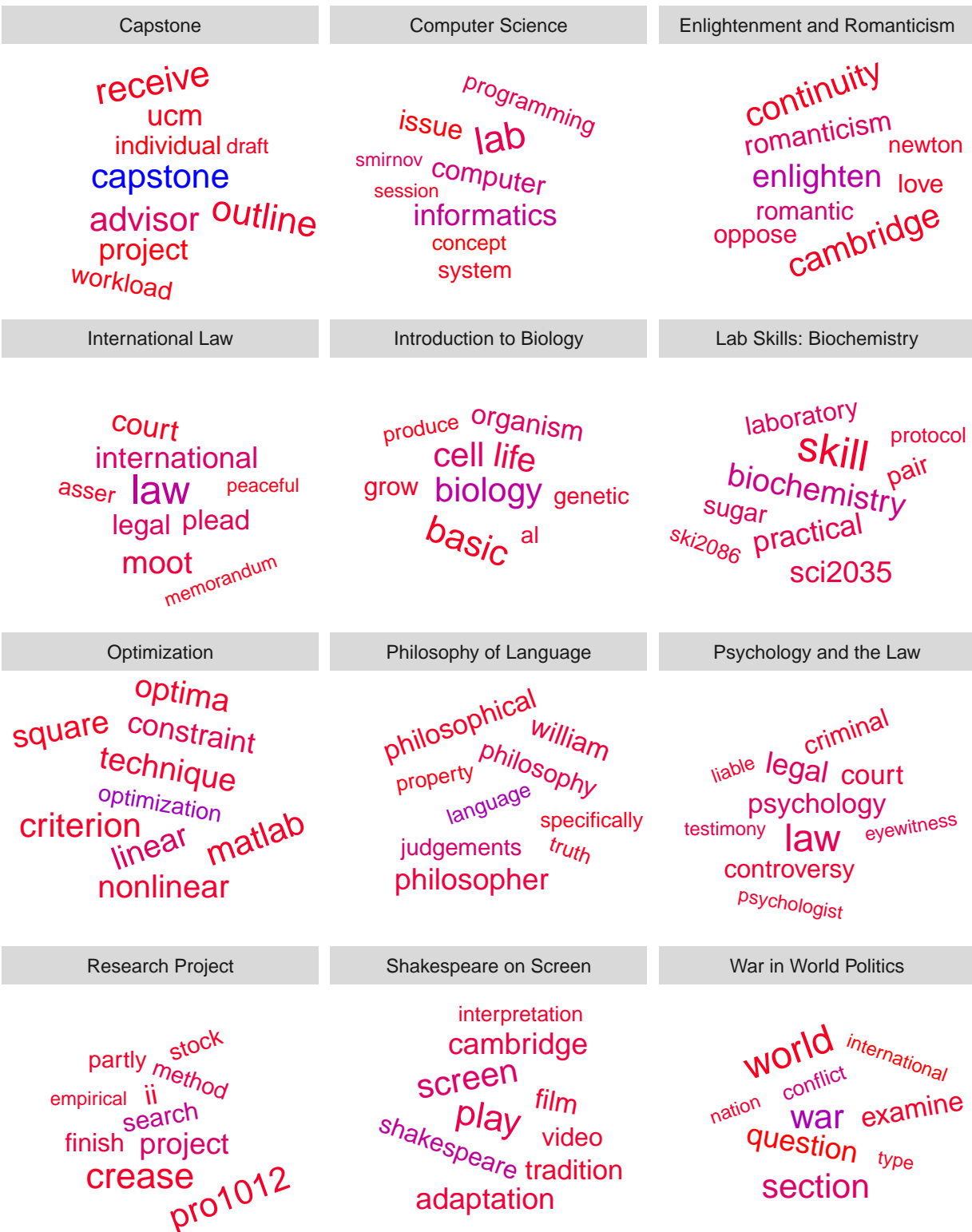
3.2.1 Results

The tf-idf does a pretty good job at isolating the most important terms of the courses/clusters. If the names of the courses/clusters were absent from the plots, it would be fairly easy to guess them from the terms. We also observe interesting elements concerning the content of the clusters. For instance, while the cluster History gives a central place to the European continent (with terms like Europe, european), the cluster International Relations focuses more on China (chinese).

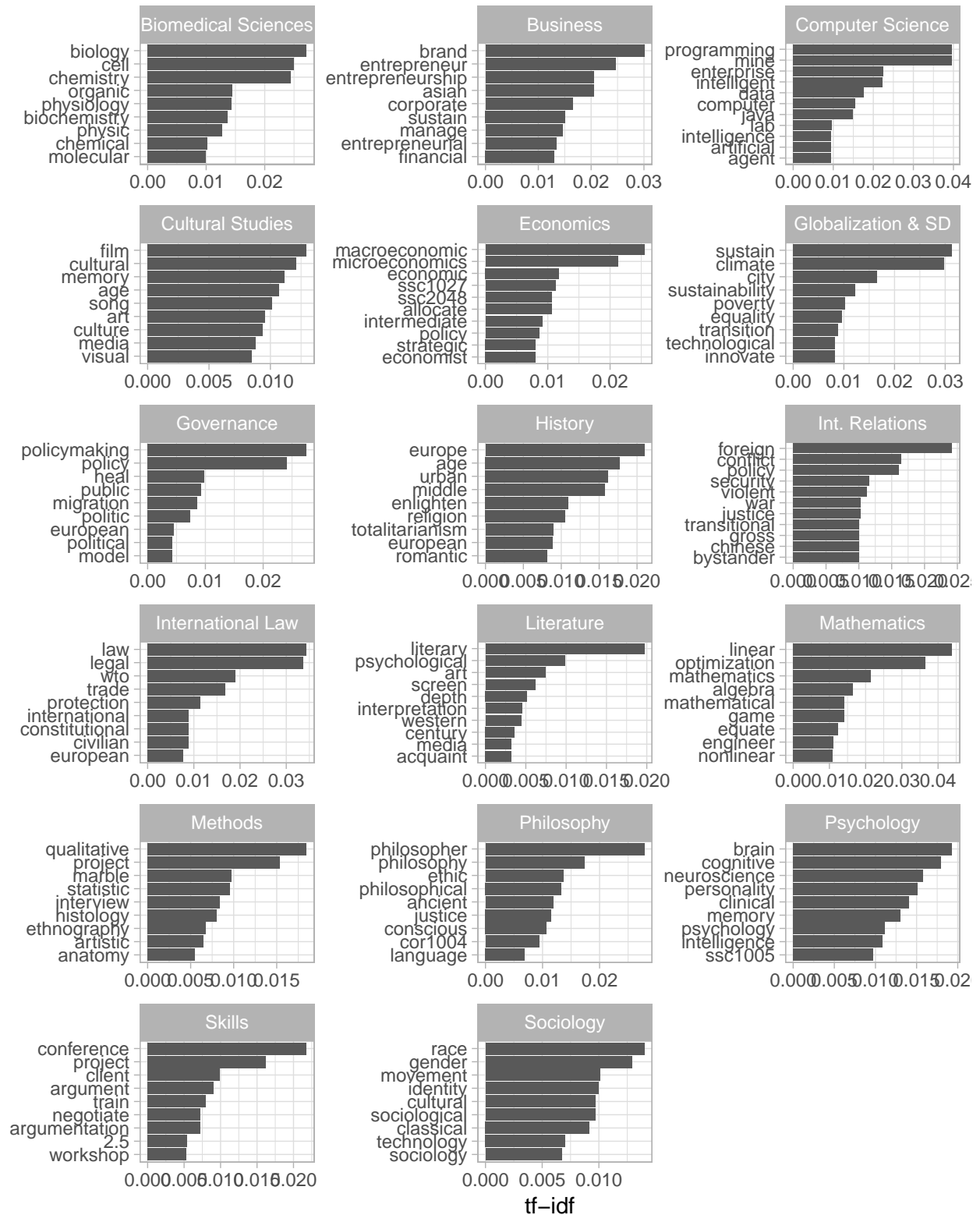
Distinctive Terms of Courses



Distinctive Terms of Courses



Distinctive Terms of Clusters



Distinctive Terms of Clusters

Biomedical Sciences	Business	Computer Science
chemical physiology organic cell biology physic chemistry molecular	financial corporate sustain asian manage brand entrepreneur	enterprise agent computer mine data lab java programming artificial
Cultural Studies	Economics	Globalization & SD
visual song culture film art media cultural age memory	microeconomics allocate macroeconomic strategic economic intermediate policy ssc1027 economist	poverty transition climate innovate sustain sustainability city equality technological
Governance	History	Int. Relations
politic heal european political policy migration public model	totalitarianism europe urban european middle age religion enlighten romantic	security transitional war foreign policy violent conflict justice chinese gross
International Law	Literature	Mathematics
protection international wto law legal civilian european trade	psychological screen media century acquaint art literary depth western interpretation	optimization algebra equate mathematical game linear mathematics engineer
Methods	Philosophy	Psychology
qualitative anatomy statistic project marble histology artistic interview	philosopher justice cor1004 philosophy ancient language ethic conscious	personality cognitive clinical psychology ssc1005 brain memory
Skills	Sociology	
2.5 workshop train argument client project negotiate conference	sociology identity cultural race gender classical	

3.3 Topic Emergence

We compare the content of the 2014-2015 (oldest available) and 2018-2019 (most recent) course catalogues to identify terms and themes that have declined and emerged these last few years. To accomplish this, we compare the frequency of the terms in the two catalogues by taking their `log ratio`. A positive value indicates that the term has become more frequent since 2014-2015 and a negative value indicates a decline in the use of the term. We plot the forty terms with the highest absolute `log ratio`. Again, we display the information both as a barplot and wordcloud.

```
d_emergence <- d_description_stem %>%
  filter(`Calendar Year` %in% c("2014-2015", "2018-2019")) %>%
  count(`Calendar Year`, `word`) %>%
  spread(key = `Calendar Year`, value = n, fill = 0) %>%
  rename(new = "2018-2019", old = "2014-2015") %>%
  mutate(n = old + new,
         log_ratio = log( (new+1) / (sum(new)+1)) /
                       ((old+1) / (sum(old)+1)) ),
         Trend = case_when(log_ratio<0 ~ "Declining",
                           log_ratio>0 ~ "Emerging")) %>%
  filter(n > 15) %>%
  top_n(50, abs(log_ratio))

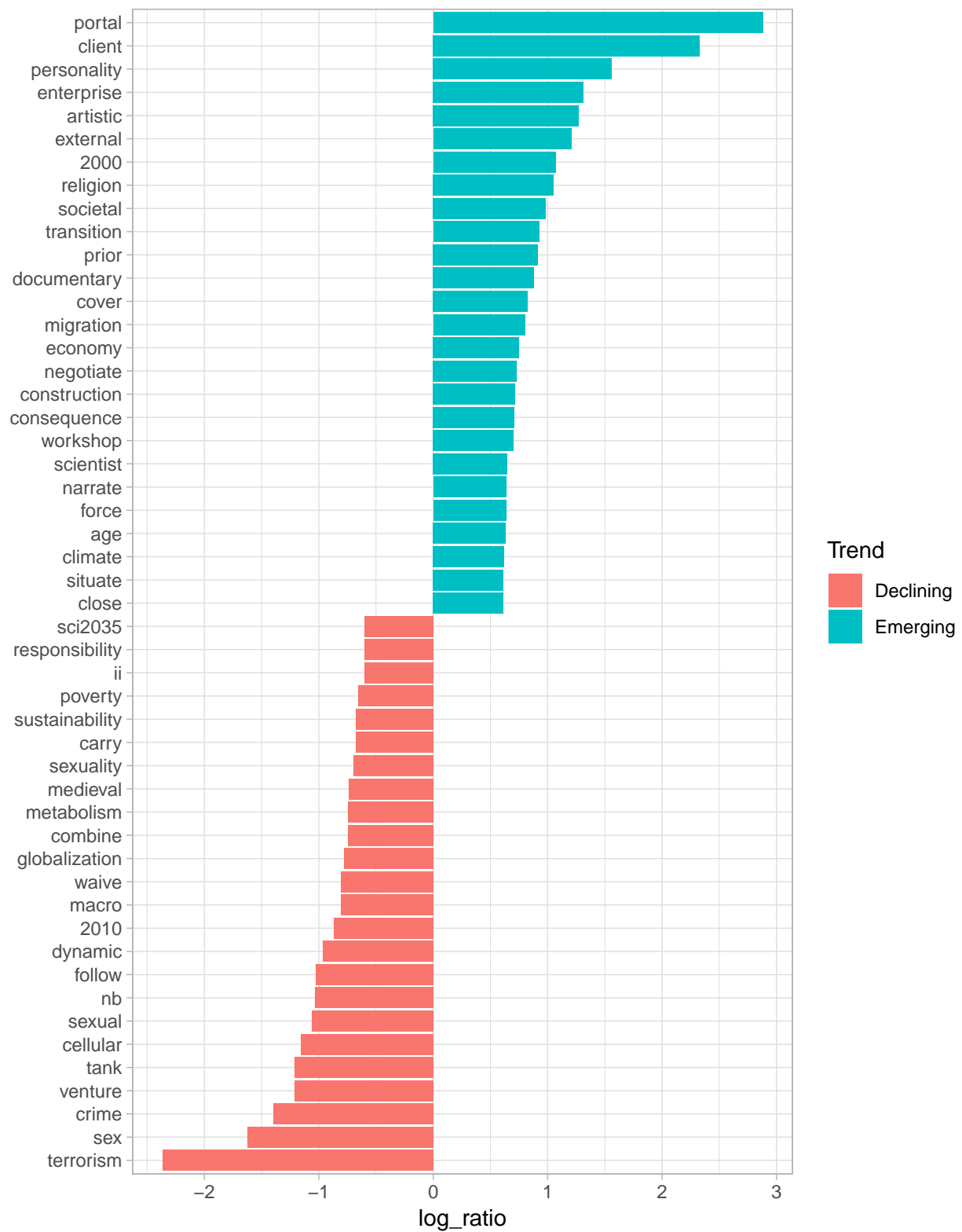
print(d_emergence)
```

```
## # A tibble: 50 x 6
##   word      old  new    n log_ratio Trend
##   <chr>   <dbl> <dbl> <dbl>   <dbl> <chr>
## 1 2000      10   35   45     1.07 Emerging
## 2 2010      16    7   23    -0.870 Declining
## 3 age       24   52   76     0.636 Emerging
## 4 artistic    4   19   23     1.27 Emerging
## 5 carry      13    7   20    -0.675 Declining
## 6 cellular   16    5   21    -1.16 Declining
## 7 client      1   22   23     2.33 Emerging
## 8 climate    10   22   32     0.622 Emerging
## 9 close      12   26   38     0.615 Emerging
## 10 combine   14    7   21    -0.744 Declining
## # ... with 40 more rows
```

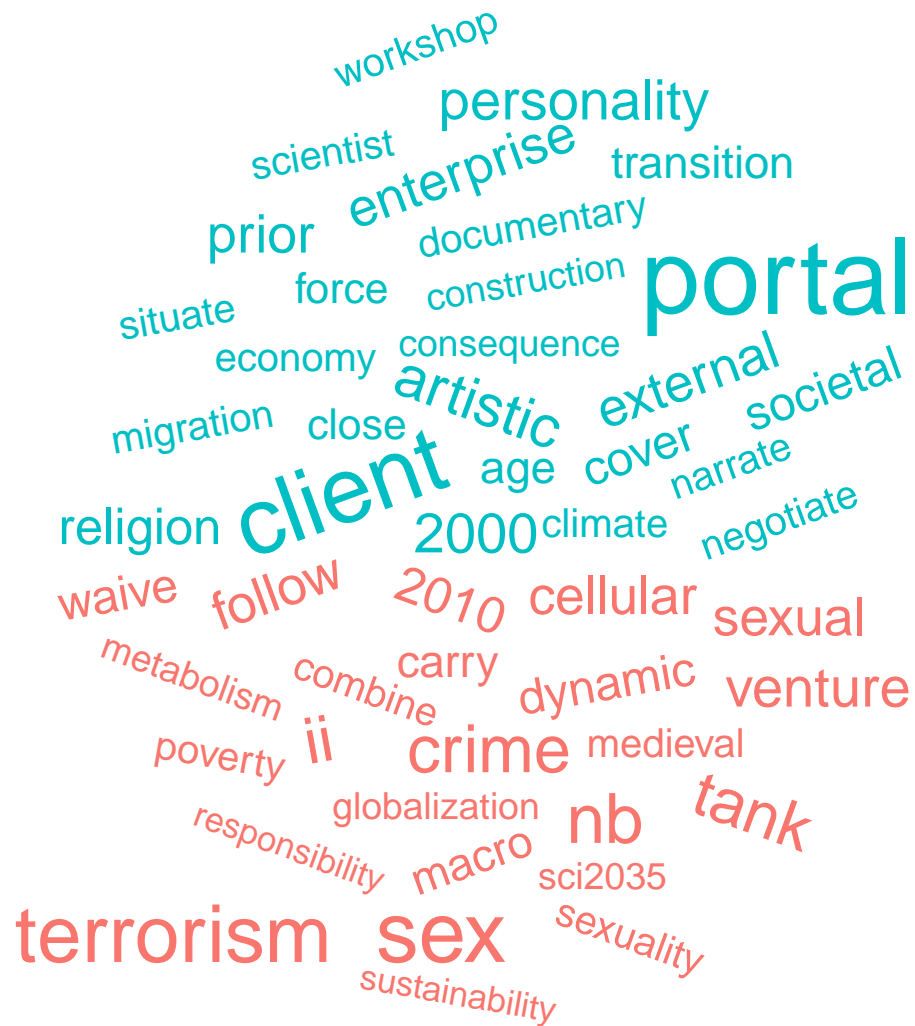
3.3.1 Results

From the two plots, we can observe that topics that have made the news these last few years such as religion, migration and climate have become more important in the catalogues. This shows that the college has done a good job at adapting its curriculum to the development of the world. We also observe that the themes of sexuality (with terms like `sex`, `sexual` and `sexuality`) and terrorism (`terrorism`, `crime`) have declined. It is interesting to note that the terms `globalization`, `poverty` and `sustainability` have become less important. As for the term `portal`, its “emergence” is due to the introduction of a new online student *portal* system at the college. The term is not mentioned once in 2014-2015 and appears 19 times in 2018-2019. This shows that human interpretation and a good knowledge of the data is crucial to avoid false alarms in such analysis.

Emerging and Declining Terms Between 2014–2015 and 2018–2019



Emerging (blue) and Declining (red) Terms Between 2014–2015 and 2018–2019



3.4 LDA

Finally, we use the Latent Dirichlet Allocation (LDA) algorithm to conduct a topic analysis of the college's curriculum at the course- and the cluster-level. Given a corpus of documents and a predetermined number of topics, the LDA algorithm outputs a topic model which gives the importance of each term to the topics (beta distribution) and the importance of each topic to the documents (gamma distribution). In other words, the LDA find the mixture of words associated with each topic and the mixture of topics associated with each document. The advantage of the LDA over regular clustering methods is that it allows for overlap of terms across topics and of topics across documents, thereby offering a model that is closer to natural language.

3.4.1 Fitting Model

In the LDA algorithm, the number of topics has to be determined in advance. We build four models with respectively 5, 12, 17 and 25 topics. For

TODO: use package `ldatuning` to determine best number of topics. (<https://cran.r-project.org/web/packages/ldatuning/vignettes/topics.html>)

```
d_cast <- d_description_stem %>%  
  filter(`Calendar Year` == "2018-2019") %>%  
  count(Code, word) %>%  
  cast_dtm(Code, word, n)
```

```
LDA_5  <- LDA(d_cast, k = 5 , control = list(seed = 123))  
LDA_12 <- LDA(d_cast, k = 12, control = list(seed = 123))  
LDA_17 <- LDA(d_cast, k = 17, control = list(seed = 123)) # There are 17 clusters  
LDA_25 <- LDA(d_cast, k = 25, control = list(seed = 123))
```

3.4.2 Results

We present the output of a topic model with three plots which respectively show (i) the most important terms for each topic (ii) the main courses/clusters of each topic and (iii) the most important topics of each course/cluster. For each model, we present the results at the course- and the cluster-level. To keep the script concise, we only include a selection of plots for the model with 12 topics. In the first four plots, the topics are unlabelled; in the last three, the topics are labelled. The complete output of each model (two triplets of plots with unlabelled topics) can be found in the directory.

Firstly, the third plot shows that most topics are covered in several clusters. For instance, **topic 10** is covered in several clusters of the humanities (**History**, **Literature** and **Philosophy**) and social sciences (**International Relations**, **International Law** and **Economics**). For a liberal arts program this is a desirable outcome since it encourages students interested in a particular topic to take classes in different clusters, thereby broadening their academic horizon⁶. At the same time, this pattern may be artificially created by the fact that there are much more than twelve topics present in the curriculum, meaning that the LDA algorithm combines unrelated themes into the same topic. This is for instance the case for **topic 3** which combines the themes of law (**law**, **legal**) and international (**european**, **international**) (see first plot). Increasing the number of topics in the model should solve this issue.

We also observe that the distribution of topics is very different at the course and the cluster level. While courses are usually heavily dominated by a single topic, clusters contain several major topics. Looking at the second plot, we observe that the courses **Computer Science**, **Optimization** and **Philosophy of Language** for instance are heavily dominated by **topic 7**. The fourth plot shows that most clusters contain several topics. Interestingly, this graph shows that the same topic (**topic 10**) dominates the clusters **History**,

⁶One of the objectives of the program.

International Relations and Literature. This indicates either that the content of the three clusters share some similarity or that **topic 10** contains several different themes.

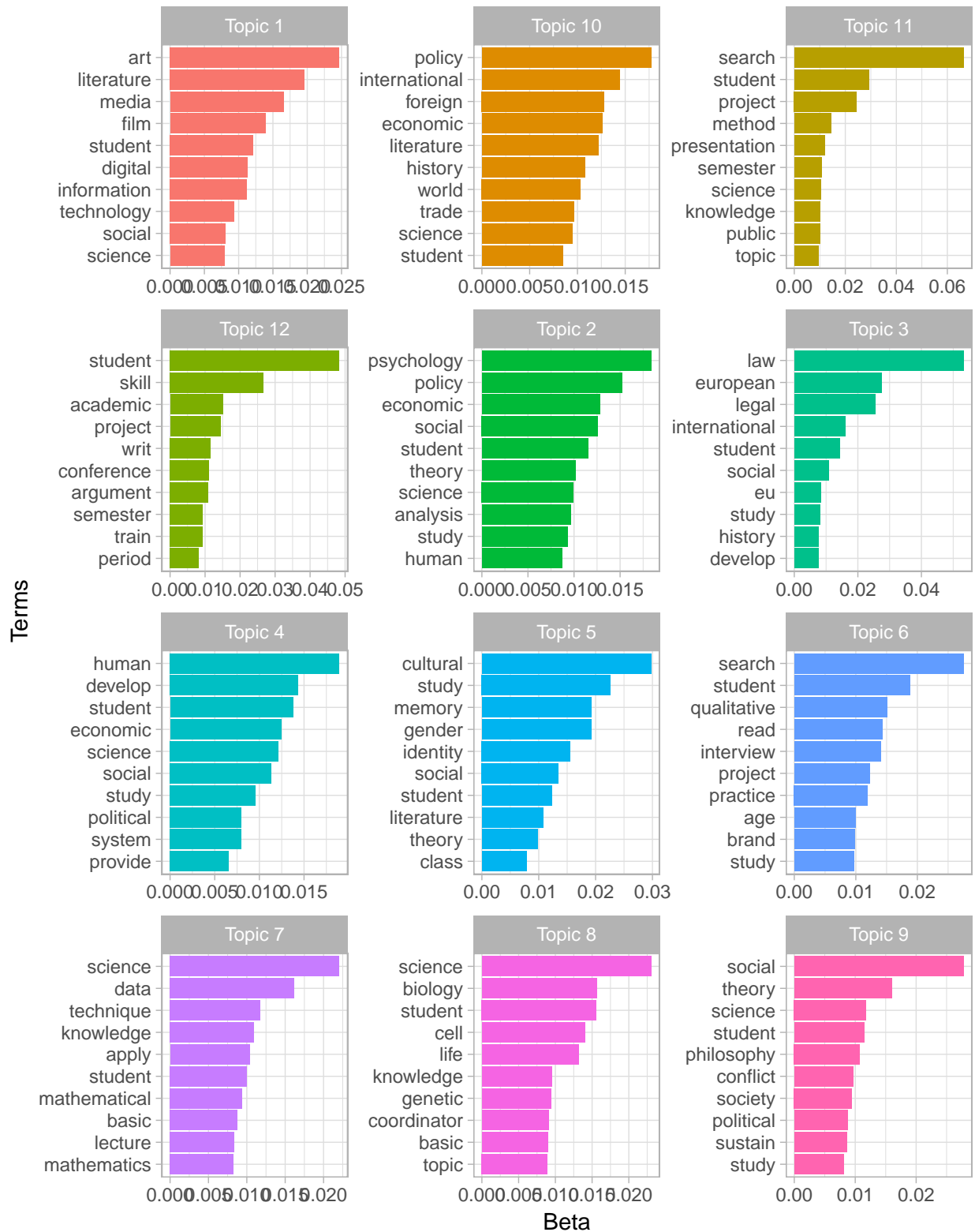
So far, we have only analyzed models with unlabelled topics. In order to give more substance to our analysis, we assign labels to the topics. To accomplish this, we look at the first plot and identify the common theme(s) among the terms. Some topics are easier to label than others. **topic 6** is for instance pretty straightforward: it is dominated by the term **search** and also contain the terms **qualitative**, **read**, **interview** and **study**: **topic 6** corresponds to qualitative research skills and we therefore label it **Qual. Res.**⁷. Labelling **topic 7** on the other hand is more trick. The best label I could find is **Engineering**. I have labelled each topic and present the results in the last three plots.

The labels give us a better image of the actual content of the courses and the clusters. The last plot indicate sthta most clusters cover the topics that we expect them to cover, indicating that the current division of courses in clusters is backed up by the content of the courses. As expected, the cluster **Sociology** covers the topics of **Society** and **Culture** and the cluster **Cultural Studies** covers the topics of **Arts**, **Culture** and **Qual. Res.**. Yet, I am surprised by the absence of certain topics in some clusters. For instance, the cluster **history** lacks the topics of **culture** and **society** and the topic of **research** is also barely present in **Biomedical Sciences**. The former shows the heavy focus of the **history** cluster on war and conflicts (**Foreign Policy**) and the latter reflects the absence of research projects in the classes of the biomedical cluster.

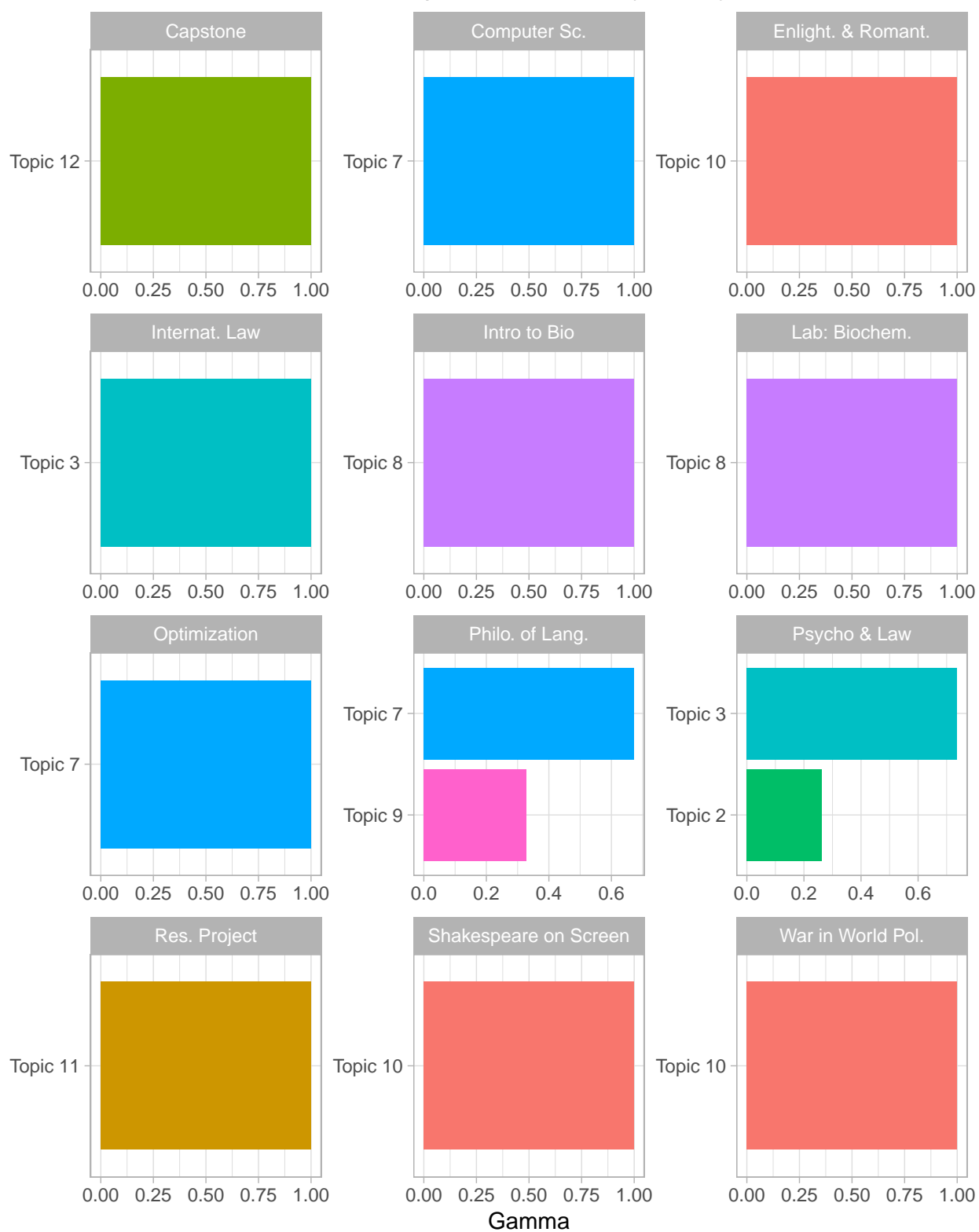
⁷This is a subjective choice and the reader may find a more fitting label.

3.4.2.1 Unlabelled Topics

Main Terms of each Topic (k = 12)



Main Topics of Courses (k = 12)



Main Clusters of each Topic (k = 12)



Main Topics of Clusters (k = 12)



3.4.2.2 Labelled Topics

Main Terms of each Topic (k = 12)



Main Topics of Courses (k = 12)



Main Topics of Clusters (k = 12)

