Digital humanities

Cvičení: analýza textů

Jindřich Marek

Práce s textem z digitální knihovny

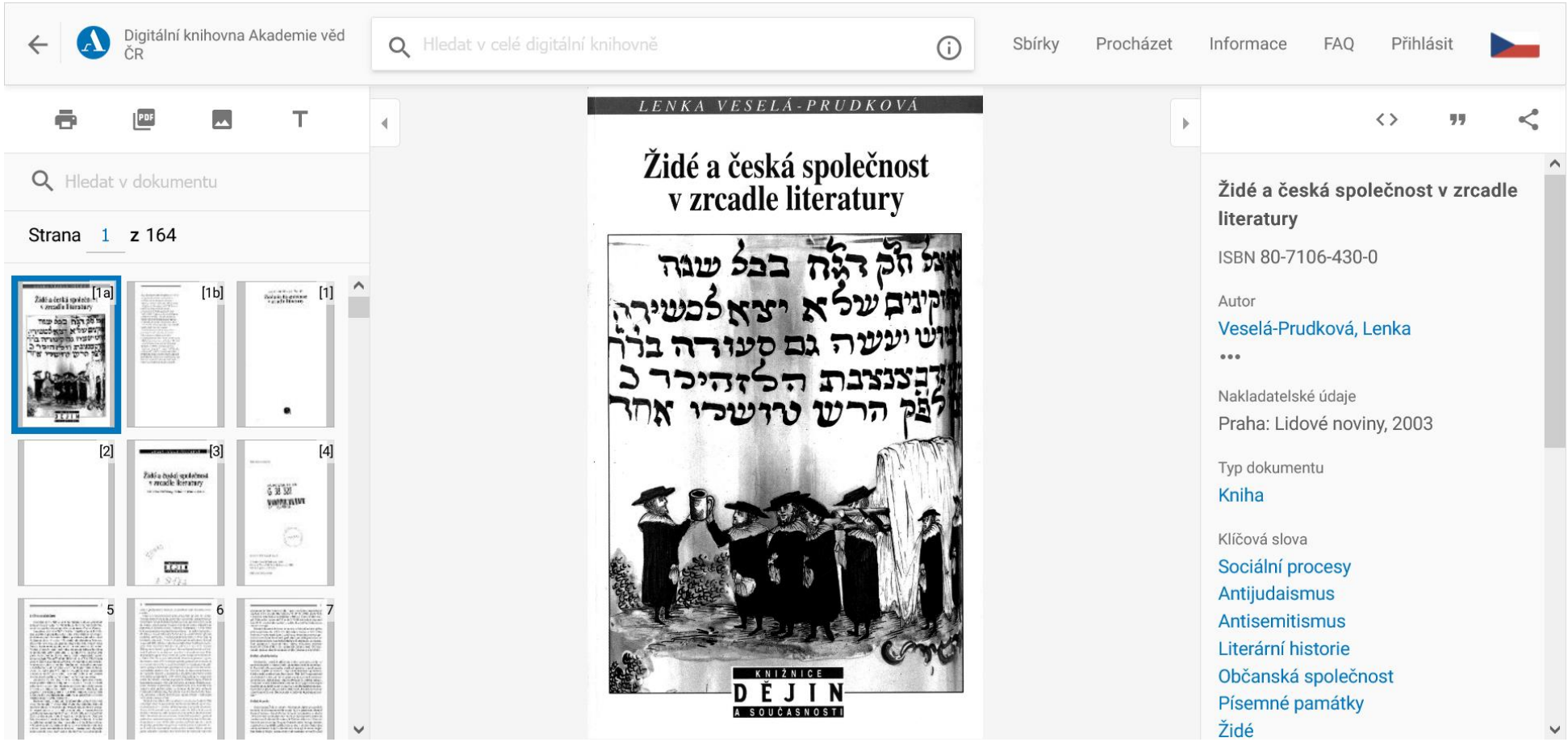
- vyhledání knihy v digitální knihovně KNAV
- stažení metadat
- z metadat přes API extrakce textu
- lemmatizace textu
- vizualizace textu
  - wordcloud

Knihy

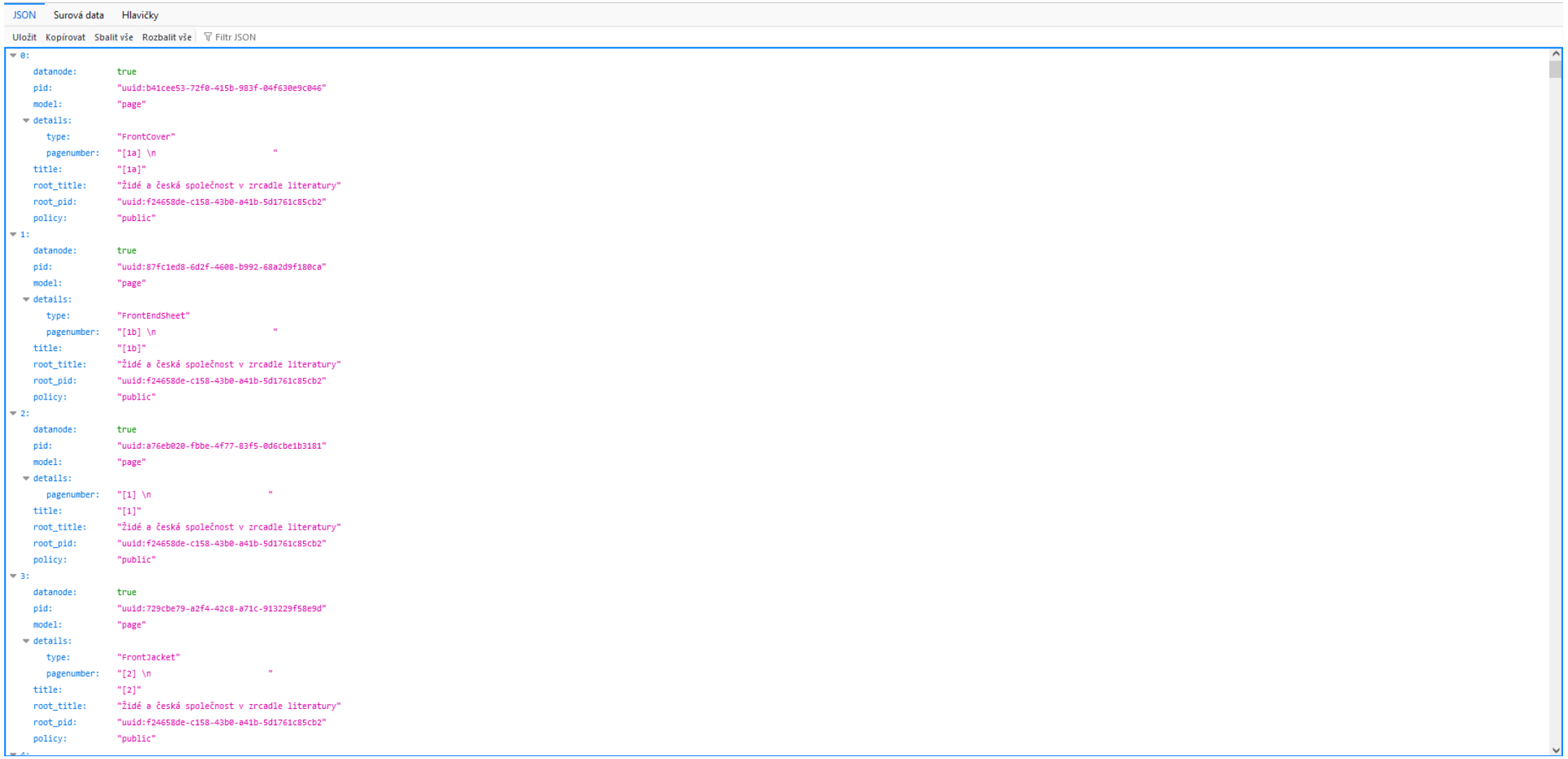

LENKA VESELÁ-PRUDKOVÁ
Židé a česká společnost
v zrcadle literatury
KNIŽNICE
DĚJIN
A SOUČASNOSTI


KAREL WEINFURTER
MAGIE V LÁSCE
NAKLADATELSTVÍ ZMATLÍK A PALIČKA, PRAHA-LETNÁ

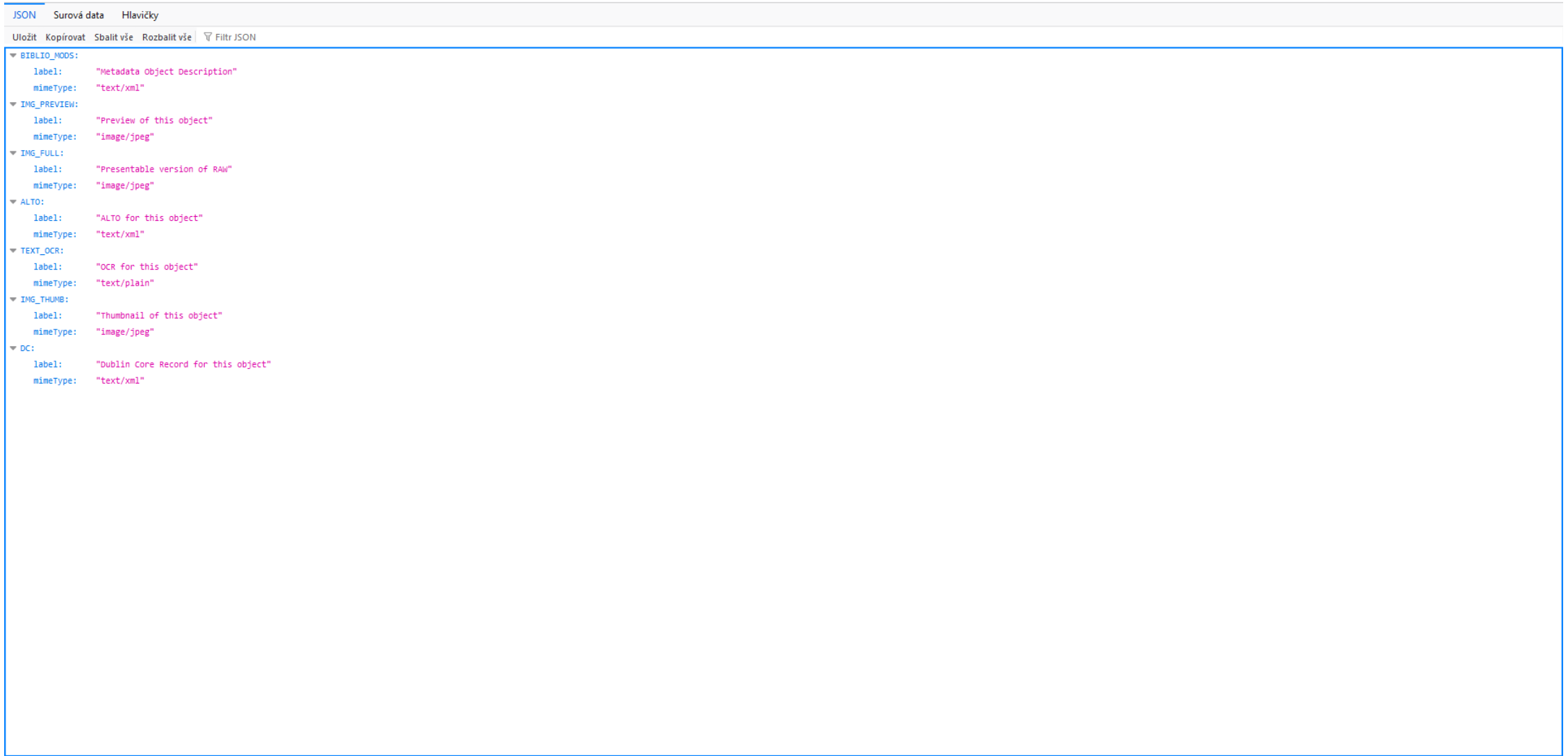
https://kramerius.lib.cas.cz/view/uuid:f24658de-c158-43b0-a41b-5d1761c85cb2?page=uuid:b41cee53-72f0-415b-983f-04f630e9c046



https://kramerius.lib.cas.cz/search/api/v5.0/item/uuid:f24658de-c158-43b0-a41b-5d1761c85cb2/children



https://kramerius.lib.cas.cz/search/api/v5.0/item/uuid:b41cee53-72f0-415b-983f-04f630e9c046/streams



https://kramerius.lib.cas.cz/search/api/v5.0/item/uuid:b41cee53-72f0-415b-983f-04f630e9c046/streams/IMG_FULL

https://kramerius.lib.cas.cz/search/api/v5.0/item/uuid:b41cee53-72f0-415b-983f-04f630e9c046/streams/TEXT_OCR

LENKA VESELÁ-PRU OKOVÁ
Židé a česká společnost
v zrcadle literatury
A SOUČASNOSTI

stazeni_textu_Vesela_Zide.R (1. část)

```r
# Load the jsonlite and httr libraries for handling JSON data and making HTTP requests.
library(jsonlite)
library(httr)

# Define the URL for the Kramerius API.
kramerius5_api <- "https://kramerius.lib.cas.cz/search/api/v5.0/item/"

# Define the UUID of the title for which you want to retrieve OCR text.
UUID_titul <- "uuid:f24658de-c158-43b0-a41b-5d1761c85cb2"

# Construct the URL for retrieving children items of the specified title.
request <- paste0(kramerius5_api, UUID_titul, "/children")

# Send a GET request to the Kramerius API to retrieve the JSON response.
response <- GET(request)

# Parse the JSON response into R data structures.
parsed_json <- fromJSON(content(response, "text"))
```

stazeni_textu_Vesela_Zide.R (2. část)

```r
# Extract URLs for retrieving OCR text streams.
urls <- paste0(kramerius5_api, parsed_json$pid, "/streams/TEXT_OCR")

# Retrieve OCR text from each URL and store it in a list.
text_list <- lapply(urls, function(url) {
  response <- GET(url)
  content(response, "text", encoding = "UTF-8")
})

# Convert the list of OCR text into a character vector.
text_list <- as.character(text_list)

# Write the OCR text to a text file named "stazeny_text_Vesela_Zide.txt" in the current working directory.
writeLines(text_list, "stazeny_text_Vesela_Zide.txt")
```

LENKA VESELÁ-PRUDKOVÁ
Židé a česká společnost
v zrcadle literatury
Od středověku k počátkům emancipace
DĚJIN
|A SOUČASNOSTI

Věnováno rodičům
KNIHOVNA AV CR
G 38 322
05185/05
Knižnici řídí Zdeněk Hojda

ISBN 80-7106-430-0

5
1. Obraz a skutečnost
Ztvárnění Židů a židovství, které budeme sledovat v tiskařské
produkci českých zemí v průběhu 16. až 18. století, odráželo i his¬
torický vývoj židovského společenství na území Cech a Moravy.
Legendy o usazování Židů v českých zemích sahají až k mytic¬
kým počátkům přemyslovského státu, avšak kritická historiogra¬
fie klade nejstarší židovská sídliště v pražském podhradí a v okolí
Vyšehradu až do 11. století." Po zániku obou lokalit se Židé sou¬
středili do nové osady na pravém vltavském břehu (dnešní Staré
Město), která zůstala jejich duchovním centrem až do 20. století.
Vcelku příznivé životní podmínky pro rozvoj židovského města
se ale zhoršily v první polovině 13. století, kdy i u nás byli Židé
pod vlivem církevních koncilů omezováni v hospodářské i spole¬
čenské oblasti. Vláda Přemysla Otakara II. (1253-1278) přinesla
českým Židům písemné potvrzení jejich právního a společenské¬
ho postavení, které s menšími obměnami stvrzovali i panovníci
v následujících staletích. Přemyslovo privilegium (Statuta Judae-
orum) na jedné straně Židům zaručovalo ochranu panovníka
a značnou vnitřní samosprávu, na straně druhé se stali jedním
z královských regálů s příslušnými finančními povinnostmi.
Ani Židé v Českém království se nevyhnuli pronásledováním,
která postihla židovské obyvatele v západní Evropě ve druhé
polovině 14. století. Jejich intenzita a počet se sice nedaly srovnat
s vlnou protižidovských útoků v německých oblastech, ale
pogrom v pražském ghettu v roce 1389 a upálení desítky Židů
o několik let později nenechaly nikoho na pochybách o silném
vnitřním napětí v české společnosti.
Husitské hnutí, ostatně jako každé období oslabení královské
moci, znamenalo pro české Židy existenční nejistotu a faktické
zhoršení situace. V českých a moravských městech trvale vzrůsta¬
la nespokojenost a protižidovské nálady z hospodářských

wordcloud_Vesela_Zide.R – 1. část (lemmatizace)

```r
# Load necessary libraries for data manipulation and visualization.
library(readxl)
library(readr)
library(dplyr)
library(udpipe)
library(RColorBrewer)
library(wordcloud2)
library(htmlwidgets)

# Get the directory of the currently executing script.
this_dir <- dirname(parent.frame(2)$ofile)
setwd(this_dir)
```

```r
# Download and load the UDPipe model for Czech language.
udmodel <- udpipe_download_model(language = "czech")
udmodel <- udpipe_load_model(file = udmodel$file_model)

# Read the OCR text from the downloaded file.
file <- "stazeny_text_Vesela_Zide.txt"
y <- read_lines(file, skip = 0, n_max = -1L)

# Perform tokenization, tagging, and lemmatization using UDPipe.
x <- udpipe_annotate(udmodel, y)
lv <- as.data.frame(x, detailed = TRUE)

# Convert lemma column to lowercase.
lv$lemma <- tolower(lv$lemma)
```

wordcloud_Vesela_Zide.R – 2. část (vyčištění lemmat, wordcloud)

```r
# Read Czech stopwords from file.
stopwords <- readLines("stopwords_cz.txt")

# Define a function to remove stopwords from text.
remove_stopwords <- function(text, stopwords) {
  words <- unlist(strsplit(text, "\\s+"))
  filtered_words <- words[!tolower(words) %in% stopwords]
  cleaned_text <- paste(filtered_words, collapse = " ")
  return(cleaned_text)
}

# Apply the remove_stopwords function to remove stopwords.
lv$lemma <- sapply(lv$lemma, remove_stopwords, stopwords)

# Remove leading and trailing whitespaces.
lv$lemma <- trimws(lv$lemma)
```

```r
# Convert the lemma column to a factor.
lv$lemma <- as.factor(lv$lemma)

# Count the occurrences of each lemma and arrange in descending order.
lv_framed <- lv %>% count(lemma, sort = TRUE)

# Convert lemma column to character and filter out lemmas with less than 3 characters.
lv_framed$lemma <- as.character(lv_framed$lemma)
df <- lv_framed %>% filter(nchar(lemma) > 2)

# Create a word cloud using wordcloud2 package.
hw <- wordcloud2(data = df, size = 1.6, color = 'random-dark')

# Save the word cloud as an HTML file.
saveWidget(hw, "wordcloud_vysledek_Vesela_Zide.html", selfcontained = FALSE)
```

žid
židovský
český
rok
století
společnost
praha
pražský
země
dílo
autor
jan
rok
vydání
kniha
číslo
knihopis
tisk
doba
událost
dějiny
historický
křesťanský
hebrejský
literatura
protižidovský
překlad
křesťan
prostředí
kronika
text
první
především
jeden
město
domácí
podobný
starý
německý
zpráva
stát
tiskárna
veleslavín
hebrejština

LENKA VESELÁ-PRUDKOVÁ
Židé a česká společnost
v zrcadle literatury
KNIŽNICE
DĚJIN
A SOUČASNOSTI


KAREL WEINFURTER
MAGIE V LÁSCE
NAKLADATELSTVÍ ZMATLÍK A PALIČKA, PRAHA-LETNÁ

tematicke_modelovani.R – 1. část (načtení knihoven, stažení modelu)

```r
# Load required libraries
libraries <- c(
  "quanteda", "quanteda.textstats", "quanteda.textplots", "topicmodels", "readtext",
  "tidytext", "dplyr", "jsonlite", "udpipe", "stopwords", "readxl", "stringr",
  "LDAvis", "wordcloud", "ggplot2", "gridExtra", "RColorBrewer", "forcats", "readr"
)
invisible(lapply(libraries, library, character.only = TRUE))

# Set working directory
this_dir <- dirname(parent.frame(2)$ofile)
setwd(this_dir)

# Download and load UD model for Czech
udmodel <- udpipe_download_model(language = "czech")
udmodel <- udpipe_load_model(file = udmodel$file_model)
```

tematicke_modelovani.R – 2. část (funkce: pouze podstatná jména)

```r
# Function to process text and save nouns to a file
process_text <- function(file, output_file) {
  text <- read_lines(file, skip = 0, n_max = -1L)
  annotations <- udpipe_annotate(udmodel, text)
  nouns <- as.data.frame(annotations, detailed = TRUE) %>%
    filter(upos == "NOUN") %>%
    mutate(lemma = tolower(lemma)) %>%
    pull(lemma)
  writeLines(nouns, output_file)
}

# Process text files and extract nouns
process_text("stazeny_text_Vesela_Zide.txt", "stazeny_text_Vesela_Zide_nouns.txt")
process_text("stazeny_text_magie.txt", "stazeny_text_magie_nouns.txt")
```

tematicke_modelovani.R – 3. část (očištění a tokenizace)

```r
# Load stopwords
stopwords_cs <- stopwords::stopwords("cs", source = "stopwords-iso")
stopwords_cz <- c(stopwords_cs, readLines("stopwords_cz.txt"))

# Read text files into corpus
y <- readtext(c("stazeny_text_Vesela_Zide_nouns.txt", "stazeny_text_magie_nouns.txt"))
data_txt <- corpus(y)

# Tokenization and preprocessing
tokens <- tokens(data_txt) %>%
  tokens_remove(min_nchar = 3) %>%
  tokens_remove(pattern = c("*-*", "und", "der", "býti")) %>%
  tokens_remove(stopwords_cz)
```

tematicke_modelovani.R – 4. část (tematické modelování)

```r
# Create document-feature matrix
dfm <- dfm(tokens)
dfm_df <- as.data.frame(t(dfm))
```

```r
# Topic modeling
dtm <- convert(dfm, to = "topicmodels")
set.seed(1234)
topic_model <- LDA(dtm, method = "Gibbs", k = 4, control = list(alpha = 0.1))
```

tematicke_modelovani.R – 5. část (JSON a vizualizace)

```
# Compute JSON for visualization
dtm <- dtm[slam::row_sums(dtm) > 0,]
phi <- as.matrix(posterior(topic_model)$terms)
theta <- as.matrix(posterior(topic_model)$topics)
vocab <- colnames(phi)
doc.length <- slam::row_sums(dtm)
term.freq <- slam::col_sums(dtm)[match(vocab, colnames(dtm))]
json <- createJSON(phi = phi, theta = theta, vocab = vocab,
                   doc.length = doc.length, term.frequency =
term.freq,
                   reorder.topics = FALSE)
serVis(json)
```

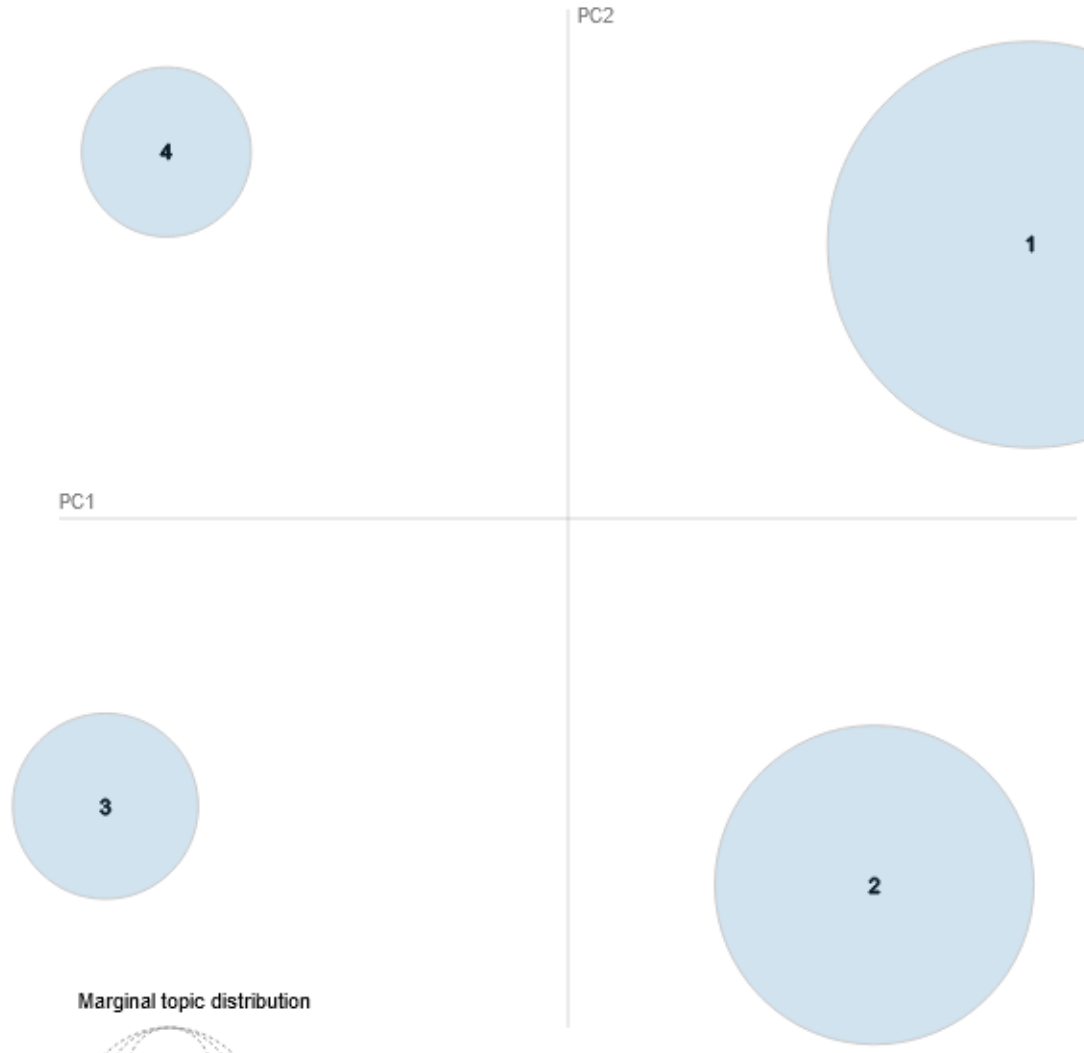
Selected Topic: 0 Previous Topic Next Topic Clear Topic

Slide to adjust relevance metric:(2)

λ = 1

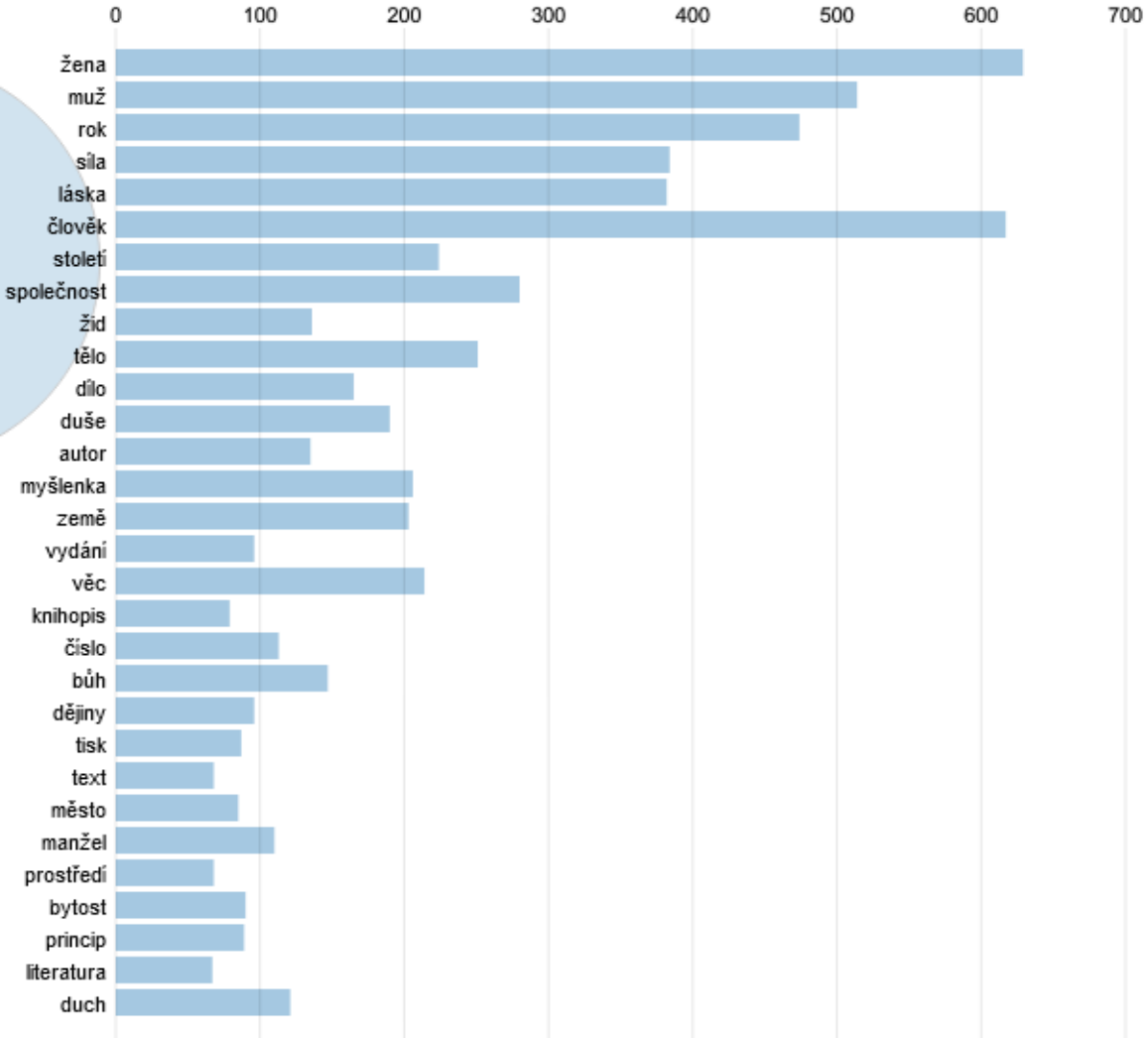
0.0 0.2 0.4 0.6 0.8 1.0

Intertopic Distance Map (via multidimensional scaling)



Marginal topic distribution

Top-30 Most Salient Terms1



Overall term frequency

Estimated term frequency within the selected topic

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

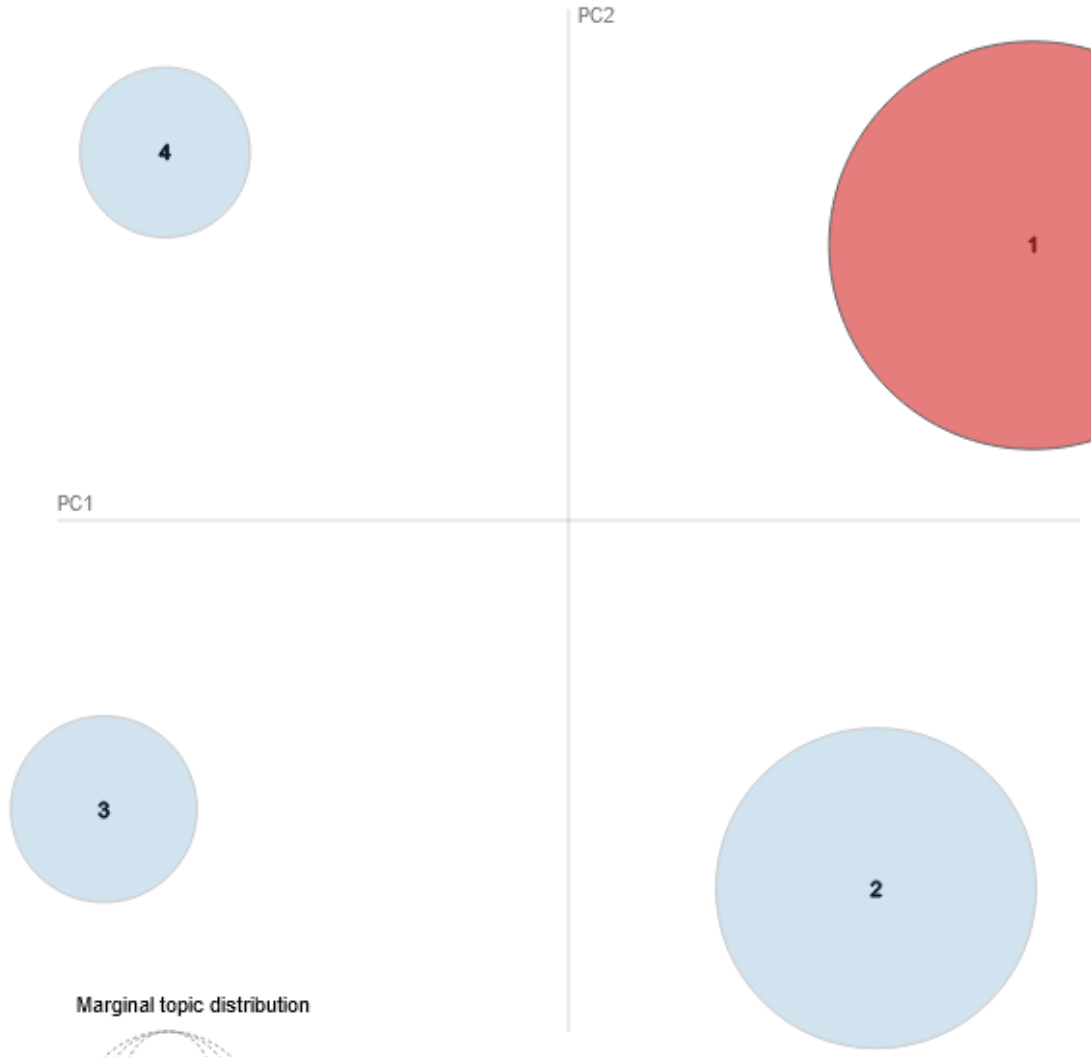
Selected Topic: 1 Previous Topic Next Topic Clear Topic

Slide to adjust relevance metric:(2)

λ = 1

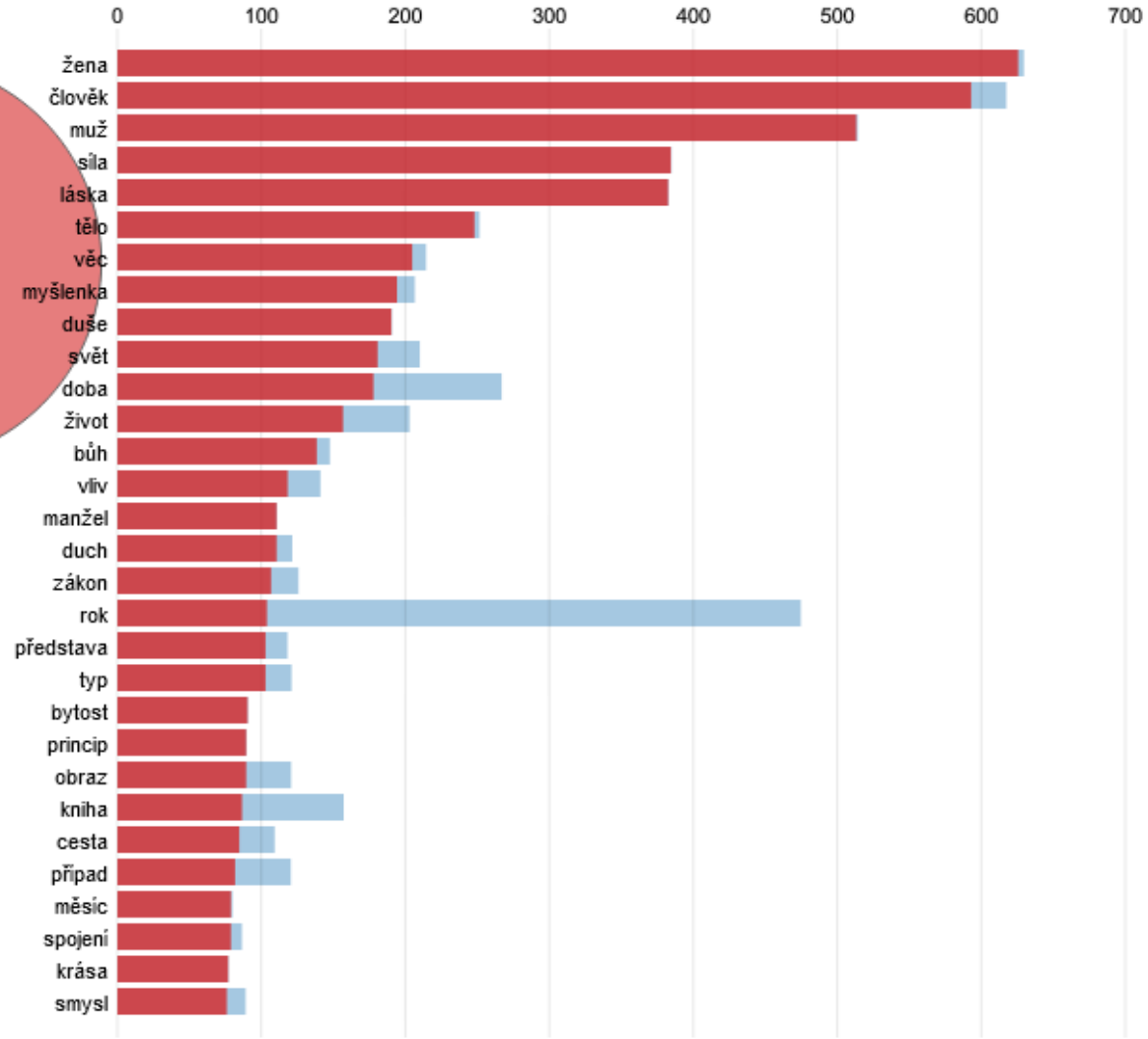
0.0 0.2 0.4 0.6 0.8 1.0

Intertopic Distance Map (via multidimensional scaling)



Marginal topic distribution

Top-30 Most Relevant Terms for Topic 1 (50% of tokens)



Overall term frequency

Estimated term frequency within the selected topic

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

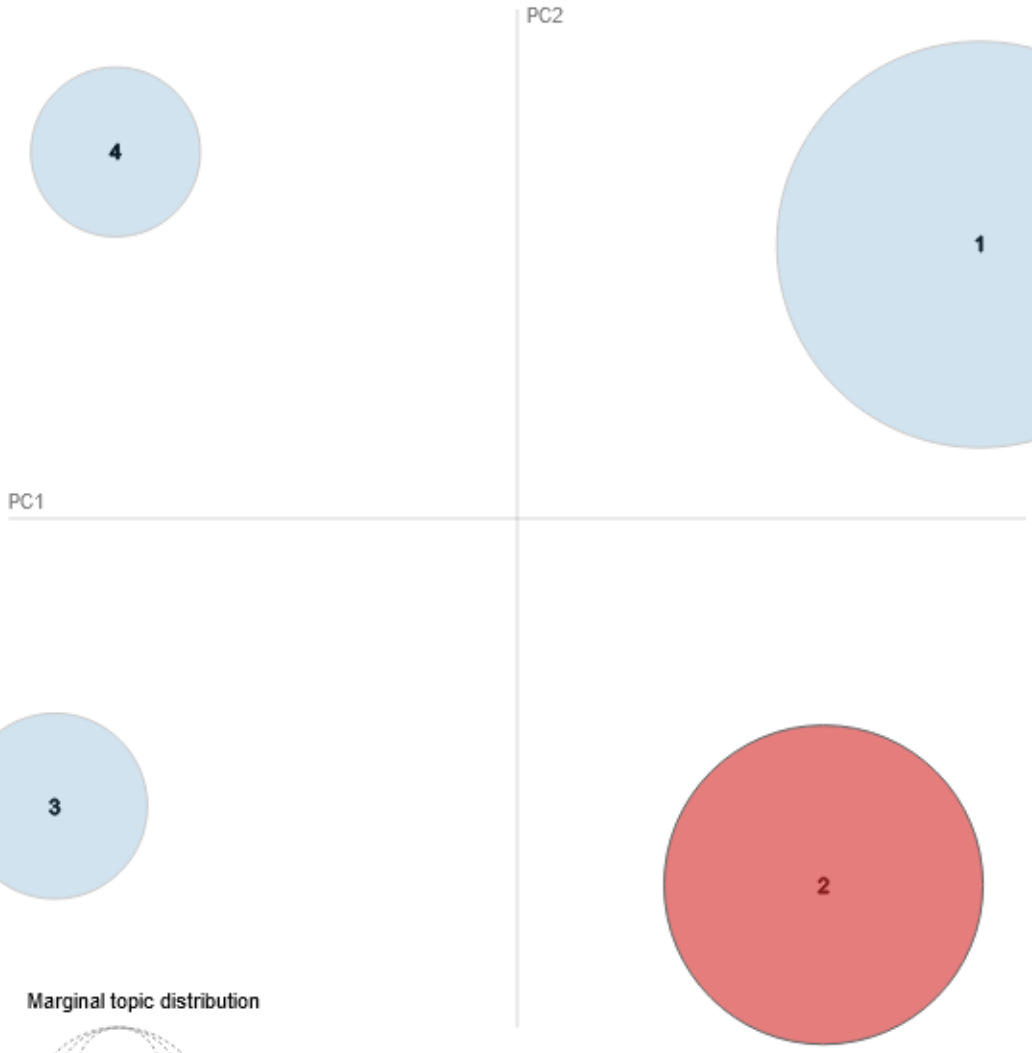
Selected Topic: 2 Previous Topic Next Topic Clear Topic

Slide to adjust relevance metric:(2)

λ = 1

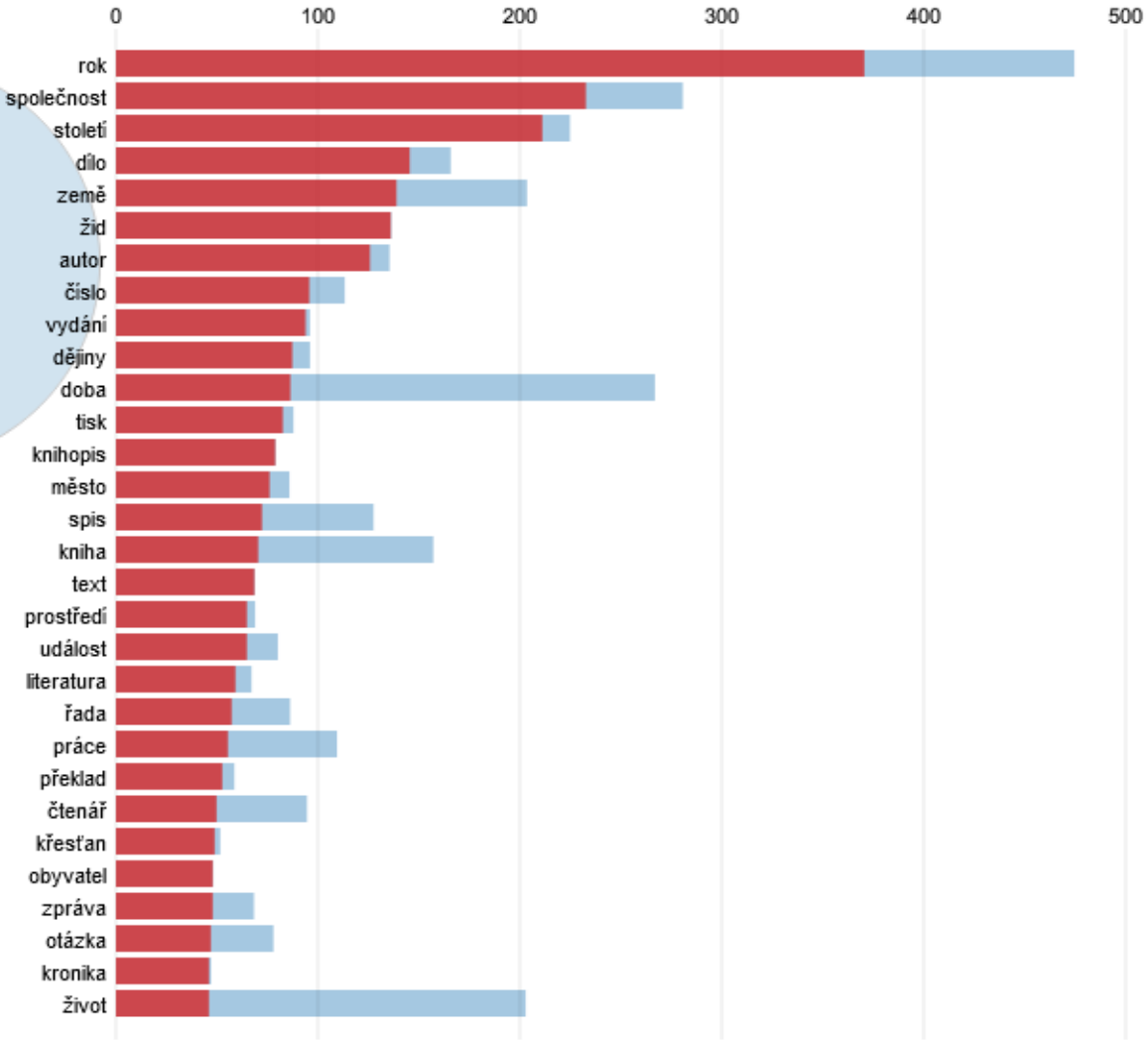
0.0 0.2 0.4 0.6 0.8 1.0

Intertopic Distance Map (via multidimensional scaling)



Marginal topic distribution

Top-30 Most Relevant Terms for Topic 2 (30.9% of tokens)



Overall term frequency

Estimated term frequency within the selected topic

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

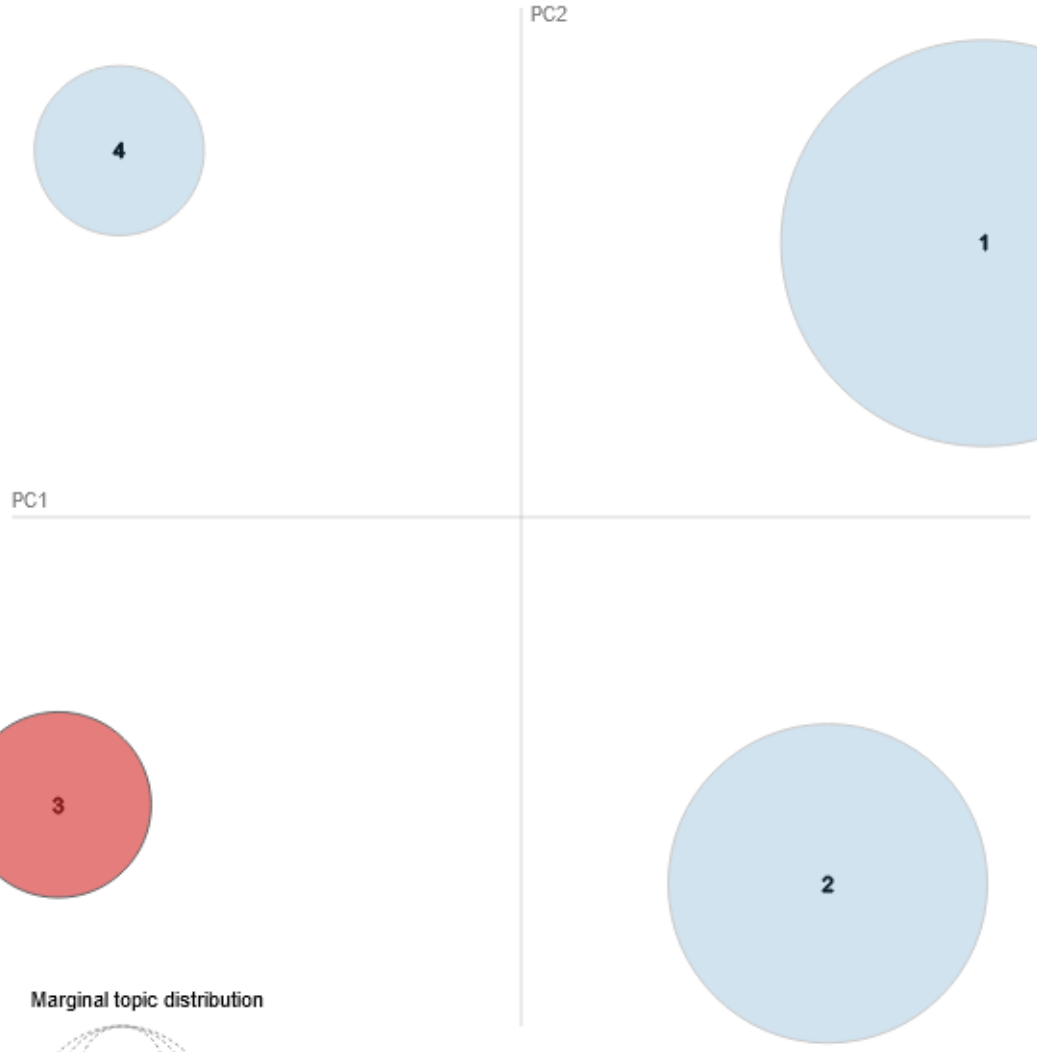
Selected Topic: 3 Previous Topic Next Topic Clear Topic

Slide to adjust relevance metric:(2)

λ = 1

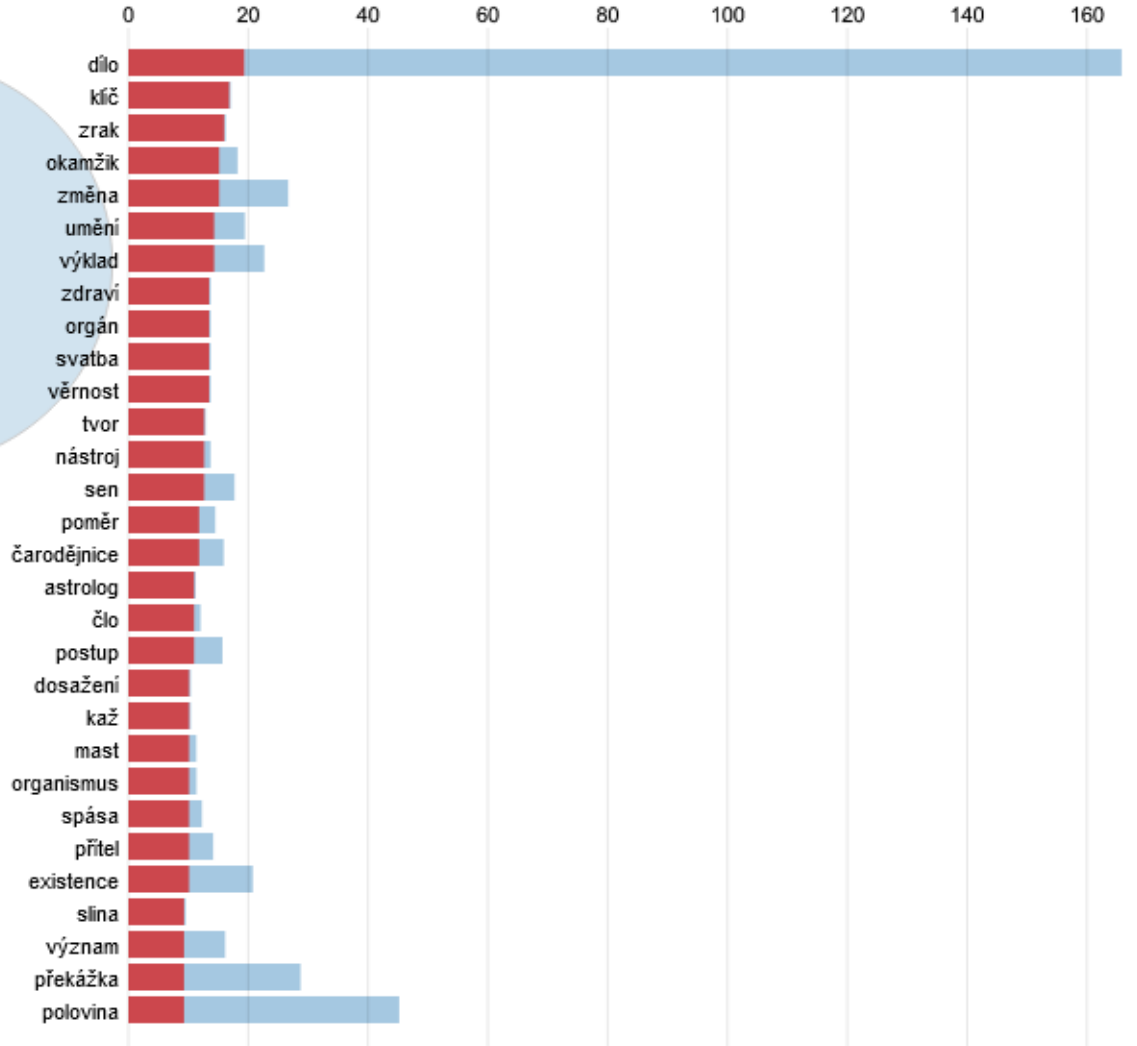
0.0 0.2 0.4 0.6 0.8 1.0

Intertopic Distance Map (via multidimensional scaling)



Marginal topic distribution

Top-30 Most Relevant Terms for Topic 3 (10.4% of tokens)



Overall term frequency

Estimated term frequency within the selected topic

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

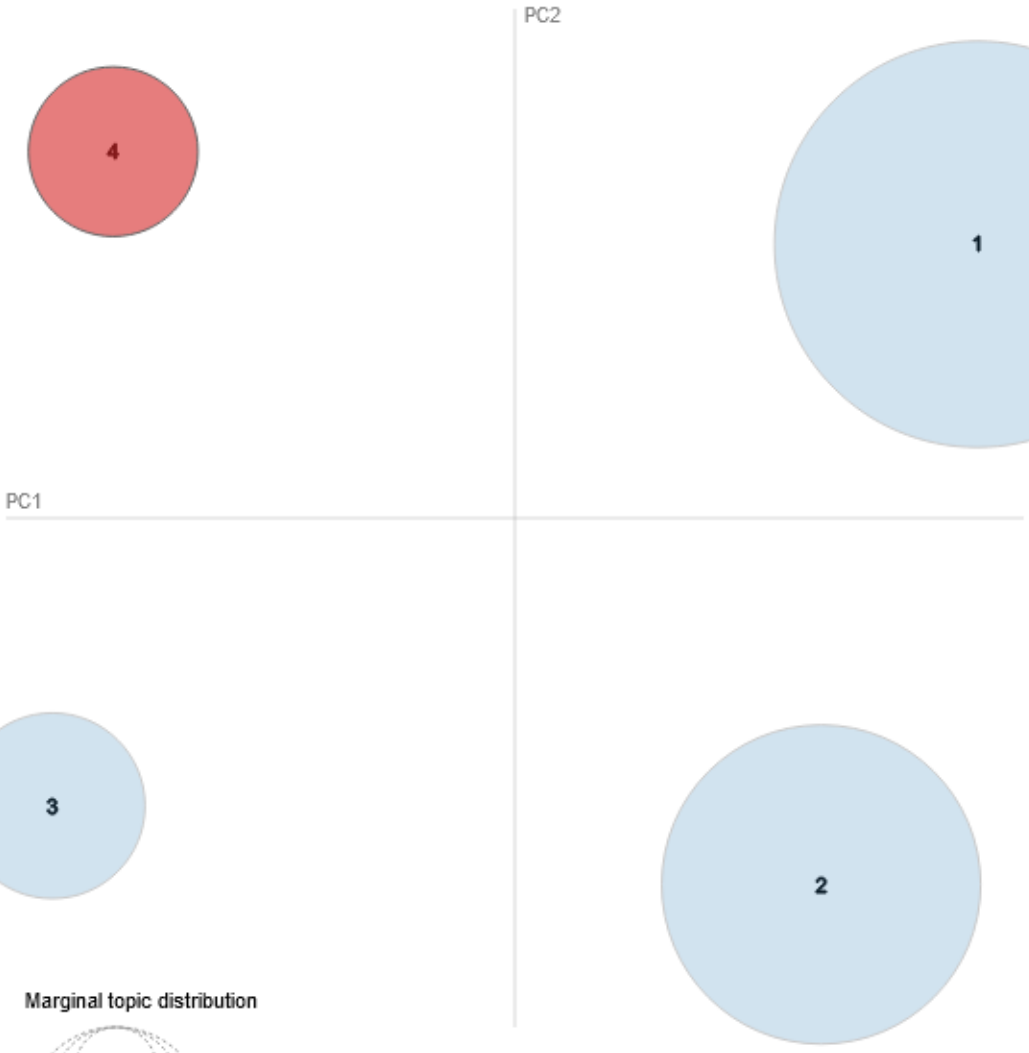
Selected Topic: 4 | Previous Topic | Next Topic | Clear Topic

Slide to adjust relevance metric:(2)

λ = 1

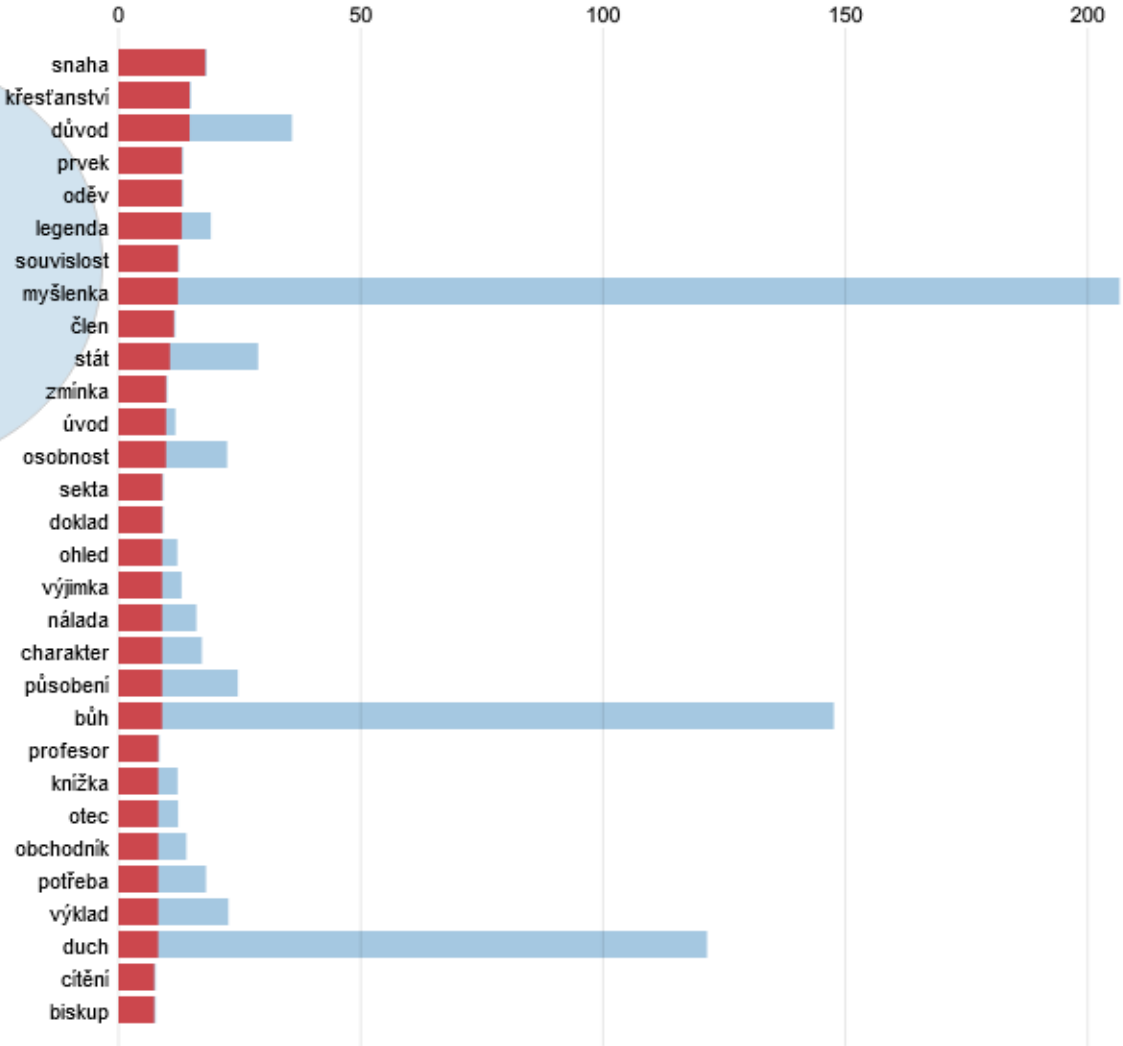
0.0 0.2 0.4 0.6 0.8 1.0

Intertopic Distance Map (via multidimensional scaling)



Marginal topic distribution

Top-30 Most Relevant Terms for Topic 4 (8.7% of tokens)



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)