Airbnb

Marcell P. Granát & Zója M. Szabó

March 6, 2021

Contents

Abstract

Here is the abstract.

Introduction

[leíró statok: eu bizottság felhívása]

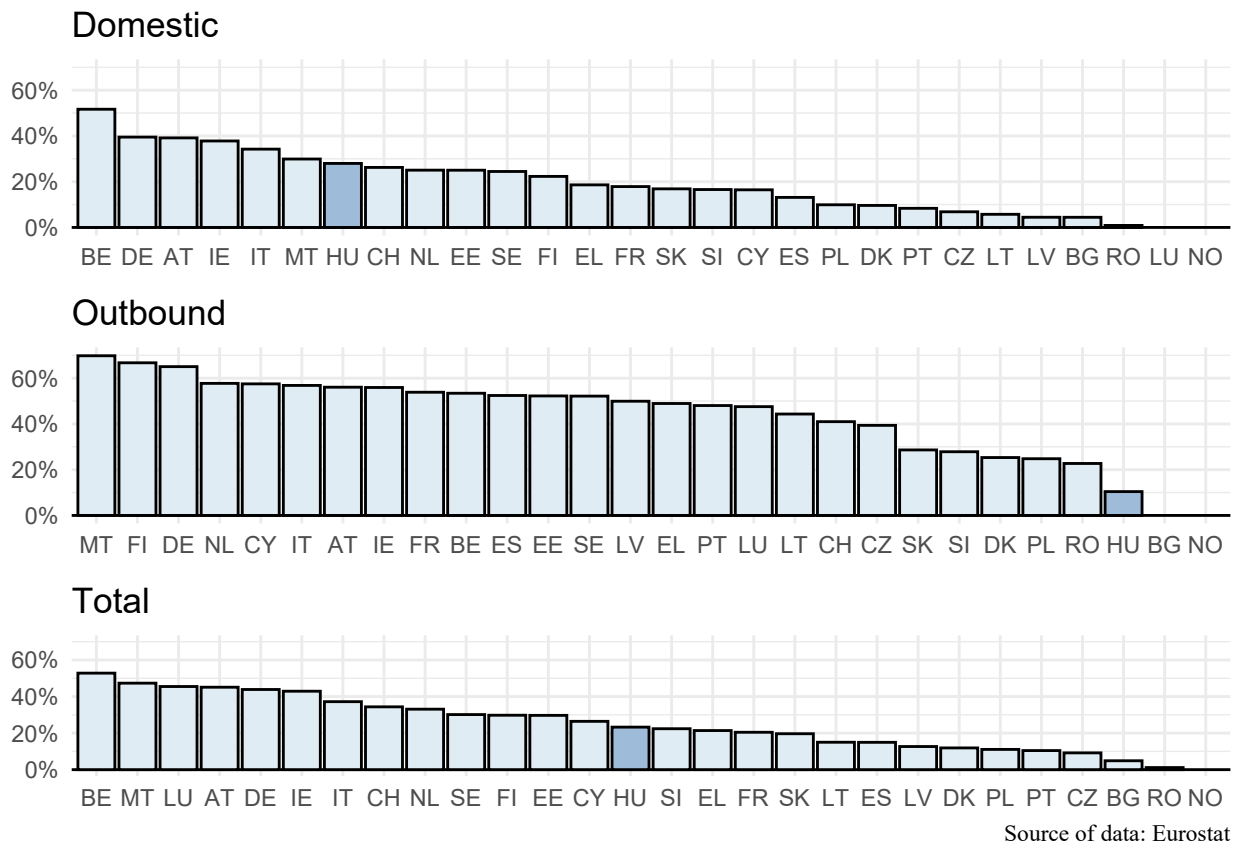

Figure 1: Proportion of internet bookings of the main means of accommodation by countries and the partner type

```
[1] "sf"         "data.frame"
```

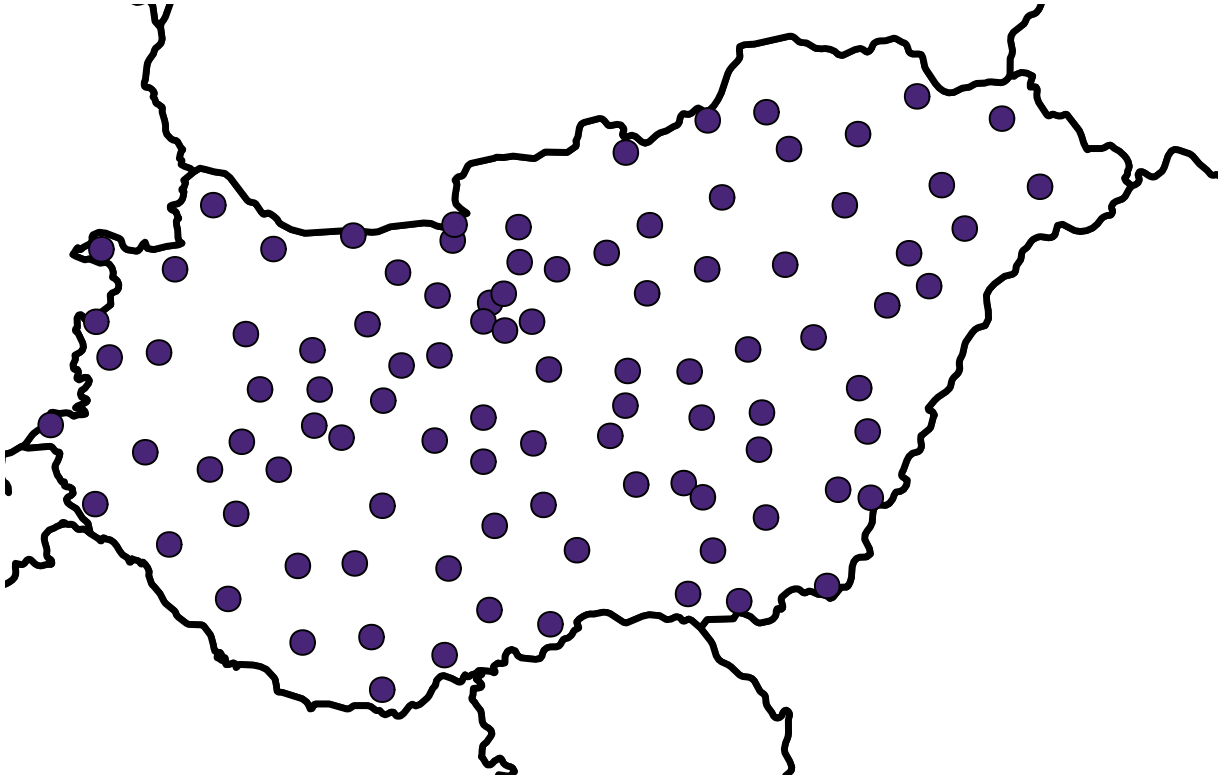

Figure 2: Starting points to our scrapping algorithm

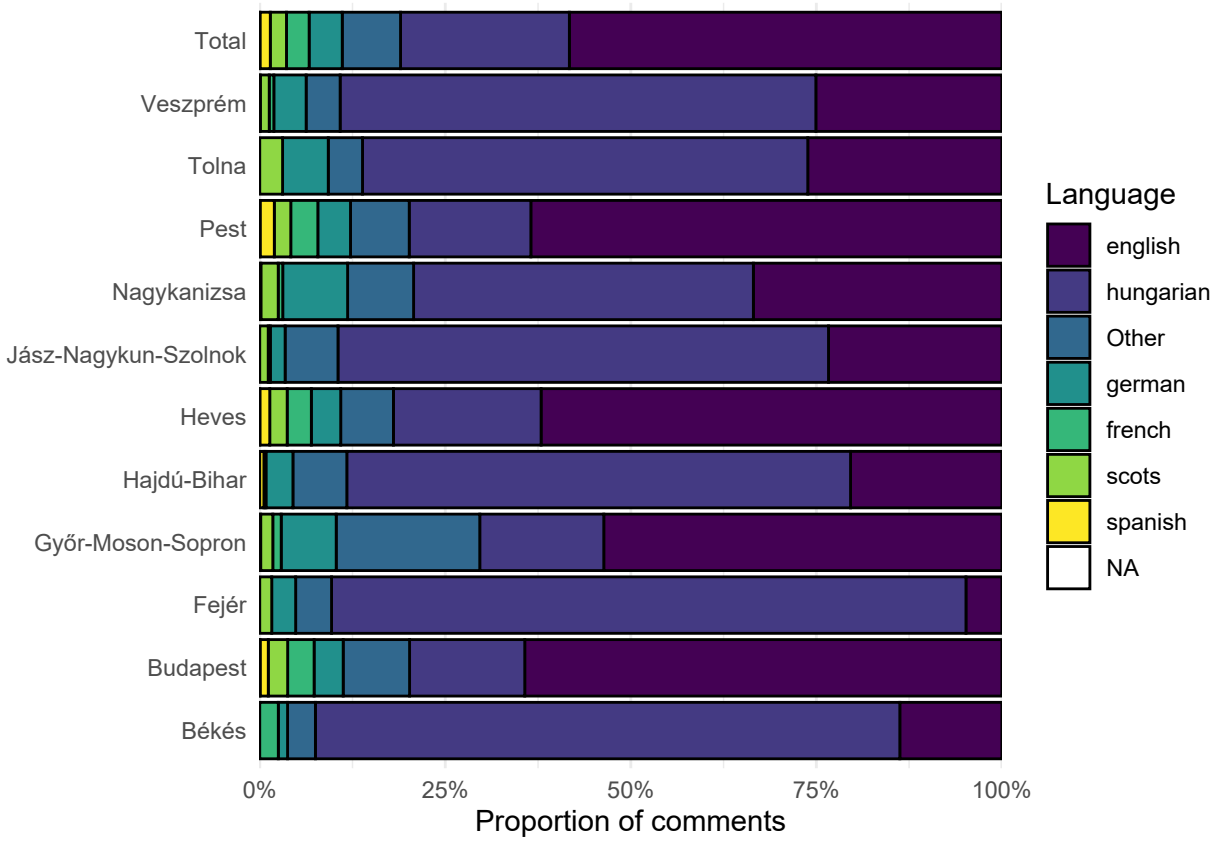

Figure 3: Most common languages found in the comments by counties

Figure 4: Words which correspond to high or low rating

Theoretical consideration

Data

Explore data

Model building

Consequences

Summary

References

Silge, J. & Robinson, D. (2017), *Text mining with R: A tidy approach*, O'Reilly Media, Sebastopol, CA.

Zervas, G., Proserpio, D. & Byers, J. W. (2017), 'The rise of the sharing economy: Estimating the impact of airbnb on the hotel industry', *Journal of Marketing Research* **54**(5), 687–705.

Appendix: R codes

```r
# Packages ------------------------------------------------------------------------------

library(tidyverse)
library(knitr)
library(tidytext)

load('dat.RData')
## Gg theme ===================================================================================

update_geom_defaults("point", list(fill = "#B1339E",
                                   shape = 21,
                                   color = "black",
                                   size = 1.4))
update_geom_defaults("line",
                     list(color = "midnightblue", size = 1.4))

update_geom_defaults("smooth", list(color = "red4", size = 1.4))

update_geom_defaults("density",
                     list(color = "midnightblue", fill =  "midnightblue",
                          alpha = .3, size = 1.4))

extrafont::loadfonts(device="win")

theme_set(theme_minimal() + theme(
  legend.direction = "vertical",
  plot.caption = element_text(family = "serif")
))

f.plot_eurostat <- function(x){
  eurostat::get_eurostat('tour_dem_ttorg', time_format = 'num') %>%
    filter(trip_arr %in% c('ACC_WEB', 'TOTAL') & duration == 'N1-3' & time == 2017 & purpose == 'TOTAL'
    pivot_wider(names_from = trip_arr, values_from = values) %>%
    mutate(
      value = ACC_WEB/TOTAL
    ) %>%
    filter(partner == x) %>%
    mutate(geo = fct_reorder(geo, -value)) %>%
    ggplot() +
    aes(geo, value, fill = geo == 'HU') +
    geom_hline(yintercept = 0) +
    geom_col(color = 'black') +
    scale_fill_brewer(palette = 3, guide = F) +
    scale_y_continuous(labels = scales::percent, limits = c(0, .7)) +
    labs(
      x = NULL, y = NULL, title = case_when(
        x == 'DOM' ~ 'Domestic',
        x == 'OUT' ~ 'Outbound',
        T ~ 'Total'
      )
    )
}
```

```r

ggpubr::ggarrange(
  f.plot_eurostat('DOM'),
  f.plot_eurostat('OUT'),
  f.plot_eurostat('WORLD') +
    labs(caption = 'Source of data: Eurostat'),
  ncol = 1
)

hun_cities <- read_csv("worldcities.csv") %>%
  filter(country == "Hungary") %>%
  select(city, lat, lng, admin_name, population)

cities <- readxl::read_excel("cities.xlsx")

library("rnaturalearth")
library("rnaturalearthdata")

world <- ne_countries(scale = "large", returnclass = "sf")
class(world)

merge(cities, hun_cities, by = 'city') %>%
  tibble() %>%
  ggplot() +
  geom_sf(data = world, size = 1.2, fill = 'white', color = 'black') +
  coord_sf(xlim = c(16, 23.4), ylim = c(45.5, 48.7), expand = FALSE) +
  geom_point(aes(x = lng, y = lat), size = 4,
             shape = 21, fill = viridis::viridis(1, begin = .1)) +
  theme(
    text = element_blank()
  )

lapply(dat, function(x) {
  tibble(city = x[["source"]][["city"]], comments = x$comments) %>%
    mutate(language = textcat::textcat(comments))
}) %>%
  reduce(rbind) %>%
  mutate(
    language = fct_lump(language, n = 6) %>%
      fct_infreq()
  ) %>%
  merge(hun_cities) %>%
  {rbind(., mutate(., admin_name = 'Total'))} %>%
  mutate(admin_name = fct_reorder(admin_name, admin_name == 'Total')) %>%
  ggplot() +
  aes(y = admin_name, fill = language) +
  scale_x_continuous(labels = scales::percent, limits = c(0,1), expand = c(0,0)) +
  geom_bar(color = "black", position = position_fill()) +
  scale_fill_viridis_d() +
  labs(x = 'Proportion of comments', y = NULL, fill = "Language")

dat_words <- lapply(dat, function(x) {
  tryCatch({
    tibble(comments = x$comments) %>%
```

```r
      mutate(language = textcat::textcat(comments)) %>%
      filter(language == "english") %>%
      tail(-2) %>%
      tidytext::unnest_tokens(words, comments, to_lower = T) %>%
      mutate(assesment = x[["source"]][["assesment"]], n_reviews = x[["source"]][["n_reviews"]])
  }, error = function(e) NULL)
}
) %>%
  {reduce(Filter(f = Negate(is.null), .), rbind)}

dat_words <- dat_words %>%
  filter(n_reviews > 10 & words != 'bővebben') %>%
  mutate(type = case_when(
    assesment >= quantile(assesment, .99) ~ 'High',
    assesment <= quantile(assesment, .01) ~ 'Low',
    T ~ 'Middle'
  ))

dat_words_High <- dat_words %>%
  filter(type == 'High') %>%
  group_by(words) %>%
  summarise(n = n()) %>%
  ungroup() %>%
  merge(dat_words %>%
          group_by(words) %>%
          summarise(total = n()) %>%
          ungroup()) %>%
  filter(n >= 10) %>%
  mutate(
    rtf = n/total, type = 'High'
  ) %>%
  arrange(desc(rtf)) %>%
  anti_join(data.frame(words = c(stopwords::stopwords(), "also", "can"))) %>%
  head(50)

dat_words_Low <- dat_words %>%
  filter(type == 'Low') %>%
  group_by(words) %>%
  summarise(n = n()) %>%
  ungroup() %>%
  merge(dat_words %>%
          group_by(words) %>%
          summarise(total = n()) %>%
          ungroup()) %>%
  filter(n >= 10) %>%
  mutate(
    rtf = n/total, type = 'Low'
  ) %>%
  arrange(desc(rtf)) %>%
  anti_join(data.frame(words = c(stopwords::stopwords(), "also", "can"))) %>%
  head(50)
rbind(dat_words_Low, dat_words_High) %>%
  reshape2::acast(words ~ type, value.var = "rtf", fill = 0) %>%
```

```r
wordcloud::comparison.cloud(colors = viridis::viridis(2, direction = -1 , end = .7),
                            max.words = 100)
```