Collection, Management, and Analysis of Twitter Data

Using the Twitter API for Academic Research and BERT

Andreas Küpfer
MZES & University of Darmstadt

kuepfer@pg.tu-darmstadt.de
@ankuepfer

Social Science Data Lab (MZES), Zoom

May 04, 2022

TECHNISCHE UNIVERSITÄT DARMSTADT

About The Richness of Social Media Data

Social Scientists in a Gold Rush

- Social media posts are full of potential (after filtering out bots) for data mining and analysis
- Recognizing this potential, platform providers increasingly restrict free access to such data
- Also Twitter acted accordingly until the beginning of 2021 with the release of their new Twitter API v2
- For academic purposes, the whole timeline of tweets can be crawled (conditional, more on this later)
- …hopefully this will continue after Elon Musk introduces *free speech* on Twitter

Is Twitter Research Only About Textual Data?

- Clearly no!
- Looking at social network interactions (follower, likes, ...) without any textual parts often reveals valuable information, e.g. about an ideological position or importance of a user within a social network
- Analyzing shared media elements (pictures, videos, ...) introduces another level of information

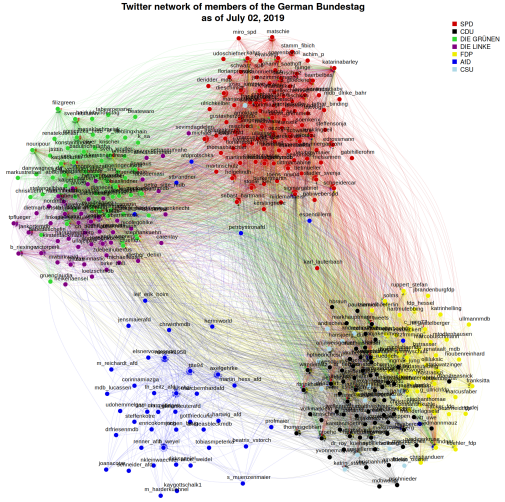


Fig. 1: Connections between German MPs (July 2019) (WZB Data Science Blog (Konrad, 2019))

Social Science Applications

- WHAT THE HASHTAG? (Small, 2010)
- Birds of the Same Feather Tweet Together. Bayesian Ideal Point Estimation Using Twitter Data (Barberá, 2015)
- Geospatial sentiment analysis using twitter data for UK-EU referendum (Agarwal et al., 2017)
- Politics, sentiments, and misinformation: An analysis of the Twitter discussion on the 2016 Austrian Presidential Elections (Kušen et al., 2018)
- Finding the bird's wings: Dimensions of factional conflict on Twitter (Sältzer, 2020)
- Presidential Twitter in the Face of COVID-19: Between Populism and Pop Politics (Manfredi-Sánchez et al., 2021)
- Does Twitter Affect Stock Market Decisions? Financial Sentiment Analysis During Pandemics: A Comparative Study of the H1N1 and the COVID-19 Periods (Valle-Cruz et al., 2021)
- Progress and push-back: How the killings of Ahmaud Arbery, Breonna Taylor, and George Floyd impacted public discourse on race and racism on Twitter (Nguyen et al., 2021)
- ...

Twitter API in Detail...

Twitter API v1.1 (until 12/2020) - Free Standard Version

- No full archive search (only past 7 days/3200 tweets)
- Smaller range of retrievable meta data
- Less query options to narrow down search
- Only premium API allowed for enhanced search

Twitter API v2 (since 01/2021)

Different Product Tracks

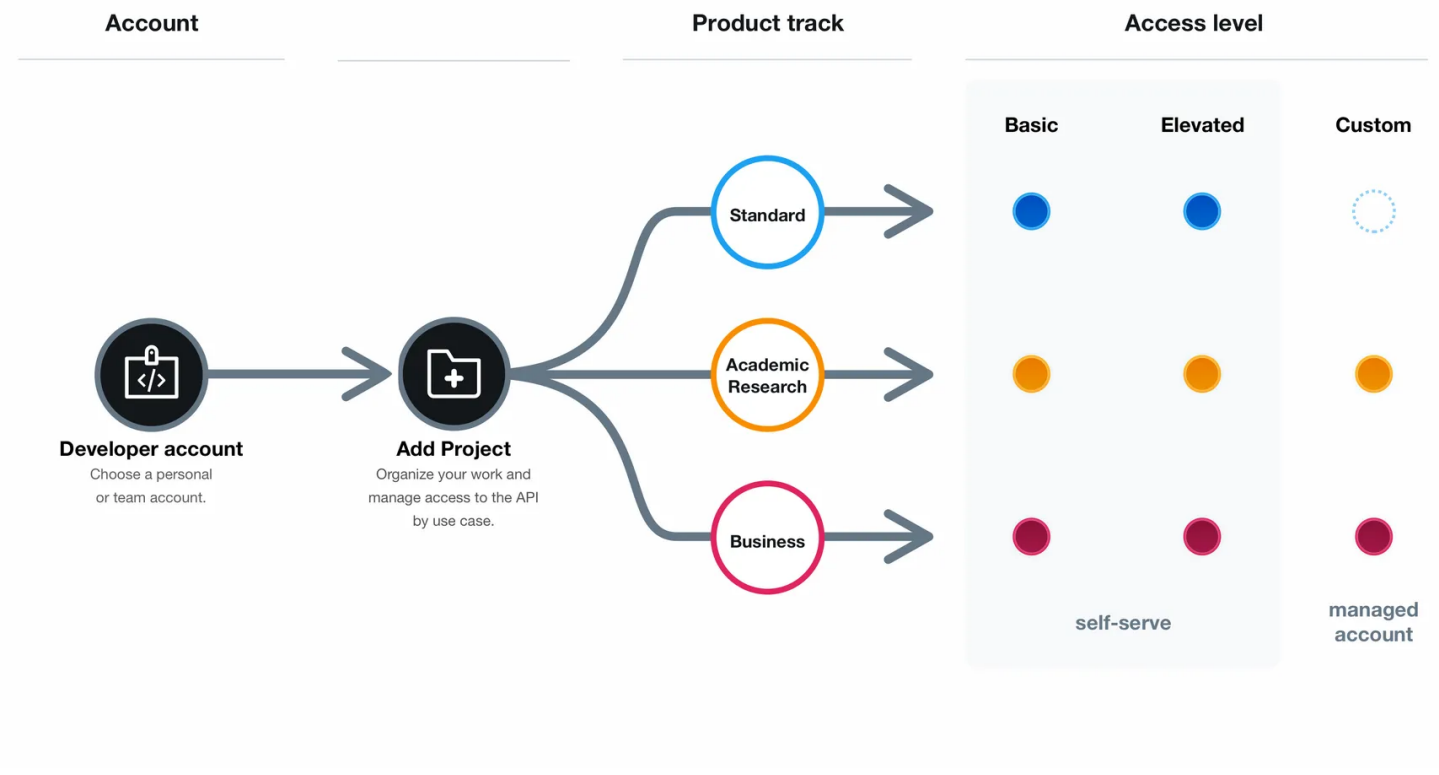


Fig. 2: New Twitter API Product Tracks (Twitter Developer Platform Blog, 2020)

Twitter API v2 (since 01/2021)

- Two main objects: tweets & users (including a lot of meta data)
- Media, spaces, lists, places, ...

- Tweet cap: 10 million tweets / month
- Query rules: 1024 characters
- Streaming rates: X requests / 15 minutes (depending on the endpoint)

- Cost: free
- Other limitations: removed tweets, deleted accounts, blocked accounts, ...

(Twitter Developer Platform)

Twitter API v2 (since 01/2021)

Queries

Really useful to pre-filter results: e.g.

> (putin OR selenskyj) -is:retweet lang:en has:geo

More on Queries

Twitter API v2 (since 01/2021)

Paging

Results are delivered in chunks of small size (pages) where you have to iterate through.

```
"meta": {
  "newest_id": "1204860593741553664",
  "oldest_id": "1204860580630278147",
  "next_token": "b26v89c19zqg8o3fobd8v73egzbdt3qao235oql",
  "result_count": 10
}
```

More on Paging

Twitter API v2 (since 01/2021)

Rate Limit

Maximum number of requests in a given time interval
e.g. full-archive search 300 requests/15 minutes/environment as well as a maximum of 1 request/1 second

More on Rate Limits

Academic Track: Getting Access & First Steps

Are you Eligible?

- You are either a master's student, doctoral candidate, post-doc, faculty, or research-focused employee at an academic institution or university.
- You have a clearly defined research objective, and you have specific plans for how you intend to use, analyze, and share Twitter data from your research.
- You will use this access for non-commercial purposes.

(Twitter Developer Platform product page)

Academic Track: Getting Access & First Steps

Before starting the process a Twitter account is required!

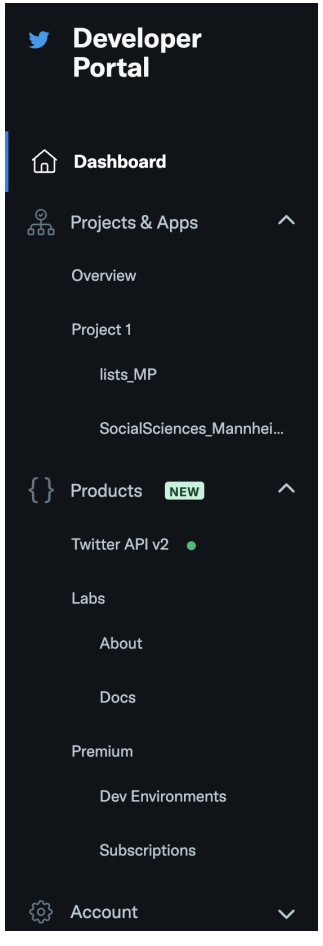
The Application Process

Fig. 3: Twitter Application Steps for Academic Track

- **Basic Info**: verify phone number, select country, ...
- **Academic Profile**: link to an official profile (department website or similar), role, ...
- **Project Details**: information about findings, description of the project itself and how the API should be used there (methodologies, how the outcomes will be shared, ...), ...
- **Review** & **Terms**: overview of the previous steps & developer agreement and policy
- After submission you receive a decision (usually) in a few days

Academic Track: Getting Access & First Steps

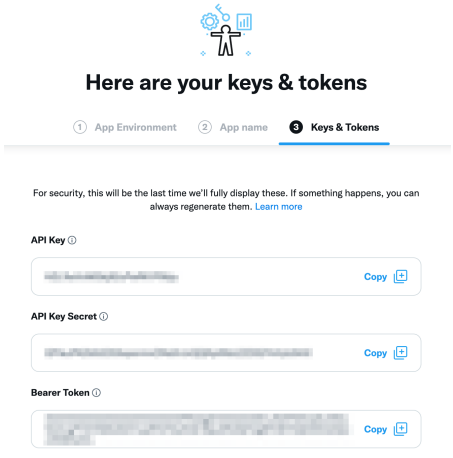
Developer Portal



- Manage projects and environments (which belong to a project)
- Generate API keys ("credentials" for API access)
- Overview of real-time monthly tweet cap usage
- Check available API endpoints and their specifics

Academic Track: Getting Access & First Steps

Creating a project, app and receiving an API key



- **Keep them private and do not push them to GitHub or similar!**
- API key ≈ username (also called consumer key)
- API key secret ≈ password (also called consumer secret)
- Bearer token ≈ special access token (also called authentication token)

Using the API: Packages in Python or R - and more

Postman as a Playground

- https://www.postman.com/
- Easy application to try out different queries, tokens, … and see the API results immediately - without any programming knowledge

- Tutorial to use the Twitter API with Postman: https://developer.twitter.com/en/docs/tutorials/postman-getting-started

Postman as a Playground

- Then... why don't just use Postman, retrieve the data and we're done with this part?
- There are several obstacles which can be tackled by a package/program code:
  - Building flexible queries (e.g. a list of users to retrieve tweets from)
  - Handling large responses which come split up during paging
  - Handle rate limit restrictions
  - Transforming responses into manageable data structures (e.g. dataframe and comma-separated values)

Packages which are ready for v2 (R & Python)

R

- https://github.com/cjbarrie/academictwitteR
- https://github.com/MaelKubli/RTwitterV2

Python

- https://github.com/twitterdev/search-tweets-python/tree/v2 (searchtweets-v2)
- https://github.com/tweepy/tweepy

Code Walkthrough to Crawl Tweets of German MPs

What is needed?

- Academic track API access
- Your personal API token
- A preferred programming language/development environment and a package: in our case *academictwitteR* (Barrie and Ho, 2021) in R

The upcoming code describes a very basic scenario, there is much more feasible with the API packages.

Code Walkthrough - Libraries & Authorization

R (academictwitteR)

```r
# install.packages("pacman")

## ---- Packages ----
pacman::p_load(
  dplyr,
  academictwitteR,
  quanteda,
  purrr
)

# set authorization for the API (Bearer Token)
set_bearer()
```

Code Walkthrough - Read Twitter IDs of German MPs

R (academictwitteR)

```
# load German MP Twitter IDs
german_mps ← read.csv("data/MP_de_twitter_uid.csv", colClasses=c("user_id"="character"))
```

Where to get user IDs from?

- Recommendation to use the Twitter user ID (e.g. *819914159915667456*) and not the user handle (e.g. *@StephPilsinger*) → handle can be changed by user
- In case you only have access to the handle, there is an v2 API endpoint to receive an user object from a handle: */2/users/by/username/:username*

- The Twitter Parliamentarian Database (van Vliet, 2020)
- Public Twitter lists (e.g. https://twitter.com/i/lists/912241909002833921) → use with caution
- legislatoR R Package (Göbel and Munzert, 2021)

Code Walkthrough - Retrieve Tweets

R (academictwitteR)

```r
# function to retrieve tweets in a specific time period of a single user
# (list of user IDs would be possible but one should keep
# the max. query string of 1024 characters in mind)
get_tweets_from_user ← function(user_id) {
  # Another option is to add "query" parameter
  get_all_tweets(
    users = user_id,
    start_tweets = "2021-01-01T00:00:00Z",
    end_tweets = "2021-09-30T00:00:00Z",
    data_path = "data/raw/",
    n = Inf)
}

# call function for each MP in the list
walk(german_mps[["user_id"]], get_tweets_from_user)
```

Code Walkthrough - Retrieve Tweets

R (academictwitteR)

```r
# in case of an interruption in between the data collection
# of a user resume the collection process:
resume_collection(data_path = "data/raw/")

# in case you want to update a set of crawled tweets
# update the collection until a specified end date:
update_collection(data_path = "data/raw/", end_tweets = "2022-04-30T00:00:00Z")
```

Code Walkthrough - Data Transformation & Storage

R (academictwitteR)

```r
# concatenate all retrieved tweets into one dataframe and select which columns to keep
# Another option: set parameter "user" to TRUE to retrieve user information
tweets_df ← bind_tweets(data_path = "data/raw/", output_format = "tidy") %>%
  select(tweet_id, text, author_id, user_username, created_at,
    sourcetweet_type, sourcetweet_text, lang)

write.csv(tweets_df, "data/raw/tweets_german_mp.csv", row.names = FALSE)
```

Afterwards: Pre-processing & Analysis

You're ready to move on! → usual steps applied to textual data (lowercasing, stopwords removal, stemming, ...) depending on the method at hand, e.g. in R:

```
library(quanteda)
tweet_corpus ← corpus(tweets_df[["text"]],
  docnames = tweets_df[["tweet_id"]]) # sets tweet_ids as document names to re-identify

# "2020 wurden in Berlin ca. 18.800 Miet- in Eigentumswohnungen umgewandelt. #Umwandlungsverbot"
dtm ← dfm(tweet_corpus %>% tokens(remove_punct = TRUE, remove_numbers = TRUE)) %>%
  dfm_tolower() %>% # removes capitialization
  dfm_remove(stopwords("german")) %>% # removes German stopwords
  dfm_wordstem(language = "german") # transforms words to their German wordstems
# "wurd berlin ca miet- eigentumswohn umgewandelt #umwandlungsverbot"
```

Recommended Resources (Methods Bites)

- Advancing Text Mining with R and quanteda
- Quantitative Analysis of Political Text

Reproducibility: Why this can be an Issue?

- "Academic researchers are permitted to distribute an unlimited number of Tweet IDs [...] on behalf of an academic institution [...]."
- **But**: Tweets can be deleted deleted, accounts suspended, ...
- However, there are also platforms like the one below, which tracks *public figures* and therefore justify the publication even of deleted tweets:

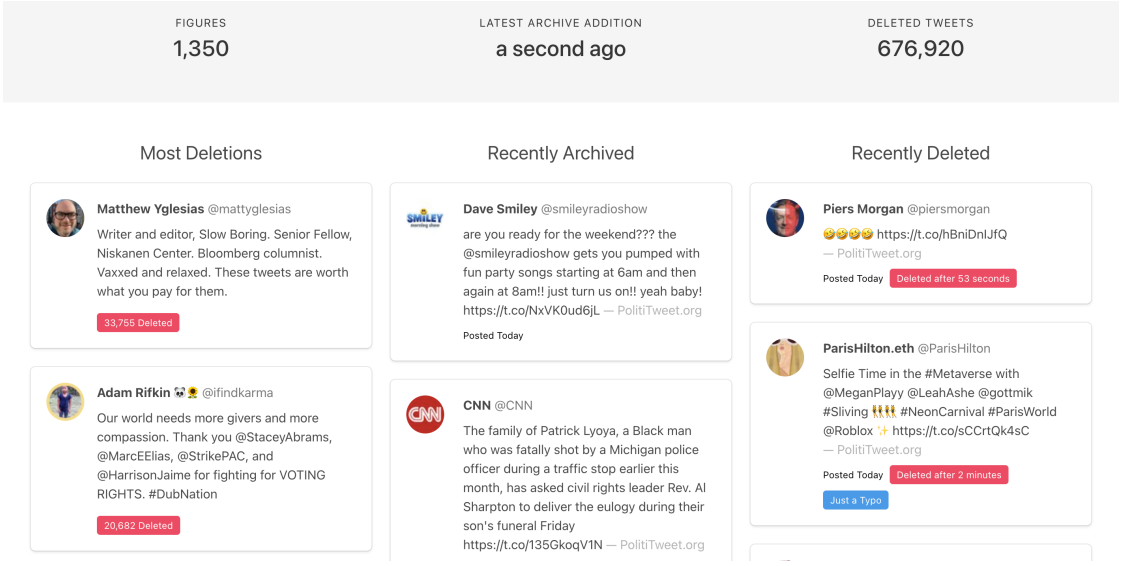


Fig. 4: Landing page of Politweet.org

Tweets in Action: An NLP Application based on BERT

Goal of the Research

- Estimation of legislator-level salience and position based on political texts
- Having a flexible, less data-hungry (than traditional methods) pipeline which is able to adapt to (sub-)domains of policies, different sources as well as languages

Application

- Estimating policy-specific legislator-level postion and salience scores based on textual contents of tweets using BERT and hierarchical shrinkage estimators
- Analyzing variation in legislator-level position and salience scores as a function of characteristics of legislators and electoral districts

What is the Data Basis and Annotation Approach?

- Tweets of all German MPs on Twitter published between the dates 01/01/2017 and 30/09/2021
- A random sample based on a pre-filtered portion of the original tweets (active learning and pre-classification / using a simple dictionary approach)
- **First**, annotated according to different domain-related policies (multi-class)
- **Second**, annotated with either -1 (left), 0 (neutral) or 1 (right) depending on the annotated policy

Fig. 5: Housing-related tweet of a German MP

Coding Scheme: A brief Overview...

Based on national manifestos of German parties

1. Curb speculation with real estate (left -1)
2. Facilitate acquisition of building land and private homes (right 1)
3. Incentivize construction (right 1)
4. Public and non-profit housing (left -1)
5. Rent control and tenant protection (left -1)
6. Expropriation (left -1)
7. Housing as a basic right (left -1)

Coding Scheme: A brief Overview...

Based on national manifestos of German parties

1. Curb speculation with real estate (left -1) $\rightarrow state$
2. Facilitate acquisition of building land and private homes (right 1) $\rightarrow private$
3. Incentivize construction (right 1) $\rightarrow private$
4. Public and non-profit housing (left -1) $\rightarrow state$
5. Rent control and tenant protection (left -1) $\rightarrow state$
6. Expropriation (left -1) $\rightarrow state$
7. Housing as a basic right (left -1) $\rightarrow state$

What is BERT?

- A so-called Transformer is able to capture contextual representations of each word in an input text (by using *attention*)
- BERT models are pre-trained (unsupervised) on huge corpora (Wikipedia, Twitter, ...), e.g. *cardiffnlp/twitter-xlm-roberta-base* for Twitter data
- They can be fine-tuned with a small amount of annotated data which makes them highly adaptable to a *problem* at hand

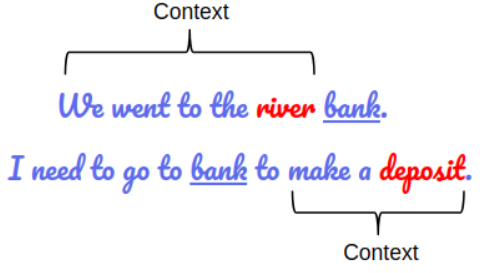


Fig. 6: BERT context-sensitivity example (Source)

More on BERT

Two-Step Approach: Salience & Position Estimation

- **First step**: fine-tune pre-trained BERT models with annotated tweets of each category and a random sample of tweets not belonging to the issue domain
- Predict policy categories for full corpus of tweets

Two-Step Approach: Salience & Position Estimation

- **First step**: fine-tune pre-trained BERT models with annotated tweets of each category and a random sample of tweets not belonging to the issue domain
- Predict policy categories for full corpus of tweets

- **Second step**: fine-tune pre-trained BERT models with annotated positional stances of each category
- Predict positions for each previously predicted policy-related tweet (determined by step one)

Why Hierarchical Models?

- We have estimates at the legislator level, but some of them are quite unreliable...

- Partial pooling via hierarchical models
- Random intercepts for legislators
- Shrinkage property: estimates for groups with less cases *borrow* more average information from the variance of other groups; estimates for groups with many cases rely mainly on their own variance

- Applied to the salience and position predictions of BERT

Why Hierarchical Models?

- Results used for substantial interpretation as well as adjustment of legislator estimates
- e.g. position model equation (*reference category for party is AfD*):

$$
\begin{aligned}
\text{Position}_i \sim & N\left(\mu, \sigma^2\right) \\
\mu = & \alpha_{j[i]} + \\
\alpha_j \sim & N(\gamma_0^\alpha + \\
& \gamma_{1_j}^\alpha \text{Housing Speaker}_{\text{TRUE}} + \\
& \gamma_{2_j}^\alpha \text{party}_{\text{CDU/CSU}} + \gamma_{3_j}^\alpha \text{party}_{\text{FDP}} + \gamma_{4_j}^\alpha \text{party}_{\text{Grüne}} + \\
& \gamma_{5_j}^\alpha \text{party}_{\text{Linke}} + \gamma_{6_j}^\alpha \text{party}_{\text{SPD}} + \\
& \gamma_{7_j}^\alpha \text{Median Rental Price} + \\
& \gamma_{8_j}^\alpha \text{House Ownership Rate} + \\
& \gamma_{9_j}^\alpha \text{House Ownership Rate} \times \text{Median Rental Price} \\
& , \sigma_{\alpha_j}^2), \text{ for Legislator j} = 1, \ldots, \text{J}
\end{aligned}
$$

(Preliminary) Results

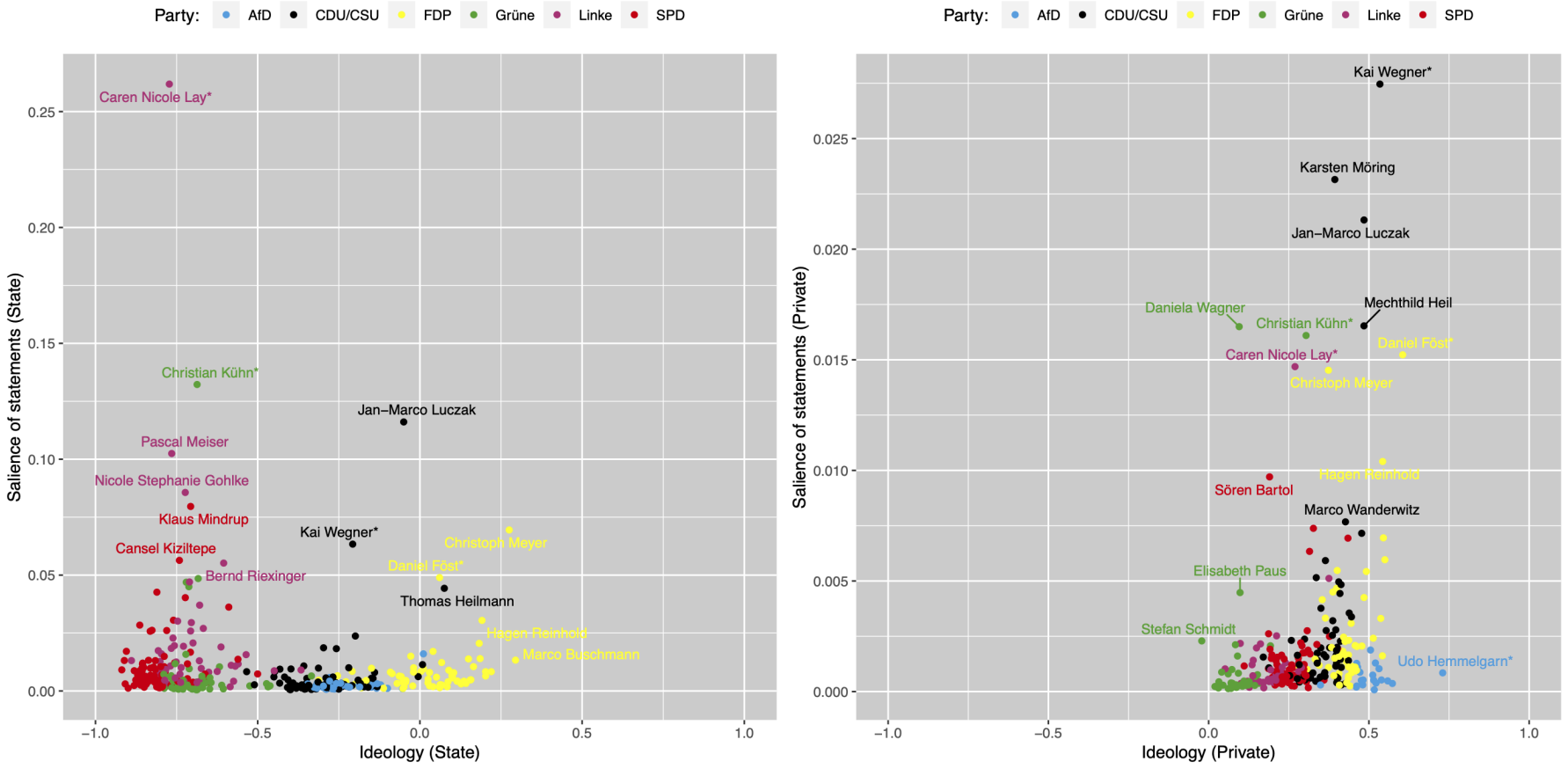


Fig. 7: Housing state and private salience and position per legislator

(Preliminary) Results

With increasing ownership rates, the importance of rental prices for salience decreases for both policy domains

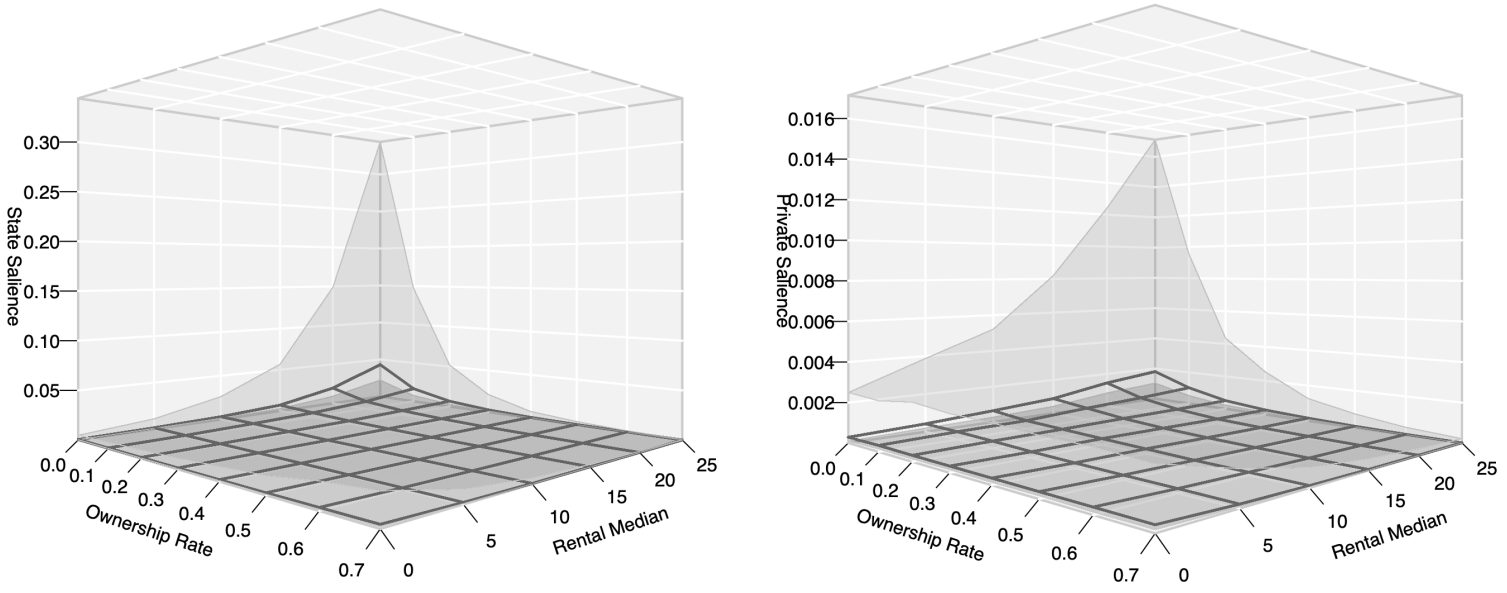


Fig. 8: Housing state and private salience (interaction of ownership rate and rental prices)

Validation & Outlook

- Initial validation with Wordfish positions on housing topics in national party manifestos shows similar order and pattern as party-aggregated BERT estimates

- Second annotation workflow with updated coding scheme
- Performance evaluation of BERT by iteratively decreasing the amount of data for fine-tuning
- Comparison against and with different combinations:
  - SVM (salience) + BERT (positions)
  - BERT (salience) + WordFish (positions)
  - Dictionary (salience) + BERT (positions)
  - ...

Questions?

Methods Bites Blog post containing additional resources & hints about the Twitter API v2 published soon!