Why learn topic modeling

TOPIC MODELING IN R

Pavel Oleinikov

Associate Director, Quantitative Analysis Center, Wesleyan University

What are topic models

- Topics give us a quick idea what a document is about.
- A topic is a label for a collection of words that often occur together. E.g., ***weather*** includes words: rain, storm, snow, winds, ice
- Topic modeling is the process of finding a collection of topics fitted to a set of documents

Rise of popularity

- Topic models are a way to get an idea what the documents are about, quickly.
- Because topics are quantified, it is possible to track topic prevalence, compute similarity (or distance) between topics, and use tools like linear regression.
- More technical applications involve use of topic models as classifiers, or as input to other tools like text segmentation.

Topic models - descriptive side

- Our course will focus on one specific implementation of topic modeling algorithms, called Latent Dirichlet Allocation (LDA)
- LDA takes a document-term matrix as its input
- LDA returns two matrices: one contains prevalence of topics in documents, the other - probability of words belonging to topics.

Illustration

- We have a tiny collection of documents.
- They refer to two topics: ***restaurants*** and ***loans***.
- A collection of documents is also called a **corpus**.

```
the_corpus =
c("Due to bad loans, the bank agreed to pay the fines",
"If you are late to pay off your loans to the bank, you will face fines",
"A new restaurant opened in downtown",
"There is a new restaurant that just opened on Warwick street",
"How will you pay off the loans you will need for the
restaurant you want opened?")
```

Illustration

This corpus is converted into a document-term matrix (dtm).

A dtm is a bag-of-words representation of text: the word order is lost.

```
     Terms
Docs  bank fines loans pay new opened restaurant
  d_1    1     1     1   1   0      0          0
  d_2    1     1     1   1   0      0          0
  d_3    0     0     0   0   1      1          1
  d_4    0     0     0   0   1      1          1
  d_5    0     0     1   1   0      1          1
```

Note: document 5 has both topics

The model is fitted by calling function `LDA`

```r
lda_mod <- LDA(x=d, k=2, method="Gibbs",control=list(alpha=1, delta=0.1, seed=10005, keep=1))
```

And the result is two tables, for terms and topics.

```
    bank   fines   loans     pay      new   opened restaurant
1 0.1963 0.1963 0.2897 0.2897 0.00935 0.00935    0.00935
2 0.0115 0.0115 0.0115 0.0115 0.24138 0.35632    0.35632
```

```
        1     2
d_1 0.833 0.167
d_2 0.833 0.167
d_3 0.200 0.800
d_4 0.200 0.800
d_5 0.667 0.333
```

Topic modeling - the other parts

- Matrices are not a good way to present the results. We need to use charts.
- There are choices which words to keep and which ones to exclude from a document-term matrix.
- Documents can be constructed in multiple way: they can be based on chapters in a novel, on paragraphs, or even on a sequence of several words.
- The LDA algorithm relies on control parameters which can impact the output.

Let's practice!

TOPIC MODELING IN R

Counting words

TOPIC MODELING IN R

Pavel Oleinikov

Associate Director, Quantitative Analysis Center, Wesleyan University

Splitting text

- The task of splitting text into words is also called 'tokenization'
- Package `tidytext` has function `unnest_tokens()` that does the splitting.

```
unnest_tokens(data, input=text, output=word,
  format="text", tokens="word", drop=TRUE, to_lower=TRUE)
```

- It returns a tidy table, with one word per row.

Example of using unnest_tokens

We have a data frame named `book`

```
book
```

```
  chapter                          text
1       1             It is what it is
2       2 What goes around comes around
```

We call `unnest_tokens` :

```
book %>%
  unnest_tokens(input=text, output=word,
                token="words", format="text",
                drop=T, to_lower=T)
```

And obtain a table

```
    chapter   word
1         1     it
1.1       1     is
1.2       1   what
1.3       1     it
1.4       1     is
2         2   what
2.1       2   goes
2.2       2 around
2.3       2  comes
2.4       2 around
```

Counting words

- We will use package `dplyr` to count word frequencies.
- Function `count()` groups rows by chapter and word and returns the word frequency.

```
book %>%
  unnest_tokens(input=text, output=word) %>%
  count(chapter, word)
```

- Each chapter-word pair now has its own row.

```
  chapter word       n
    <dbl> <chr>  <int>
1       1 is         2
2       1 it         2
3       1 what       1
4       2 around     2
5       2 comes      1
6       2 goes       1
7       2 what       1
```

Getting the top words 1

- We often are interested in the most frequent words.
- They can be extracted using `dplyr` functions
- To get the top-n words: group words by chapter, sort/arrange by count in descending order, keep rows whose number is less than `n`

Getting the top words 2

```
book %>%
  unnest_tokens(input=text,
    output=word) %>%
  count(chapter, word) %>%
  group_by(chapter) %>%
  arrange(desc(n)) %>%
  filter(row_number() < 3) %>%
  ungroup()
```

```
  chapter word       n
    <dbl> <chr>  <int>
1       1 is         2
2       1 it         2
3       2 around     2
4       2 comes      1
```

Casting counts into a document-term matrix

- Casting a table means transforming it into a different format
- A document-term matrix (dtm) contains counts of words.
- Each row corresponds to a document, each column - to a word.
- In our case, each chapter is its own document.
- Package `tidytext` has function `cast_dtm` to do this transformation.
- Just add `cast_dtm` after `count`

```
cast_dtm(data, document=chapter, term=word, value=n)
```

Example of using cast_dtm()

```
dtm <- book %>%
  unnest_tokens(input=text,
output=word) %>%
count(chapter, word) %>%
  cast_dtm(document=chapter,
    term=word, value=n)

as.matrix(dtm)
```

```
    Terms
Docs is it what around comes goes
   1  2  2    1      0     0    0
   2  0  0    1      2     1    1
```

Let's practice!

TOPIC MODELING IN R

Displaying results with ggplot

TOPIC MODELING IN R

Pavel Oleinikov

Associate Director, Quantitative Analysis Center, Wesleyan University

Frequencies and probabilities

- We will be interested in displaying two kinds of data:
  - Word counts, and
  - Probabilities of topics and words.
- `ggplot` can do it all!
- Counts come as a tidy table. Results of LDA can be converted into a tidy format using function `tidy()`

From LDA model to tidy table

- When we fit a topic model, the result is an LDA model object.
- It contains two matrices: **beta** and **gamma**
  - **beta** contains probabilities of words in topics
  - **gamma** contains probabilities of topics in documents

```
lda_mod <- LDA(x=d2, k=2, method="Gibbs",
               control=list(alpha=1, delta=0.1, seed=10005))
str(lda_mod)
```

```
...
  ..@ beta           : num [1:2, 1:34] -5.68 -3.58 -3.29 -5.98 -5.68 ...
  ..@ gamma          : num [1:5, 1:2] 0.231 0.167 0.875 0.846 0.333 ...
```

Using function tidy

- Function `tidy` takes an LDA model object and returns a tidy table with a specified matrix.

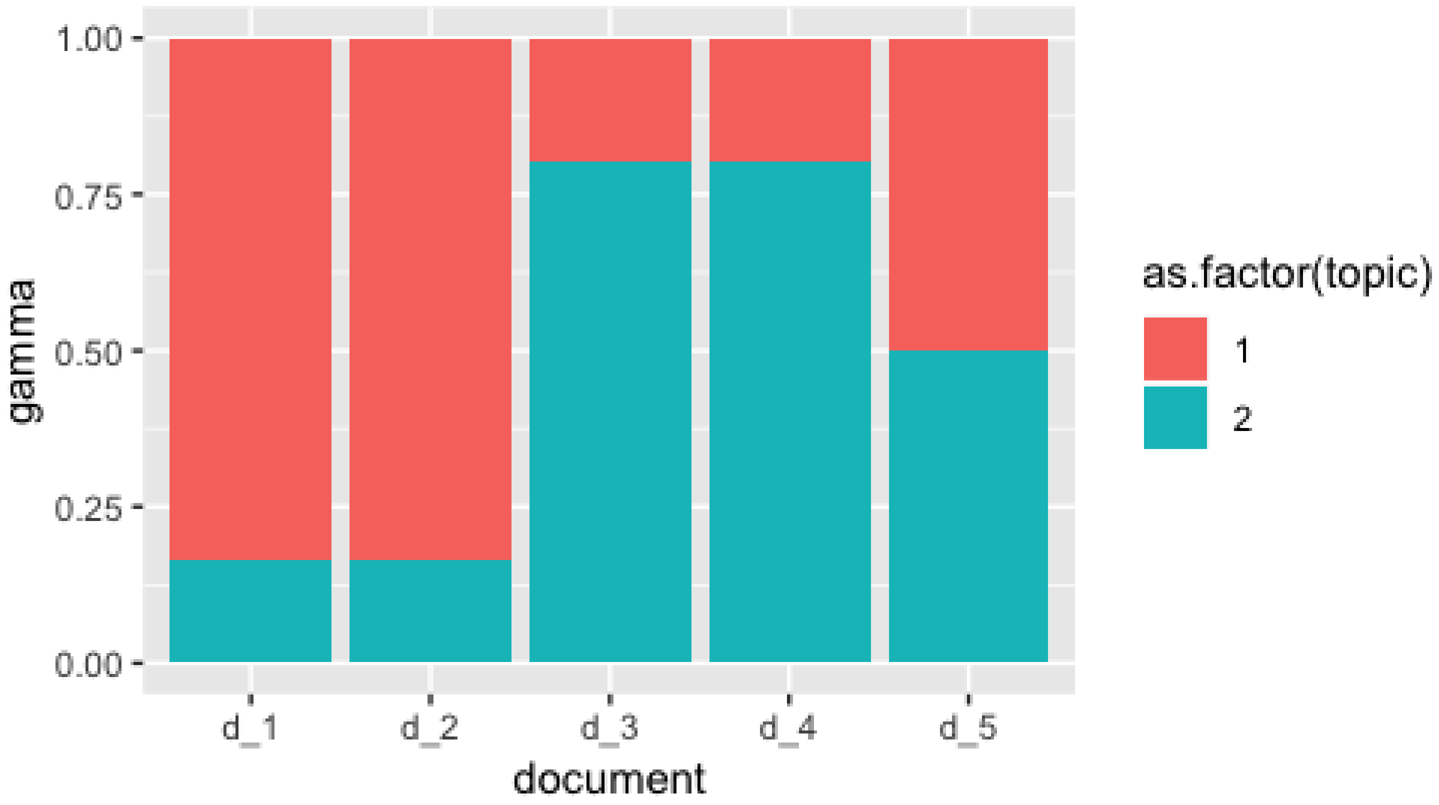
```
tidy(lda_mod, matrix="gamma")
```

```
   document topic gamma
   <chr>    <int> <dbl>
 1 d_1          1 0.231
 2 d_2          1 0.167
 3 d_3          1 0.875
 4 d_4          1 0.846
 5 d_5          1 0.333
 6 d_1          2 0.769
 7 d_2          2 0.833
 8 d_3          2 0.125
 9 d_4          2 0.154
10 d_5          2 0.667
```

Stacked columns chart

- `geom_col()` in `ggplot2` will produce a column chart
- by default, the columns will be stacked
- Calling `ggplot2` : the aesthetics specifies that values for axis `x` will come from column `document` , for axis `y` - from column `gamma`

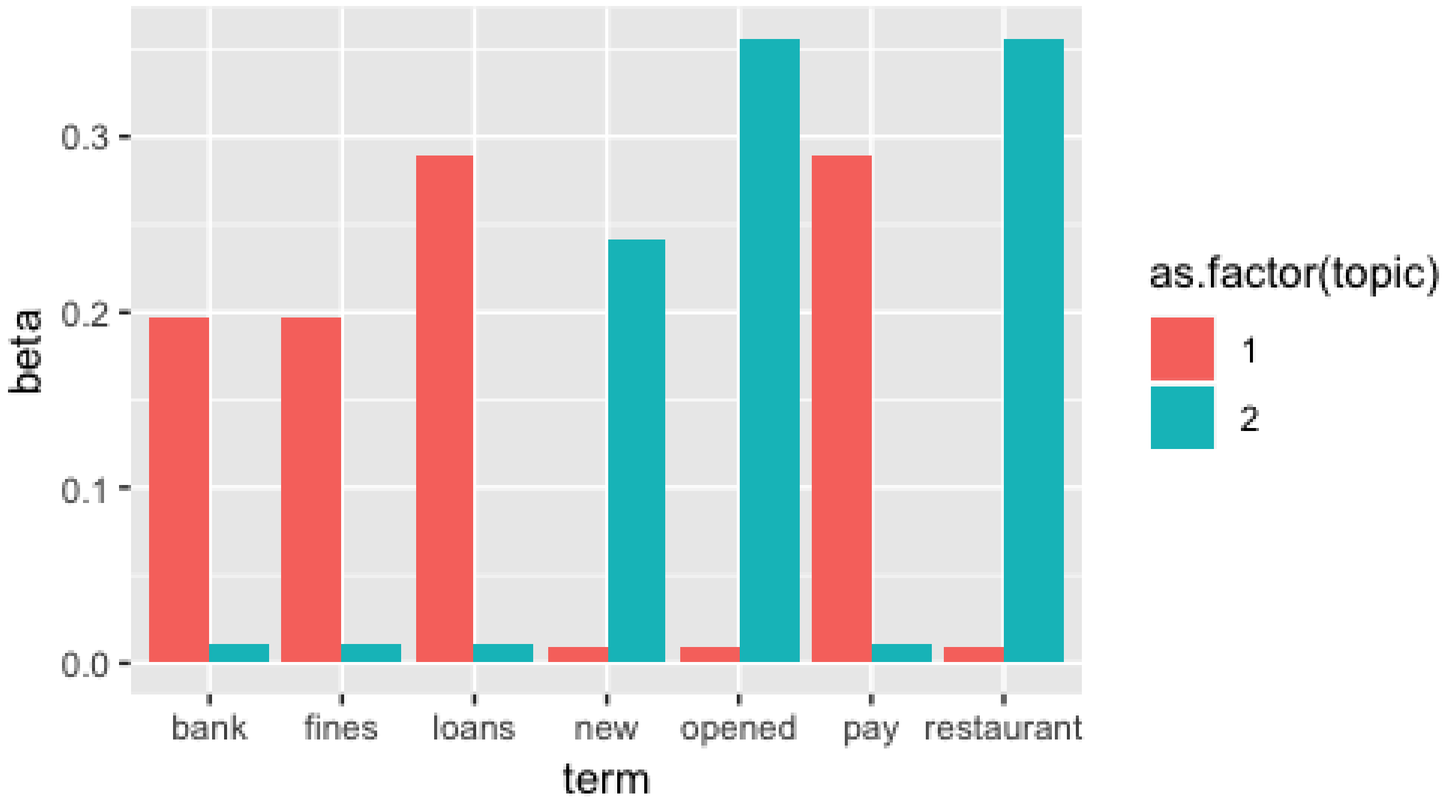
```
tidy(lda_mod, matrix="gamma") %>%
  ggplot(aes(x=document, y=gamma)) +
  geom_col(aes(fill=as.factor(topic)))
```



Dodged columns

- Matrix `beta` contains probabilities of words.

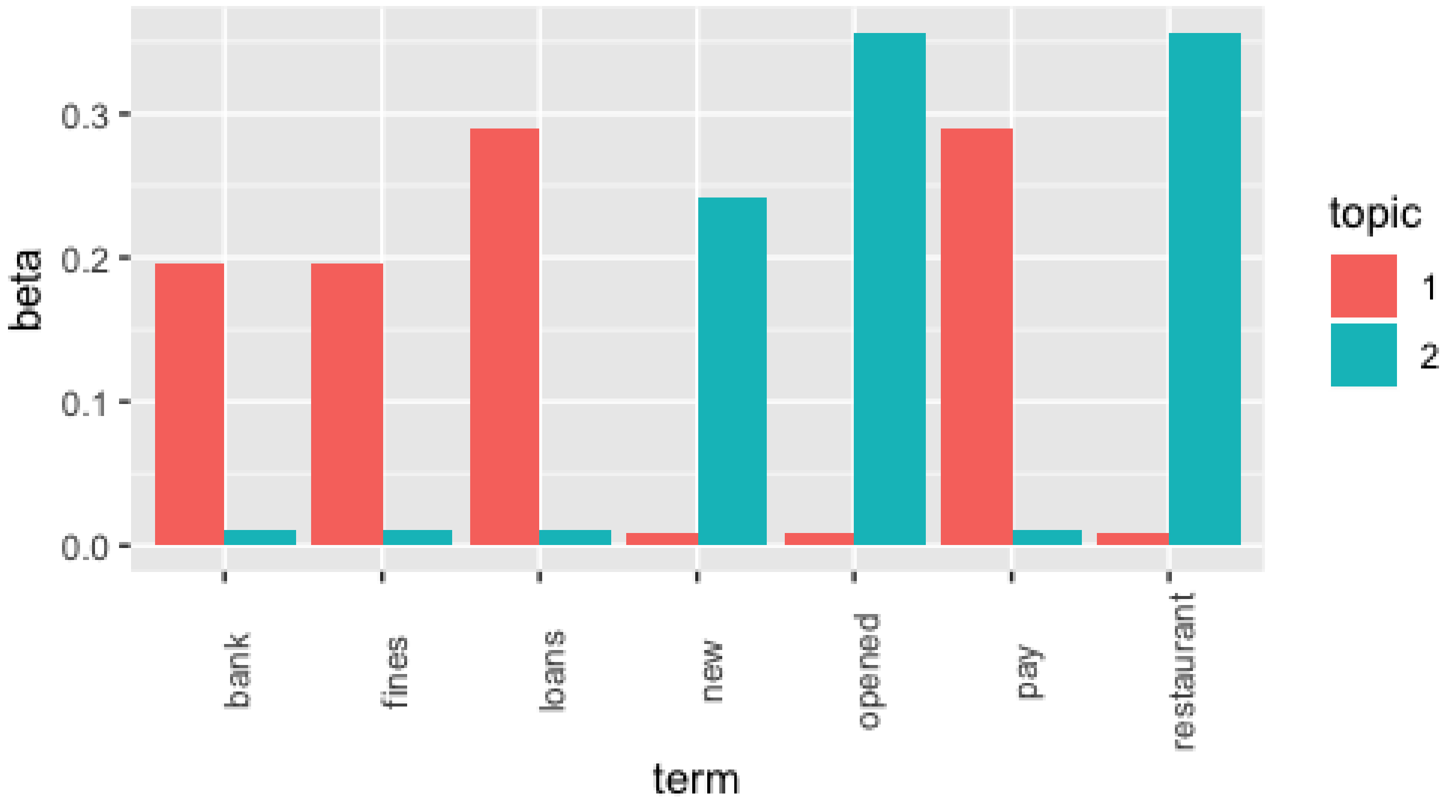
```
tidy(lda_mod, matrix="beta") %>%
ggplot(aes(x=term, y=beta)) +
geom_col(aes(fill=as.factor(topic)),
             position=position_dodge())
```



Rotated labels

- Text of labels on x axis is controlled through `axis.text.x` element in a `theme`

```
tidy(lda_mod, matrix="beta") %>%
mutate(topic = as.factor(topic)) %>%
ggplot(aes(x=term, y=beta)) +
geom_col(aes(fill=topic),
       position=position_dodge()) +
theme(axis.text.x = element_text(angle=90))
```



Let's practice!

TOPIC MODELING IN R