Levy & Franklin 2014

Note: Here we use some advanced functions. We separate those from external libraries by using code blocks that look like:

```
example command
```

Data structure

Each document is seen as a dictionary that has keys and values:

- **type**: describes the type of the document, such as regulatory evaluations performed by the FMCSA or a comment.
- **text**: the textual content of the document
- **author**: author’s name

Data selection and cleaning

This is a version which follows the computational logic I have introduced in the class. Their code version is most likely much less scary!
set `selected documents` as empty list

for each `document` in `documents`:
    if `document.type` == `document`:
        add `document` to `selected documents`

for each `document` in `selected documents`:
    set `words` as all words `document.text`
    set `cleaned text` as empty text
    for each `word` in `words`:
        if `word` not in `bad words`:
            add `word` to `cleaned text`

set `document.cleaned text` as `cleaned text` `document`
Topic modelling

store to topicmodel results from analysis on document.cleaned text and finding 8 topics

show most common words for topic 1 in topicmodel
show most common words for topic 2 in topicmodel
show most common words for topic 3 in topicmodel
show most common words for topic 4 in topicmodel
show most common words for topic 5 in topicmodel
show most common words for topic 6 in topicmodel
show most common words for topic 7 in topicmodel
show most common words for topic 8 in topicmodel

for each document in documents:

set topic as the topic number where document belongs the most in topicmodel

write to file document.author, topic