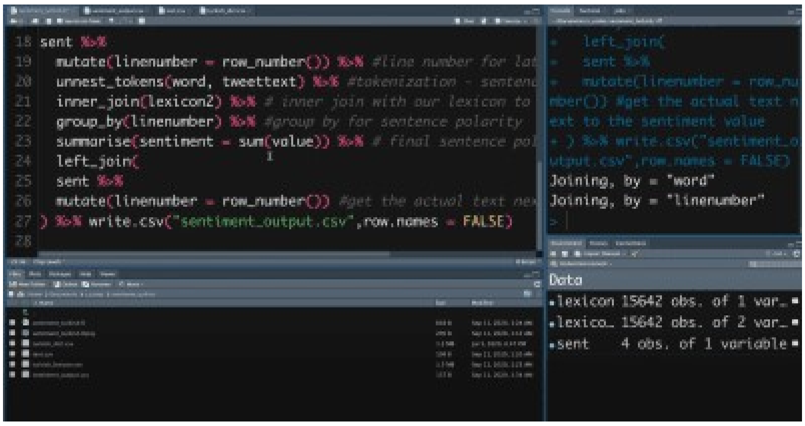
In this Sentiment Analysis tutorial, You'll learn how to use your custom lexicon (for any language other than English) or keywords dictionary to perform simple (slightly naive) sentiment analysis using R's `tidytext` package. Note: This isn't going to provide you the same accuracy as using the language model, but it's going to get you to the fastest solution (with some accuracy tradeoff). This example deals with Turkish Sentiment Analysis Script. Please note this tutorial doesn't include Text Pre-processing steps, but those are very important for any Text Analytics / NLP project.



```
sent %>%
  mutate(linenumber = row_number()) %>% #line number for lat
  unnest_tokens(word, tweettext) %>% #tokenization - sentenc
  inner_join(lexicon2) %>% # inner join with our lexicon to
  group_by(linenumber) %>% #group by for sentence polarity
  summarise(sentiment = sum(value)) %>% # final sentence pol
  left_join(
  sent %>%
  mutate(linenumber = row_number()) #get the actual text nex
) %>% write.csv("sentiment_output.csv",row.names = FALSE)
```

```
+   left_join(
+   sent %>%
+   mutate(linenumber = row_nu
mber()) #get the actual text n
ext to the sentiment value
+ ) %>% write.csv("sentiment_o
utput.csv",row.names = FALSE)
Joining, by = "word"
Joining, by = "linenumber"
>
```

```
Data
•lexicon 15642 obs. of 1 var…
•lexico… 15642 obs. of 2 var…
•sent     4 obs. of 1 variable
```

Video Walkthrough

Steps

- Read the Input Text as a Dataframe
- Load the lexicon / new language dictionary
- Select the appropriate columns – in this case, word and polarity
- Join the tokenized words from the text dataframe with the lexicon dataframe
- Roll-up the result dataframe based on the grouping variable (row_number) to get sentence level aggregated sentiment score

Code

```
library(tidyverse)

#install.packages("tidytext")
library(tidytext)

sent <- read.csv('text.csv')

lexicon <- read.table("turkish_lexicon.csv",
                      header = TRUE,
                      sep = ';',
                      stringsAsFactors = FALSE)

lexicon2 <- lexicon %>%
  select(c("WORD","POLARITY")) %>%
  rename('word'="WORD",'value'="POLARITY")


sent %>%
  mutate(linenumber = row_number()) %>% #line number for later sentence
grouping
  unnest_tokens(word, tweettext) %>% #tokenization - sentence to words
  inner_join(lexicon2) %>% # inner join with our lexicon to get the
polarity score
  group_by(linenumber) %>% #group by for sentence polarity
  summarise(sentiment = sum(value)) %>% # final sentence polarity from
words
  left_join(
  sent %>%
  mutate(linenumber = row_number()) #get the actual text next to the
sentiment value
) %>% write.csv("sentiment_output.csv",row.names = FALSE)
```