Structural Equation Modeling with Text Data

Johnny Zhang
University of Notre Dame

Workshop at the 2025 ISDSA Meeting
July 14, 2025

Outline

1. A brief introduction to structural equation modeling (SEM)

Outline

1. A brief introduction to structural equation modeling (SEM)
2. Text data

Outline

1. A brief introduction to structural equation modeling (SEM)
2. Text data
3. Methods to extract information from text data

Outline

1. A brief introduction to structural equation modeling (SEM)
2. Text data
3. Methods to extract information from text data
4. SEM with text data

Outline

1. A brief introduction to structural equation modeling (SEM)
2. Text data
3. Methods to extract information from text data
4. SEM with text data
5. R package TexSEM and online app BigSEM

Outline

1. A brief introduction to structural equation modeling (SEM)
2. Text data
3. Methods to extract information from text data
4. SEM with text data
5. R package TexSEM and online app BigSEM
6. Examples

Outline

Structural Equation Models

- Structural equation models are a collection of models:
  - Regression models
  - Mediation models
  - Factor models
  - MIMIC models

Structural Equation Models

- Structural equation models are a collection of models:
  - Regression models
  - Mediation models
  - Factor models
  - MIMIC models

- It synchronizes different models in the same general framework and allows flexible extension of them.

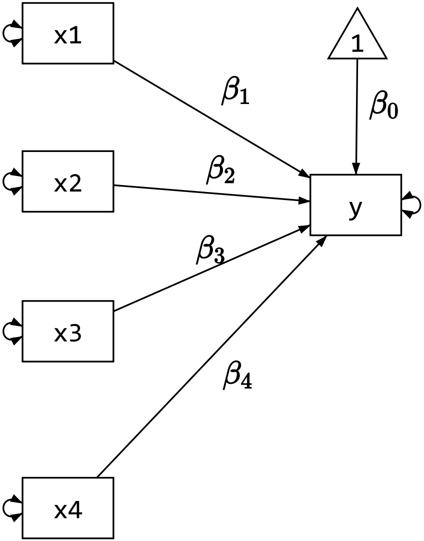
Structural Equation Models

- Structural equation models are a collection of models:
  - Regression models
  - Mediation models
  - Factor models
  - MIMIC models

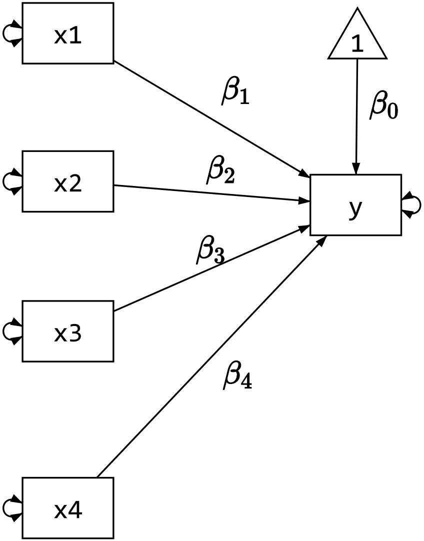
- It synchronizes different models in the same general framework and allows flexible extension of them.

- It frees researchers from estimating a model to focus on "building a model."

Example 1: Regression

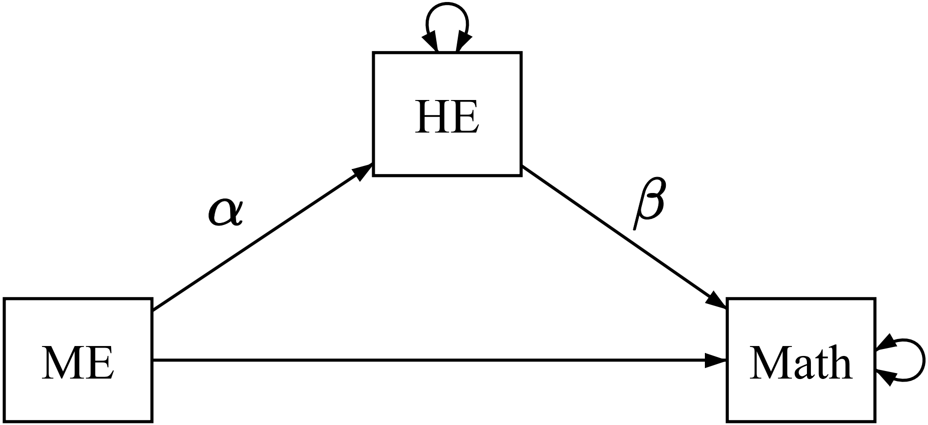


Example 1: Regression

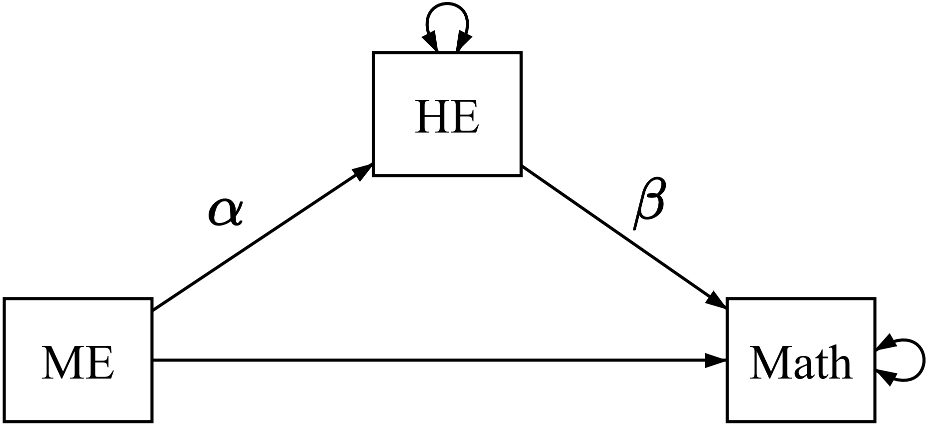


$$\begin{aligned} y =&\beta_0 + \beta_1 x_1 + \beta_2 x_2 \\ &+ \beta_3 x_3 + \beta_4 x_4 \end{aligned}$$

Example 2: Mediation or indirect effect $\alpha\beta$



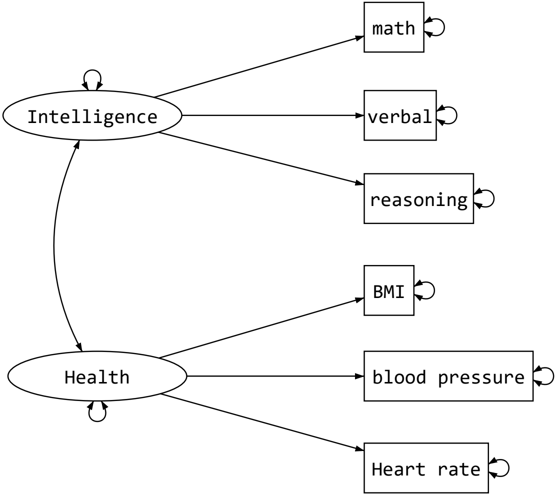
Example 2: Mediation or indirect effect $\alpha\beta$



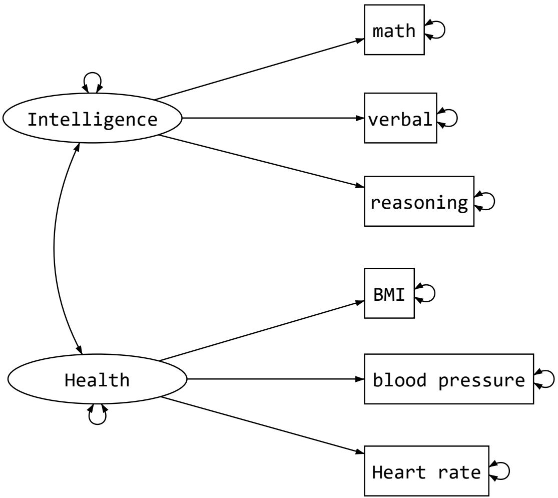
$$HE = \alpha \times ME$$

$$Math = \beta \times HE + \gamma \times ME$$

Example 3: Factor analysis

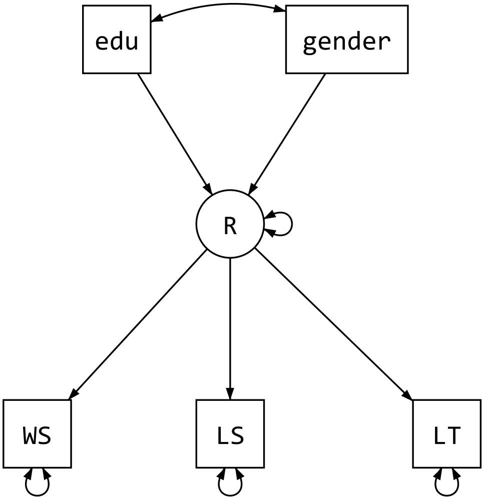


Example 3: Factor analysis

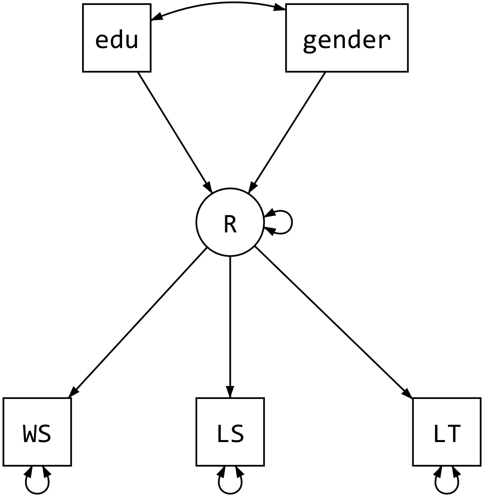


$$
\begin{aligned}
math =& \lambda_{11} \times Intelligence \\
verbal =& \lambda_{12} \times Intelligence \\
Reasoning =& \lambda_{13} \times Intelligence \\
BMI =& \lambda_{24} \times Health \\
Blood\ pressure =& \lambda_{25} \times Health \\
Heart\ rate & \lambda_{26} \times Health
\end{aligned}
$$

Example 4: MIMIC model

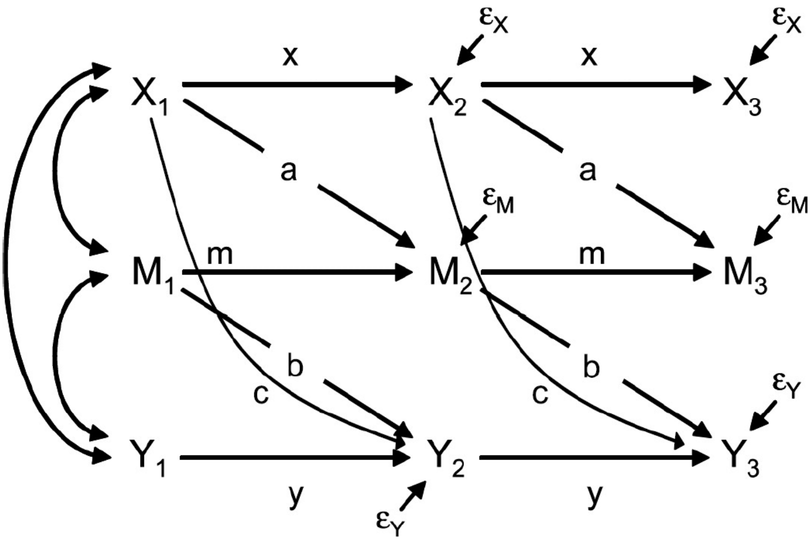


Example 4: MIMIC model



$$
\begin{aligned}
WS =& \lambda_1 \times R \\
LS =& \lambda_2 \times R \\
LT =& \lambda_3 \times R \\
R =& \beta_1 \times edu + \beta_2 \times gender
\end{aligned}
$$

Example 5: Cross-lag panel mediation model (Maxwell & Cole, 2007)



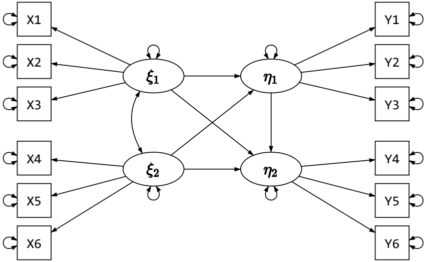
LISREL (LInear Structural RELationships) representation

$$\begin{aligned} \mathbf{x} &= \mathbf{\Lambda}_x \boldsymbol{\xi} + \boldsymbol{\delta} \\ \mathbf{y} &= \mathbf{\Lambda}_y \boldsymbol{\eta} + \boldsymbol{\epsilon} \\ \boldsymbol{\eta} &= \mathbf{B}\boldsymbol{\eta} + \mathbf{\Gamma}\boldsymbol{\xi} + \boldsymbol{\zeta} \end{aligned}$$

- $\boldsymbol{\eta}$: latent dependent (endogenous) variables
- $\boldsymbol{\xi}$: latent independent (exogenous) variables
- $\mathbf{B}$: coefficient matrix for latent dependent variables
- $\mathbf{\Gamma}$: coefficient matrix for latent independent variables
- $\mathbf{x}$ and $\mathbf{y}$: observed indicators of $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$
- $\boldsymbol{\delta}$ and $\boldsymbol{\epsilon}$: measurement error for $\mathbf{x}$ and $\mathbf{y}$
- $\mathbf{\Lambda}_x$ and $\mathbf{\Lambda}_y$: factor loadings for $\mathbf{x}$ and $\mathbf{y}$
- The first two equations are called the measurement equations.
- The third one is called the structural equation.

Path diagram

- A graphical representation of a SEM.
- Squares or rectangles: observed variables, data
- Circles or ovals: latent variables, factors, errors
- One-headed arrows: factor loadings, regression coefficients
- Two-headed arrows: variances, error variances, covariance
- `https://bigsem.psychstat.org/app/` or `https://semdiag.psychstat.org`



Variables in SEM

- Traditionally focus on continuous variables
- More general latent variable modeling framework
  - Categorical observed and latent variables
  - Count data
  - Survival data
  - ...
- New types of data
  - Text data
  - Network data
  - Image data
  - ...

Text data

- In real world, there is more qualitative information than quantitative information.
- Qualitative text data are widely collected in research and can come from many different sources.
- In diary studies, daily records on the activities and feelings of a day can be collected (Oppenheim, 2000).
- Text data can also come from the transcription of audio and video conversations from class observations (Bailey, 2008).
- For data collection using surveys or questionnaires, free response items are frequently used to solicit feedback (Rohrer et al., 2017).
- Compared to quantitative data collected through Likert scales, text data can provide more subtle information.
- Text data are largely under-analyzed in research.

Example data

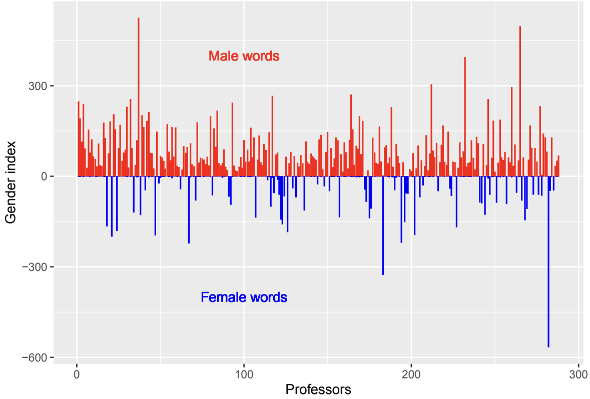
- Student evaluation of teaching data.
- 1,000 professors with 38,240 evaluations
- Each evaluation includes
  - The overall numerical rating of teaching of the instructor
  - How difficult the class was
  - Whether the student took the class for credit or not
  - Whether the class was an online class or not
  - Whether a textbook was used or not
  - The grade the student received
  - Text comment regarding the teaching of the instructor
  - A "tag" variable that kind of summarizes the evaluation
- We created a gender variable based on the text information.

Sample data

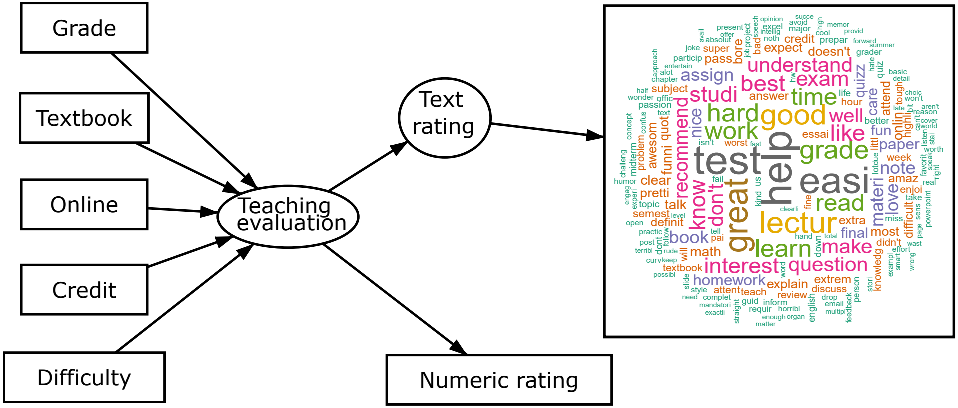
```
  id profid rating difficulty credit grade book take attendance
1  1      1      5          3      1     5    0    1          1
                                                          tags
1 respected;accessible outside class;skip class? you won't pass .
         comments
1 best professor i've had in college . only thing i dont like is the
        date gender sentiment
1 04/17/2018      1 0.1670451
```

Text can provide rich information

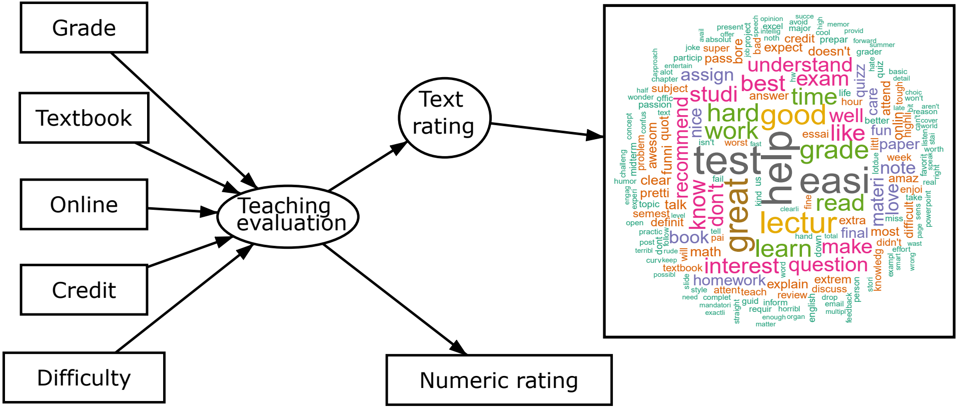
- It can convey subtle sentiment.
- It can provide a context.
- It may reveal information that is not intended to reveal.



A general SEM framework with text data

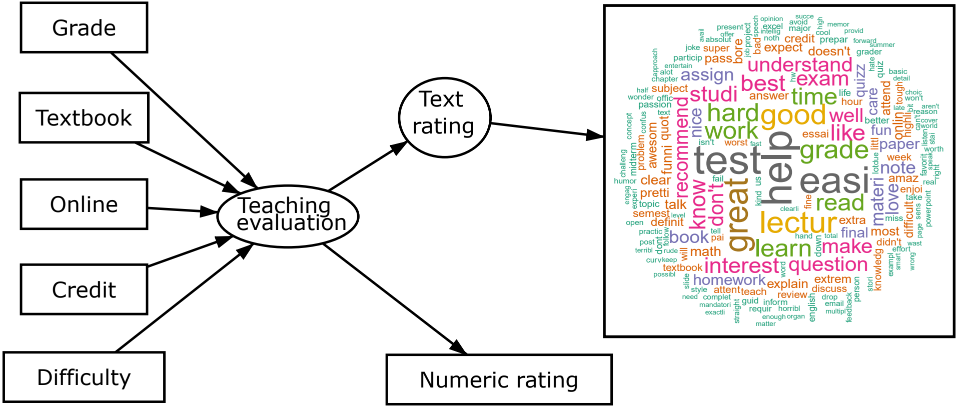


A general SEM framework with text data



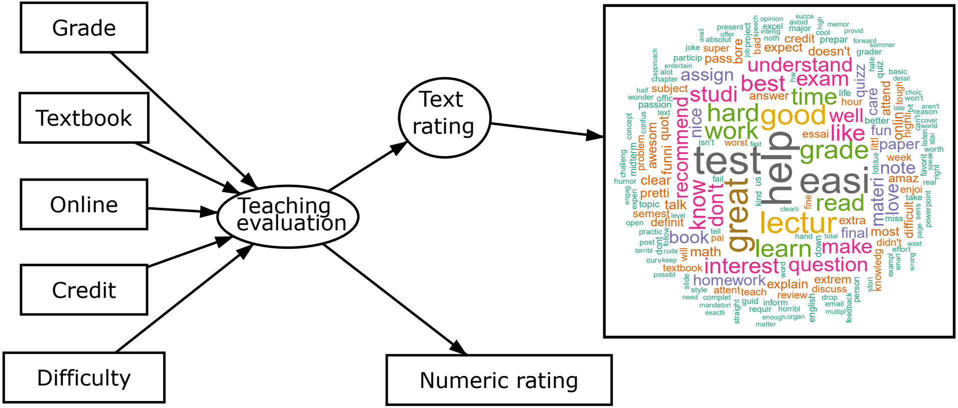
- Text as outcomes: What factors lead to the expression of the text.

A general SEM framework with text data



- Text as outcomes: What factors lead to the expression of the text.
- Text as predictors: How the writing can reduce stress.

A general SEM framework with text data



- Text as outcomes: What factors lead to the expression of the text.
- Text as predictors: How the writing can reduce stress.
- Text as mediators: How to promote diary writing then to reduce stress.

Understanding text information

- A major challenge in the analysis of text data is how to extract and quantify the information from them.
- Many methods have been developed in the area of computer science such as sentiment analysis (Hu & Liu, 2004a,b), topic modeling (Blei et al., 2003), and neural network methods (Deng & Liu, 2018).
- New large language models provide great opportunities (Brown et al., 2020; Radford et al., 2019).
- However, the existing methods often do not meet the needs of social, behavioral, and education research.
- For example, education researchers may be more interested in what factors are related to the positive and negative sentiments and how the different aspects of the comments are related to student outcomes such as course performance than obtaining the sentiments themselves.

Ways to extract text information

- Extracting or quantifying text information is an important step for utilizing text data.
- With the quantified information, many traditional statistical models can be used.
- Different ways can be used.
  1. Dictionary-based sentiment analysis
  2. Aspect-based sentiment analysis
  3. Topic modeling
  4. AI-based sentiment analysis
  5. Information extraction based on text encoders
  6. Information extraction through large language models

Dictionary-based sentiment analysis I

- The dictionary-based method is old yet efficient, in which each word is given a sentiment score.
- Many sentiment dictionaries are available such as the syuzhet dictionary (Jockers, 2017), AFINN (Nielsen, 2011), nrc (Mohammad & Turney, 2010) and bing (Hu & Liu, 2004a).
- For example, the syuzhet dictionary has a total of 10,748 words and each word has one of 16 sentiment scores ranging from -1 to 1.

word	score	word	score
warning	-0.5	illtreated	-1
extinguished	-0.25	uneducated	-0.8
pristine	1	doubtfully	-0.5
spirits	0.25	prejudices	-1

Dictionary-based sentiment analysis II

- Let W_j be the jth word in the dictionary with a total of M words, w_j be the sentiment score of the word W_j , and n_j is the frequency of the word in a text. If a word is not in the text, $n_j = 0$. The overall sentiment of a text is given by

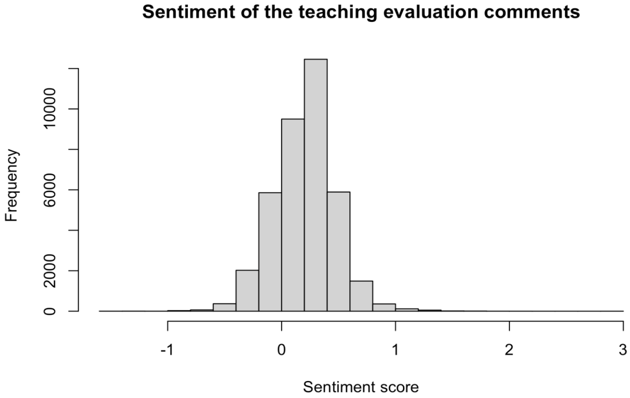
$$s = \sum_{j=1}^{M} n_j w_j, \tag{1}$$

  which is simply the sum of the scores of all the sentiment words.
- Typical methods can also take into consideration of modifiers in the text.
- If the overall sentiment of a text is of research interest, the dictionary-based sentiment analysis can be useful.

Dictionary based methods in R

```
prof1000$sentiment <- sentimentr::sentiment_by(
    prof1000$comments)$ave_sentiment
```

- For the teaching evaluation data, the code above can get the sentiment.



Aspect-based sentiment analysis

- In the aspect-based sentiment analysis, it is assumed that a text can be written around several aspects (Qu & Zhang, 2020).
- For example, one part of the teaching comment can be about the personality of the instructors and another part can be about the difficulty of the homework and exams.
- The method first extracts the aspects from the text and then obtains the sentiment score for each aspect as in the dictionary based methods.

Topic modeling I

- Topic models can be used to identify the topics and associated words in a text .
- Latent Dirichlet allocation (LDA) is a widely used method (Blei et al., 2003; Wilcox et al., 2023).
- For a given text, it can consist of one or all of K topics with different probabilities.
- Let z_{km} be the kth $(k = 1, \ldots, K)$ topic in the mth $(m = 1, \ldots, M)$ text. z_{km} takes a value between 1 and K. The topics can be generated from a multinomial distribution

$$z_{km} \sim Multinomial(\boldsymbol{\theta}_m) \tag{2}$$

Topic modeling II

- Once a topic is decided, words can be organized around it. Let $w_{mn}, n = 1, \ldots, N_m; m = 1, \ldots, M$, be the nth word to be used in the mth text and N_m denoting the total number of words in the text. w_{mn} would take a value between 1 and V with V being the total number of unique words used in all the comments. To model the process, a word is generated using

$$w_{mn} | z_{km} \sim Multinomial(\boldsymbol{\beta}_k)$$

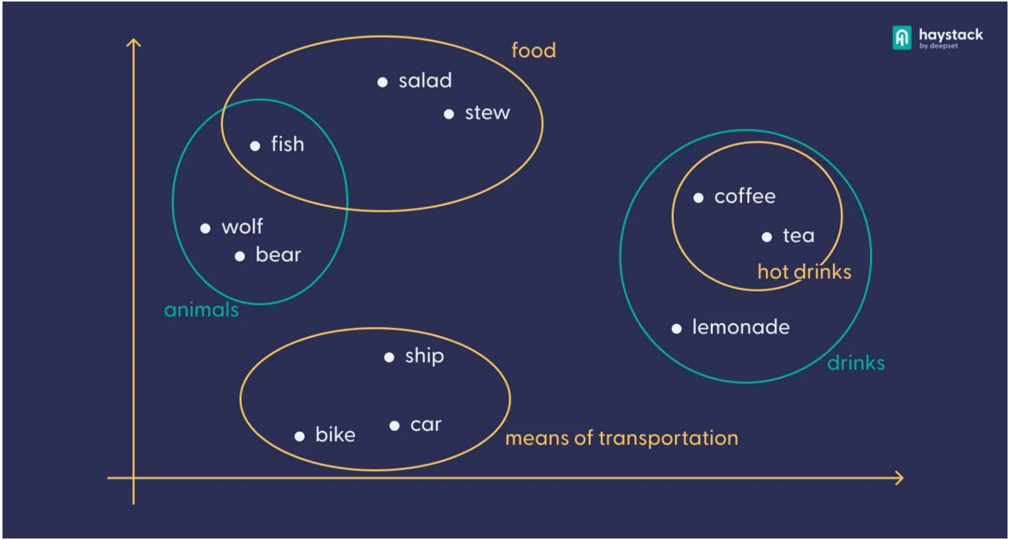
  where $\boldsymbol{\beta}_k = (\beta_{k1}, \beta_{k2}, \ldots, \beta_{kV})'$ is the probability that a word is picked given that topic k is selected.
- In topic models, the topic probabilities $\boldsymbol{\theta}$ can be used as the information extracted from the text.
- For topic modeling in R, the package topicmodels can be used.

Text embedding and encoders I

- Encoding or embedding is a way of representing data as points in n-dimensional space so that similar data points cluster together.
- It can convert text (words, sentences, or documents) into numerical vectors that capture their meaning and semantic relationships.
  - Quantification: text to numbers
  - Similar texts are closer in the space represented by the vectors.
  - Multilingual text: Dog, 狗, hund, كلب

Text embedding and encoders

- Embed words



Text embedding and encoders

- Embed sentences.

Text embedding and encoders I

- Variety of methods are available to embed words and sentences into vectors (Perone et al., 2018).
  - Latent semantic analysis (LSA)
  - Word2vec
  - Recurrent neural network
  - Long short-term memory
  - Transformer

Text embedding and encoders II

Name	Training method[1]	Embedding size
ELMo (BoW, all layers, 5.5B)	Self-supervised	3072
ELMo (BoW, all layers, original)	Self-supervised	3072
ELMo (BoW, top layer, original)	Self-supervised	1024
Word2Vec (BoW, Google news)	Self-supervised	300
p-mean (monolingual)	–	3600
FastText (BoW, Common Crawl)	Self-supervised	300
GloVe (BoW, Common Crawl)	Self-supervised	300
USE (DAN)	Supervised	512
USE (Transformer)	Supervised	512
InferSent (AllNLI)	Supervised	4096
Skip-Thought	Self-supervised	4800

- The encoders can be viewed as factor analysis or principle component analysis method yet typically nonlinear.

Text embedding and encoders III

- The Universal Sentence Encoder (USE) encodes text into a 512 dimensional vector (Cer et al., 2018).

```
test_embed <- embed_text(c('cat', 'dog', 'apple', 'animal
    ', 'fruit'))
test_embed[1, 1:5]
[1]  0.0084  0.0469  0.0510 -0.0392 -0.0675

cor(t(test_embed))
          cat   dog apple  animal fruit
cat     1.000 0.814 0.443  0.714 0.434
dog     0.814 1.000 0.431  0.785 0.450
apple   0.443 0.431 1.000  0.395 0.672
animal  0.714 0.785 0.395  1.000 0.494
fruit   0.434 0.450 0.672  0.494 1.000
```

Text embedding and encoders IV

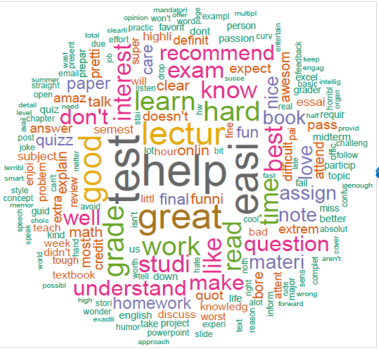
- Embeddings can inherit biases.

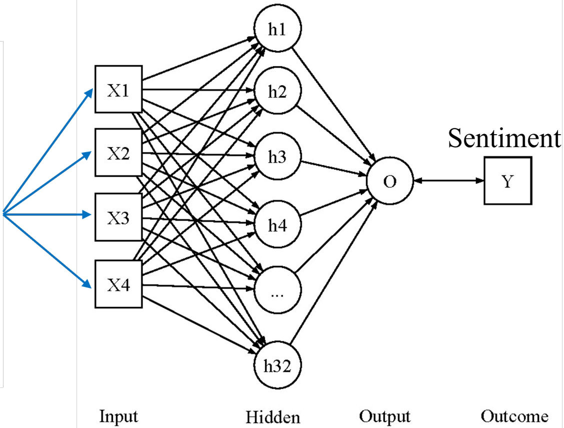
```
test_embed <- embed_text(c('male', 'engineer', '
    construction worker', 'female', 'nurse', 'elementary
    school teacher'))
round(cor(t(test_embed)), 3)
          male engineer worker female nurse teacher
male     1.000    0.381  0.236  0.946 0.309   0.135
engineer 0.381    1.000  0.534  0.383 0.527   0.324
worker   0.236    0.534  1.000  0.207 0.435   0.395
female   0.946    0.383  0.207  1.000 0.369   0.151
nurse    0.309    0.527  0.435  0.369 1.000   0.452
teacher  0.135    0.324  0.395  0.151 0.452   1.000
```

- We can also embed texts using large language models including GPT (Open AI), ERNIE (Baidu), Qwen (Alibaba), Llama (Facebook), and Gemini (Google).

Sentiment analysis based on embeddings

- With sentiment labeled text data, one can construct a model, including simple regression models and neural network models, to get the sentiment.





- The model can be trained and saved to get the sentiment of new data.

A simple yet efficient sentiment analysis model

- The R package sentiment.ai includes models based on the Universal Sentence Encoder (USE).
- USE turns the text into a 512 dimension vector.
- A regression model and boosted tree model are estimated based on labeled data.
- Sentiment scores of new text can be calculated based on a selected model.

Sentiment from sentiment.ai I

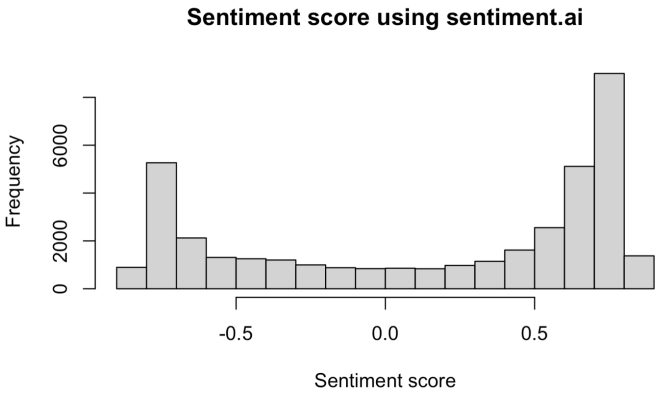
- To use the sentiment.ai package, we need to install it.
- We have included functions to install it as a part of the TextSEM package.

```
textsem_install() ## first time use it
textsem_init() ## initialize each time using the R package
```

- For the teaching evaluation data, the code below can get the sentiment.

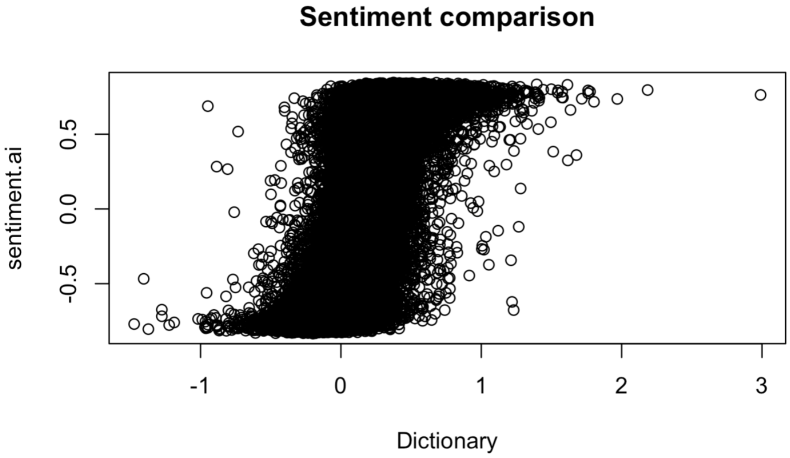
```
set.score <- sentiment_score(prof1000$comments)
```

Sentiment from sentiment.ai II



Comparison of dictionary based method and embedding method

- Correlation is 0.7.
- More complex models can be developed.



Sentiment based on LLMs I

- One can prompt a LLM to give a sentiment score.
- For example, for ChatGPT, we can use something like "Analyze the following comment and determine the sentiment score from 0 to 1 - most negative to most positive with 0.5 being neutral. Return answer of the score, with only the score, not other text: Mrs . 's class was interesting, and I would highly recommend her for any english course . She's a dramatic speaker, and this makes class fun ."
- For the teaching evaluation example, the code below can be used.

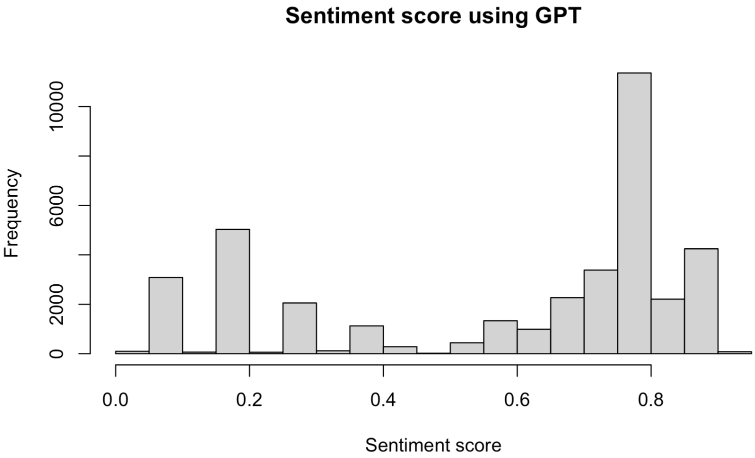
```
openai <- import("openai")
openai$api_key <- "sk-proj-xxxx"
## run all for teaching evaluation
gpt_score <- rep(0, nrow(prof1000))
for (i in 1:nrow(prof1000)) {
  response <- openai$chat$completions$create(
    model = "gpt-4.1-nano",
```

Sentiment based on LLMs II

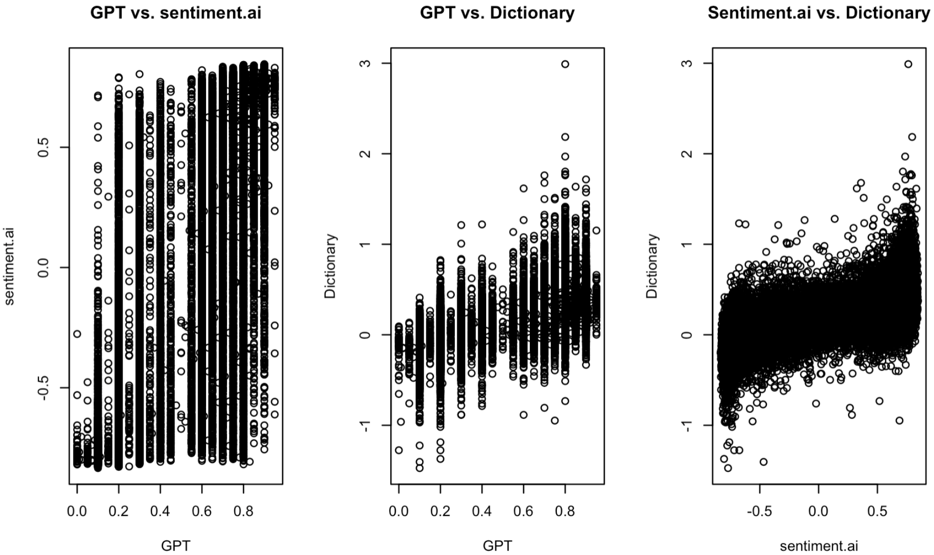
```
      messages = list(
      list(role = "system", content = "You are trained to
         analyze and detect the sentiment of teaching
         evaluation comments. If you are unsure of an answer
         , you can say 999."),
      list(role = "user", content = paste0("Analyze the
         following comment and determine the sentiment score
          from 0 to 1 - most negative to most positive with
         0.5 being neutral. Return answer of the score, with
          only the score, not other text:", prof1000$comment
         [i]))
  )
)
gpt_score[i] <- (response$choices[[1]]$message$content)
}
```

Sentiment based on LLMs III

- Correlation with dictionary: 0.68; with sentiment.ai: 0.83.



Comparison of different methods



SEM with text data

- Let $\mathbf{t}$ denote the information extracted from the text, we can write the SEM model with text information in the format of Bentler-Weeks (Bentler & Weeks, 1980) model as
$$\begin{pmatrix} \boldsymbol{\eta}_i \\ \mathbf{t}_i^{+} \end{pmatrix} = \boldsymbol{\beta} \begin{pmatrix} \boldsymbol{\eta}_i \\ \mathbf{t}_i^{+} \end{pmatrix} + \boldsymbol{\gamma} \begin{pmatrix} \boldsymbol{\xi}_i \\ \mathbf{t}_i^{-} \end{pmatrix}. \tag{3}$$
- $\mathbf{t}^{+}$ and $\mathbf{t}^{-}$ represent endogenous and exogenous variables, respectively.

Model estimation: one-stage vs. two-stage methods

- One-stage method: For topic modeling and encoders, the model can be combined and estimated as one large model.
  - Pros: can be more efficient in general
  - Cons: hard to estimate, lack of inference methods
- Two-stage method:
  - Pros: easy to estimate, can use all SEM techniques, text information can be repeatedly used
  - Cons: may lose statistical efficiency

One stage vs. two stage

Encoder	Regressor	Train RMSE	Train R^2	Test RMSE	Test R^2
BERT	Linear	0.654	0.722	0.851	0.557
BERT	Lasso (alpha=0.01)	0.883	0.494	0.897	0.507
BERT	Lasso + CV	0.745	0.640	0.783	0.625
BERT	Ridge (alpha=0.01)	0.654	0.722	0.850	0.558
BERT	Ridge + CV	0.712	0.671	**0.769**	**0.638**
BERT	BERT	0.371	0.910	0.718	0.685
BERT	BERT (emb-freezed)	0.317	0.935	**0.674**	**0.722**
DistilBERT	Linear	0.665	0.713	0.899	0.515
DistilBERT	Lasso	0.923	0.447	0.935	0.475
DistilBERT	LassoCV	0.766	0.619	0.798	0.618
DistilBERT	Ridge	0.665	0.713	0.894	0.520
DistilBERT	RidgeCV	0.765	0.620	**0.790**	**0.625**
DistilBERT	FNN (hidden=512)	1.583	-0.625	0.768	0.638
DistilBERT	DistilBERT	0.582	0.780	0.668	0.727
DistilBERT	DistilBERT (emb-freezed)	0.403	0.895	**0.657**	**0.736**
SentenceBERT	Linear	0.819	0.565	0.912	0.501
SentenceBERT	Lasso (alpha=0.01)	1.111	0.200	1.166	0.184
SentenceBERT	Lasso + CV	0.856	0.525	0.910	0.503

How to do the data analysis

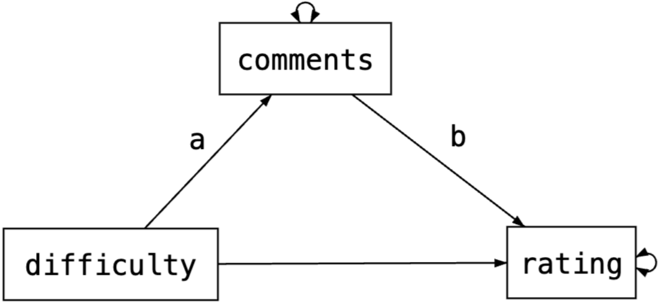
- One can first extract text data information and then fit a SEM model through any SEM software program such as OpenMx and lavaan in R or Mplus.
- We integrate the two-stage method in the R package TextSEM and the online app BigSEM.

Examples

- Sentiment based analysis
  - Example 1. Using sentiment scores from the dictionary-based sentiment analysis
  - Example 2. Using sentiment scores from sentiment.ai
  - Example 3. Using sentiment scores from ChatGPT
- Example 4. Using information extraction based on text encoders/embeddings
- Example 5. More than one text variable

Example 1. Using dictionary-based sentiment I

- In this example, the overall sentiment of comment is extracted and used as a mediator between difficulty of the course and the teaching rating.



- The model can be specified using strings as for the lavaan package

Example 1. Using dictionary-based sentiment II

```
model <- ' rating ~ difficulty + b*comments
           comments ~ a*difficulty
           ab := a*b
         '
```

- To estimate the model, we use the sem.sentiment function. By default, the dictionary based method is used.

```
res <- sem.sentiment(model = model,
                     df = prof1000,
                     text_var=c('comments'))
summary(res$estimates)
```

- The analysis created a new variable called "comments.sentiment" and replaced the text comment variable with it.
- The output is given below.

Example 1. Using dictionary-based sentiment III

```
lavaan 0.6-19 ended normally after 2 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         9

  Number of observations                         38240
  Number of missing patterns                         1

Model Test User Model:

  Test statistic                                 0.000
  Degrees of freedom                                 0

Parameter Estimates:
```

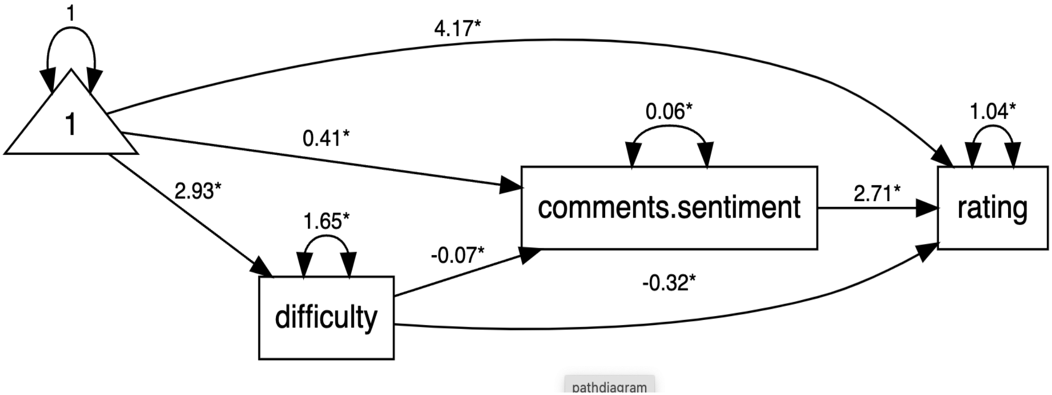
Example 1. Using dictionary-based sentiment IV

```
  Standard errors                                Standard
  Information                                    Observed
  Observed information based on                   Hessian

Regressions:
                       Estimate  Std.Err  z-value  P(>|z|)
  rating ~
    difficulty           -0.322    0.004  -74.258    0.000
    cmmnts.snt (b)        2.712    0.021  129.244    0.000
  comments.sentiment ~
    difficulty (a)       -0.072    0.001  -72.843    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
```

Example 1. Using dictionary-based sentiment V

```
    .rating             4.169    0.016  266.486    0.000
    .commnts.sntmnt     0.415    0.003  130.894    0.000
     difficulty         2.928    0.007  445.625    0.000

Variances:
                     Estimate  Std.Err  z-value  P(>|z|)
    .rating             1.044    0.008  138.275    0.000
    .commnts.sntmnt     0.062    0.000  138.275    0.000
     difficulty         1.651    0.012  138.275    0.000

Defined Parameters:
                     Estimate  Std.Err  z-value  P(>|z|)
     ab                -0.196    0.003  -63.458    0.000
```

Example 1. Using dictionary-based sentiment



Example 2. Using sentiment based on sentiment.ai I

- The R function sem.sentiment allows the use of sentiment.ai to extract sentiment information.

```
model <- ' rating ~ difficulty + b*comments
           comments ~ a*difficulty
           ab := a*b
         '

res <- sem.sentiment(model = model,
                     df = prof1000,
                     text_vars=c('comments'),
                     method = 'sentiment.ai')
summary(res$estimates)
```

- The output of the analysis: (Note that the parameter estimates cannot be directly compared as the sentiment scores have different scales.

Example 2. Using sentiment based on sentiment.ai II

```
Regressions:
                      Estimate  Std.Err  z-value  P(>|z|)
  rating ~
    difficulty          -0.225    0.004  -58.358    0.000
    cmmnts.snt (b)       1.532    0.008  187.509    0.000
  comments.sentiment ~
    difficulty (a)      -0.191    0.002  -86.943    0.000

Intercepts:
                  Estimate  Std.Err  z-value  P(>|z|)
   .rating           4.197    0.013  330.940    0.000
   .commnts.sntmnt   0.717    0.007  101.753    0.000
    difficulty       2.928    0.007  445.625    0.000

Variances:
```

Example 2. Using sentiment based on sentiment.ai III

```
                   Estimate  Std.Err  z-value  P(>|z|)
    .rating           0.781    0.006  138.275    0.000
    .commnts.sntmnt   0.306    0.002  138.275    0.000
     difficulty       1.651    0.012  138.275    0.000

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
     ab              -0.293    0.004  -78.877    0.000
```

Example 3. Using sentiment from ChatGPT I

- It is suggested to first get the sentiment from ChatGPT and then conduct the SEM analysis using lavaan.
- The reason is that ChatGPT output may include errors that need to to be fixed beforehand.
- The example is based on the sentiment scores from ChatGPT earlier, which are saved in the file gpt_scores.csv.

```
gpt_scores <- read.csv("gpt_scores.csv")

prof1000$gpt_score <- gpt_scores$gpt_score

model <- ' rating ~ difficulty + b*gpt_score
           gpt_score ~ a*difficulty
           ab := a*b
         '
```

Example 3. Using sentiment from ChatGPT II

```
res <- sem(model = model, data = prof1000)
summary(res)
```

- The output of the analysis: (Note that some parameter estimates cannot be directly compared as the sentiment scores have different scales.

```
Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  rating ~
    difficulty       -0.131    0.003  -41.851    0.000
    gpt_score   (b)   4.027    0.014  278.761    0.000
  gpt_score ~
    difficulty (a)   -0.096    0.001  -97.039    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
```

Example 3. Using sentiment from ChatGPT III

```
    .rating              0.495    0.004  138.275    0.000
    .gpt_score           0.062    0.000  138.275    0.000

Defined Parameters:
                      Estimate  Std.Err  z-value  P(>|z|)
    ab                  -0.387    0.004  -91.645    0.000
```

Example 4. Using information extraction based on text encoders/embeddings I

- We first apply the Universal Sentence Encoder to teaching comments and get the embedded vectors.
- The resulting data is a 38240×512 matrix matrix with 512 columns, each representing a dimension of the embedded vector.
- We saved the embedded vectors in the file use_embed_all.RData. They can be loaded and used in the future.

```
textsem_init()
text_embed_all <- sentiment.ai::embed_text(
   prof1000$comments, batch_size=20)
# rename the columns
colnames(text_embed_all) <- paste0('v', 1:512)
rownames(text_embed_all) <- 1:nrow(text_embed_all)
save(text_embed_all, file="use_embed_all.RData")
```

Example 4. Using information extraction based on text encoders/embeddings II

- We now investigate whether the text comment as embedded vectors is a mediator. The mediation model with text data would be (1) Model 1:

$$rating_i = \beta_0 + \sum_{j=1}^{512} \beta_j v_{ij} + c' \times difficulty_i + e_i$$

- and Model 2 - another 512 regression models below:

$$v_{ij} = \gamma_j + \alpha_j \times difficulty_i + ev_{ij}, \quad j = 1, \ldots, 512$$

- With the models, the total mediation effect is $\sum_{j=1}^{512} \alpha_j \times \beta_j$.
- Given the meaning of each embedded vector is not clear, it is not very helpful to look at individual mediation path $\alpha_j \beta_j, j = 1, \ldots, 512$.
- Although theoretically we can estimate the model as a SEM, the existing software may have trouble handling such high-dimensional data. Instead, we use regression models here directly.

Example 4. Using information extraction based on text encoders/embeddings III

- We first estimate Model 1 and save the β parameters.

```
med.data <- cbind(prof1000$rating, prof1000$difficulty,
   text_embed_all)
med.data <- as.data.frame(med.data)
names(med.data)[1:2] <- c('rating', 'diff')

m1 <- lm(rating ~ ., data = med.data)
summary(m1)$r.squared

## save the parameters and their standard errors
m1.est <- summary(m1)$coefficients[-(1:2), 1:2]
```

- We now estimate Model 2 and save the α parameters.

Example 4. Using information extraction based on text encoders/embeddings IV

```
m2.est <- array(dim=c(512, 2))

for (i in 1:512){
  temp.model <- lm(med.data[, (i+2)] ~ med.data[, 2])
  m2.est[i, ] <- summary(temp.model)$coefficients[2, 1:2]
}
```

- Given the estimates, the total mediation effect estimate is

$$\hat{med} = \sum_{j=1}^{512} \hat{\alpha}_j \times \hat{\beta}_j$$

  and its standard error can be estimated as

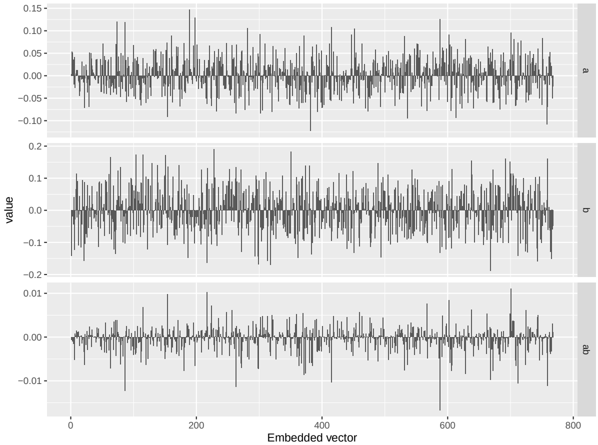
$$\hat{se}(\hat{med}) = \sqrt{\sum_{j=1}^{512} \hat{Var}(\hat{\alpha}_j \times \hat{\beta}_j)} = \sqrt{\sum_{j=1}^{512} [\hat{\alpha}_j^2 \hat{se}(\hat{\beta}_j)^2 + \hat{\beta}_j^2 \hat{se}(\hat{\alpha}_j)^2]}$$

Example 4. Using information extraction based on text encoders/embeddings V

Based on the results, we can conduct a z-test.

- Here, the mediation effect is -0.327.

Example 4. Using information extraction based on text encoders/embeddings VI



Example 4 use TextSEM I

- TextSEM includes a function to conduct similar analysis.
  - It can embed the text.
  - It can conduct dimension reduction.
  - It can then estimate the model.
  - However, it may not work as reliable yet.
- It is recommended to first embed the text and conduct dimension reduction first.
- The code below shows how to do the analysis.
  - The text was first embedded into 384 dimension vectors.
  - The vectors were reduced to 5 dimensions based on singular value decomposition.

Example 4 use TextSEM II

```
embeddings <- sem.encode(prof1000$comments,
encoder = "paraphrase-MiniLM-L6-v2")

save(embeddings, file="prof1000.emb.rda")

model <- ' rating ~ difficulty + comments
           comments ~ difficulty
         '
res <- sem.emb(sem_model = model,
               data = prof1000,
               text_var = "comments",
               emb_filepath = "prof1000.emb.rda")
summary(res$estimates)
```

- The mediation effect is -0.136.

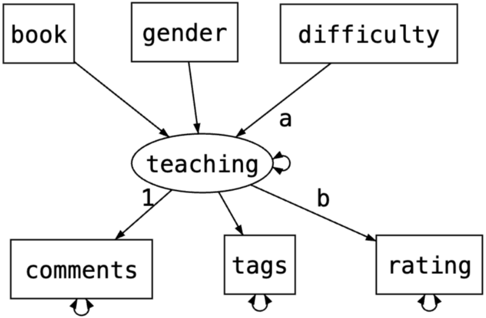
Comparison and interpretation

Methods	Mediation effects	% of total
Dictionary sentiment	-.196	37.8%
AI sentiment	-.293	56.6%
USE	-.327	63.1%
ChatGPT	-.387	76.6%
BERT (SVD 5)	-.137	26.4%

- The information in text can explain up to 76.6% of total effect among the evaluated methods.
- Class difficulty is associated with negative thoughts, which, in turn, lead to low ratings.

Example 5. More than one text variable I

- In the teaching evaluation data set, there are two text variables – comments and tags.
- We can form a teaching evaluation factor using the two text variables and the rating score.
- Then we can study the factors that are related to teaching evaluation.



Example 5. More than one text variable II

- The code for the analysis

```
model <- ' teaching =~ tags + comments + b*rating
           teaching ~ book + gender + a*difficulty
           ab := a*b
         '

res <- sem.sentiment(model = model,
                     df = prof1000,
                     text_vars=c('comments', 'tags'))

summary(res$estimates)
```

- The results are

Example 5. More than one text variable III

```
Latent Variables:
                     Estimate  Std.Err  z-value  P(>|z|)
  teaching =~
    tgs.sntmnt          1.000
    cmmnts.snt          6.566    0.225   29.130    0.000
    rating      (b)    47.398    1.633   29.020    0.000

Regressions:
                     Estimate  Std.Err  z-value  P(>|z|)
  teaching ~
    book                0.006    0.000   16.671    0.000
    gender              0.004    0.000   12.711    0.000
    difficulty (a)     -0.011    0.000  -28.388    0.000

Covariances:
```

Example 5. More than one text variable IV

```
                      Estimate   Std.Err   z-value   P(>|z|)
  book ~~
    gender              -0.006     0.001    -4.871     0.000
    difficulty           0.030     0.004     8.645     0.000
  gender ~~
    difficulty          -0.003     0.003    -0.844     0.399

Intercepts:
                      Estimate   Std.Err   z-value   P(>|z|)
   .tags.sentiment       0.096     0.001    74.411     0.000
   .commnts.sntmnt       0.372     0.003   112.293     0.000
   .rating               4.992     0.020   251.974     0.000
    book                 0.672     0.003   244.902     0.000
    gender               0.616     0.002   247.842     0.000
    difficulty           2.928     0.007   445.625     0.000
```

Example 5. More than one text variable V

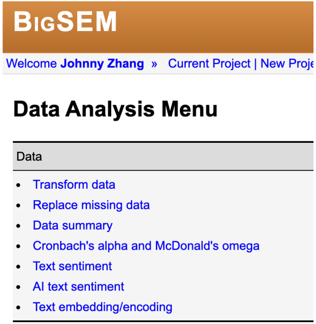
```
Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .tags.sentiment    0.028    0.000  137.753    0.000
   .commnts.sntmnt    0.039    0.000   91.868    0.000
   .rating            0.281    0.016   17.264    0.000
   .teaching          0.001    0.000   14.716    0.000
    book              0.220    0.002  120.641    0.000
    gender            0.236    0.002  138.275    0.000
    difficulty        1.651    0.012  138.275    0.000

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
    ab               -0.522    0.005 -107.623    0.000
```

Online app - BigSEM

- `https://bigsem.psychstat.org/app`
- An online app with both graphical and programming interface.
- Server setup
  - Ubuntu on Amazon Elastic Compute Cloud (EC2)
  - Apache web server + PHP + MySQL + R + Python
  - HTML + JavaScript
- Similarly analyses in R can be conducted online.

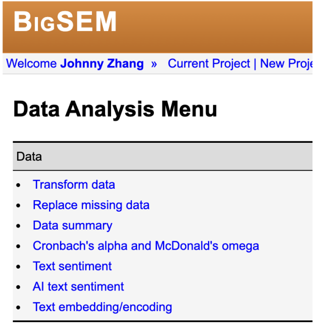
Obtain the dictionary-based sentiment scores

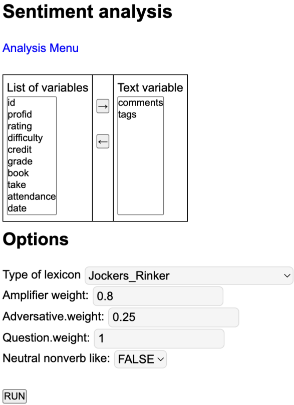
- BigSEM includes a simple app to get the sentiment scores



Obtain the dictionary-based sentiment scores

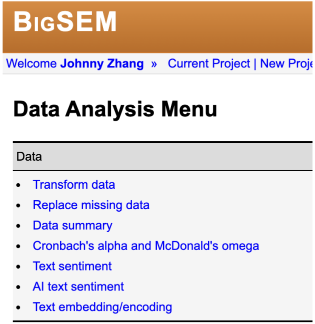
- BigSEM includes a simple app to get the sentiment scores

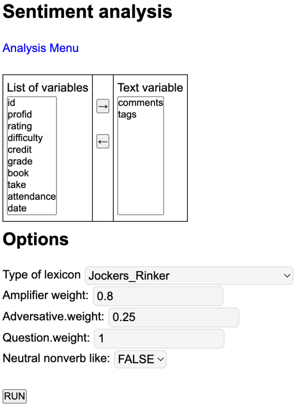


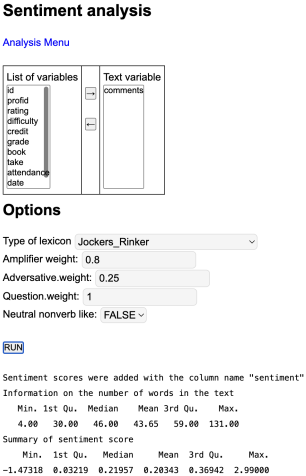


Obtain the dictionary-based sentiment scores

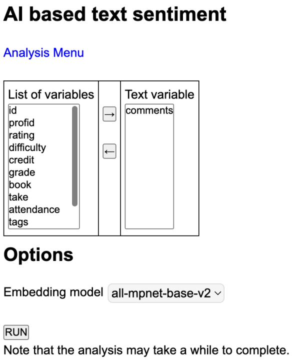
- BigSEM includes a simple app to get the sentiment scores





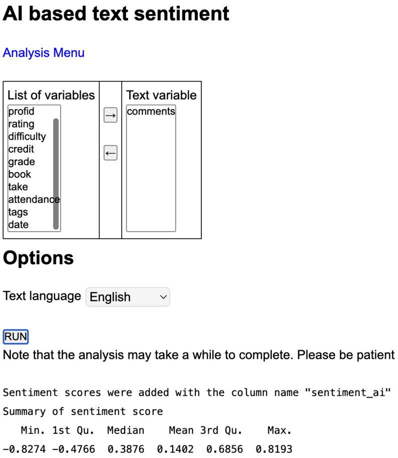


Text can be embedded using BigSEM



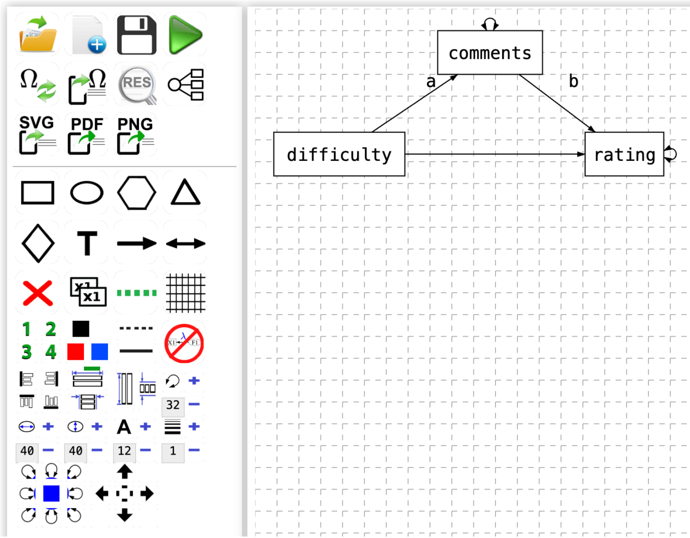
- We implemented the Sentence Transformers (a.k.a. SBERT) from `https://sbert.net/` with the pretrained models on Hugging Face.
- The embedded data are saved into an R dataset.
- Can be painfully slow on our current server ~ 20 minutes for about 500 short texts.

Sentiment analysis based on text embedding



- The R package sentiment.ai is used to get the text sentiment.

Example: Mediation analysis



How to use

AI based text sentiment

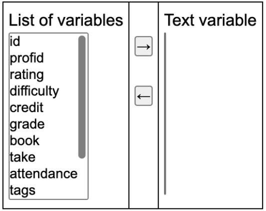
Analysis Menu

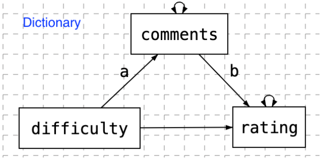
List of variables		Text variable
id profid rating difficulty credit grade book take attendance tags	→ ←	

How to use

AI based text sentiment

Analysis Menu

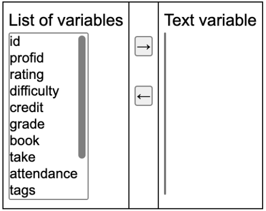


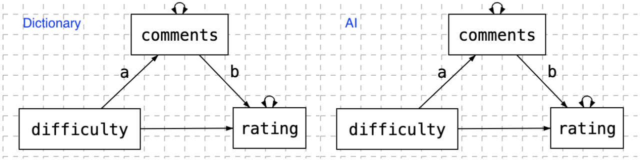


How to use

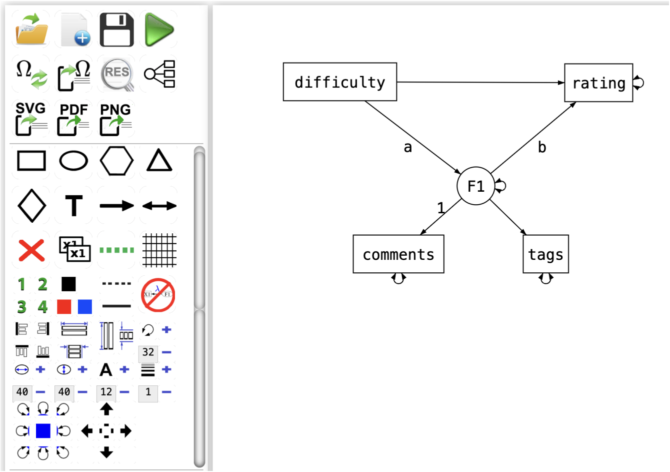
AI based text sentiment

Analysis Menu

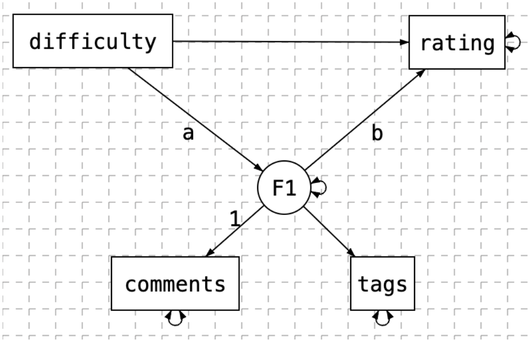




Example: Factor model



How to use



Programming interface (currently disabled)

SEM-text : textanalysis.R

Save | Run R | History description | Save a copy

Run on Tue Jul 09 2024 16:16:31 GMT-0400 (Eastern Daylight Time) Click to see the R output.

```r
## File textanalysis.R created by bigsem.org

## load the R package BigSEM
library(BigSEM)

## read in the SET data
setdata <- read.csv('prof1000.csv')

## specify the model
model <- 'comments ~ a*difficulty
          rating ~ difficulty + b*comments
          ab := a*b
          '

## estimate the model
res <- sem.text.ai(model = model, data = setdata, text_var = 'comments')

summary(res$estimates)
```

```
> summary(res$estimates)
lavaan 0.6-18 ended normally after 2 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         9

  Number of observations                         38240
  Number of missing patterns                         1

Model Test User Model:

  Test statistic                                 0.000
  Degrees of freedom                                 0

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Observed
  Observed information based on                Hessian

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  comments_ai ~
    difficulty (a)   -0.191    0.002  -86.944    0.000
  rating ~
    difficulty       -0.225    0.004  -58.357    0.000
    comments_a (b)    1.532    0.008  187.509    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .comments_ai       0.717    0.007  101.754    0.000
   .rating            4.197    0.013  330.939    0.000
    difficulty        2.928    0.007  445.625    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .comments_ai       0.306    0.002  138.275    0.000
   .rating            0.781    0.006  138.275    0.000
    difficulty        1.651    0.012  138.275    0.000

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
    ab               -0.293    0.004  -78.877    0.000

>
> proc.time()
    user   system  elapsed
1113.965   57.675  358.577
```

Summary and discussion

- An immense volume of textual data exists.
- Many new methods are available to automate the process of text data.
- However, text data are still under-utilize in research.
- We developed methods to use text data in SEM
  - Making the machine learning and AI methods more interpretable
  - Making the utilization of the text information easily possible
- To ease the use of text data for social scientists, we have develop the BigSEM app.
  - It can quantify text data using different methods.
  - It can directly use such information in SEM models.
  - It allows the convenient specification of a model.
  - It works online.
- It will open the opportunity for creative applications.
- Future directions
  - Better methods.
  - R package and online app development

Acknowledgments

- Ke-Hai Yuan, Lijuan Wang, Lingbo Tong, Austin Wyman, Tyler Wilcox, Anna Krush
- Institute of Education Sciences (R305D210023)
- Notre Dame Global

We need your feedback!

- We need your feedback to improve our software programs.
- If you can fill out our survey here: https://forms.gle/ecExNjimzPonQedE7, you can get a $25 Amazon gift card. Workshop participants only (first 20).

Q & A

- For more information
  - Zhiyong Zhang (zzhang4@nd.edu)
  - Website: `http://bigdatalab.nd.edu`

Thank you!

References I

Bailey, J. (2008). First steps in qualitative data analysis: transcribing. *Family practice*, 25(2), 127–131.

Bentler, P. M. & Weeks, D. G. (1980). Linear structural equations with latent variables. *Psychometrika*, 45, 289–308.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan), 993–1022.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877–1901.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y.-H., Strope, B., & Kurzweil, R. (2018). Universal sentence encoder.

Deng, L. & Liu, Y. (2018). *Deep learning in natural language processing*. Springer.

References II

Hu, M. & Liu, B. (2004a). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 168–177).: ACM.

Hu, M. & Liu, B. (2004b). Mining opinion features in customer reviews. In *AAAI*, volume 4 (pp. 755–760).

Jockers, M. L. (2017). Syuzhet: Extract sentiment and plot arcs from text. Retrieved from https://github.com/mjockers/syuzhet.

Mohammad, S. M. & Turney, P. D. (2010). Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text* (pp. 26–34).: Association for Computational Linguistics.

Nielsen, F. Å. (2011). Afinn.

Oppenheim, A. N. (2000). Questionnaire design, interviewing and attitude measurement.

References III

Perone, C. S., Silveira, R., & Paula, T. S. (2018). Evaluation of sentence embeddings in downstream and linguistic probing tasks.

Qu, W. & Zhang, Z. (2020). An application of aspect-based sentiment analysis on teaching evaluation. In Z. Zhang, K.-H. Yuan, Y. Wen, & J. Tang (Eds.), *New Developments in Data Science and Data Analytics: Proceedings of the 2019 Meeting of the International Society for Data Science and Analytics* (pp. 89–104). Granger, IN: ISDSA Press.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*.

Rohrer, J. M., Brümmer, M., Schmukle, S. C., Goebel, J., & Wagner, G. G. (2017). " what else are you worried about?"–integrating textual responses into quantitative social science research. *PloS one*, 12(7), e0182156.

Wilcox, K. T., Jacobucci, R., Zhang, Z., & Ammerman, B. A. (2023). Supervised latent dirichlet allocation with covariates: A bayesian structural and measurement model of text and covariates. *Psychological Methods*, 28(5), 1178–1206.