Lecture 28:
Word Embeddings

Big Data and Machine Learning for Applied Economics
Econ 4676

Ignacio Sarmiento-Barbieri

Universidad de los Andes

November 18, 2021

Recap: Text as Data

- Factor Models:
  - Unsupervized: PCA
  - Supervized:
    - PCR
    - PLS (MR)

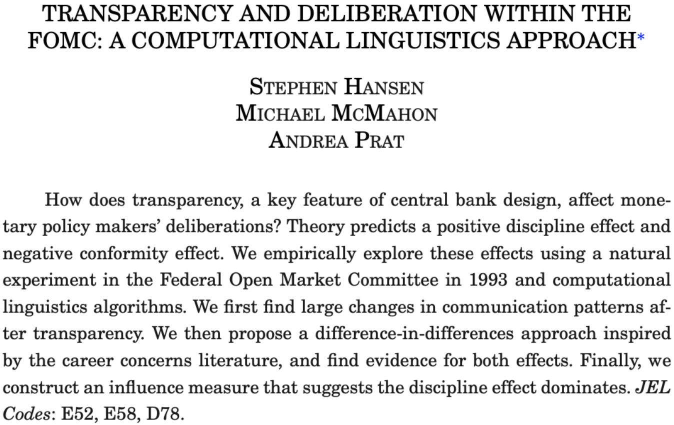
Agenda

1. Presentation Rafael
2. Latent Dirichlet Allocation
3. Word Embeddings
4. Word Embedding: Demo
5. Review & Next Steps
6. Further Readings

Latent Dirichlet Allocation

- The approach of using PCA to factorize text was common before the 2000s.
- Versions of this algorithm were referred to under the label latent semantic analysis.
- However, this changed with the introduction of topic modeling, also known as Latent Dirichlet Allocation (LDA), by Blei et al. in 2003.
- These authors pointed out that the squared error loss (i.e., Gaussian model) implied by PCA is inappropriate for analysis of sparse word-count data.

Latent Dirichlet Allocation

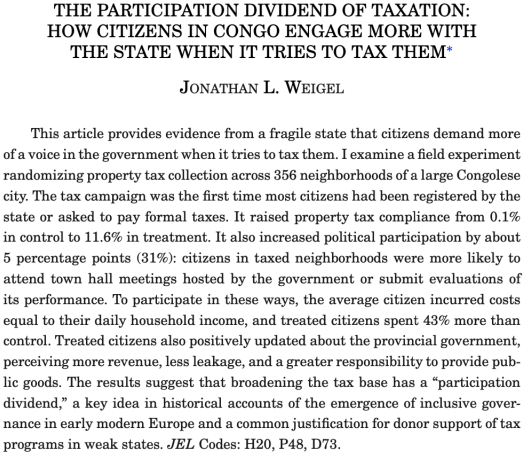


TRANSPARENCY AND DELIBERATION WITHIN THE FOMC: A COMPUTATIONAL LINGUISTICS APPROACH[*]

STEPHEN HANSEN
MICHAEL MCMAHON
ANDREA PRAT

How does transparency, a key feature of central bank design, affect monetary policy makers' deliberations? Theory predicts a positive discipline effect and negative conformity effect. We empirically explore these effects using a natural experiment in the Federal Open Market Committee in 1993 and computational linguistics algorithms. We first find large changes in communication patterns after transparency. We then propose a difference-in-differences approach inspired by the career concerns literature, and find evidence for both effects. Finally, we construct an influence measure that suggests the discipline effect dominates. *JEL Codes*: E52, E58, D78.

Latent Dirichlet Allocation



THE PARTICIPATION DIVIDEND OF TAXATION: HOW CITIZENS IN CONGO ENGAGE MORE WITH THE STATE WHEN IT TRIES TO TAX THEM*

JONATHAN L. WEIGEL

This article provides evidence from a fragile state that citizens demand more of a voice in the government when it tries to tax them. I examine a field experiment randomizing property tax collection across 356 neighborhoods of a large Congolese city. The tax campaign was the first time most citizens had been registered by the state or asked to pay formal taxes. It raised property tax compliance from 0.1% in control to 11.6% in treatment. It also increased political participation by about 5 percentage points (31%): citizens in taxed neighborhoods were more likely to attend town hall meetings hosted by the government or submit evaluations of its performance. To participate in these ways, the average citizen incurred costs equal to their daily household income, and treated citizens spent 43% more than control. Treated citizens also positively updated about the provincial government, perceiving more revenue, less leakage, and a greater responsibility to provide public goods. The results suggest that broadening the tax base has a "participation dividend," a key idea in historical accounts of the emergence of inclusive governance in early modern Europe and a common justification for donor support of tax programs in weak states. *JEL* Codes: H20, P48, D73.

Latent Dirichlet Allocation

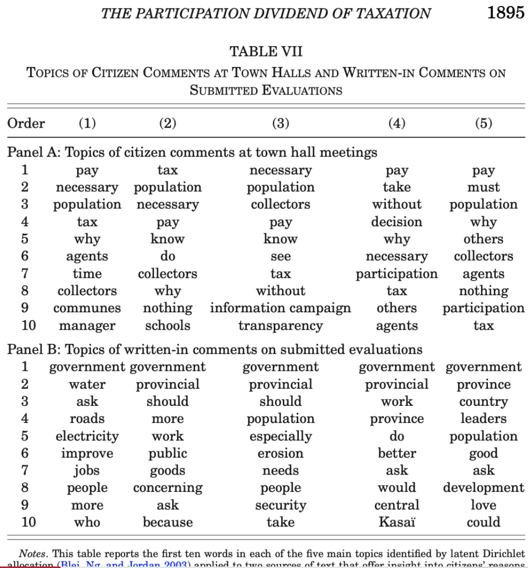


TABLE VII

Topics of Citizen Comments at Town Halls and Written-in Comments on Submitted Evaluations

Order	(1)	(2)	(3)	(4)	(5)
Panel A: Topics of citizen comments at town hall meetings					
1	pay	tax	necessary	pay	pay
2	necessary	population	population	take	must
3	population	necessary	collectors	without	population
4	tax	pay	pay	decision	why
5	why	know	know	why	others
6	agents	do	see	necessary	collectors
7	time	collectors	tax	participation	agents
8	collectors	why	without	tax	nothing
9	communes	nothing	information campaign	others	participation
10	manager	schools	transparency	agents	tax
Panel B: Topics of written-in comments on submitted evaluations					
1	government	government	government	government	government
2	water	provincial	provincial	provincial	province
3	ask	should	should	work	country
4	roads	more	population	province	leaders
5	electricity	work	especially	do	population
6	improve	public	erosion	better	good
7	jobs	goods	needs	ask	ask
8	people	concerning	people	would	development
9	more	ask	security	central	love
10	who	because	take	Kasaï	could

Notes. This table reports the first ten words in each of the five main topics identified by latent Dirichlet allocation (Blei, Ng, and Jordan 2003) applied to two sources of text that offer insight into citizens' reasons

Latent Dirichlet Allocation

- Blei et al. proposed you take the bag-of-words representation seriously and model token counts as realizations from a multinomial distribution.
- Topic models are built on a simple document generation process:
  - For each word, pick a "topic" k. This topic is defined through a probability vector over words, say, θ_k with probability θ_{kj} for each word j.
  - Then draw the word according to the probabilities encoded in θ_k .
- After doing this over and over for each word in the document, you have proportion ω_{i1} from topic 1, ω_{i2} from topic 2, and so on.

Latent Dirichlet Allocation

- This basic generation process implies that the full vector of word counts, x_i, has a multinomial distribution:

$$x_i \sim MN(\omega_{i1}\theta_1 + \cdots + \omega_{iK}\theta_K, m_i) \tag{1}$$

- where $m_i = \sum_j x_{ij}$ is the total document length and, for example,
- the probability of word j in document i will be $\sum_k \omega_{ik}\theta_{kj}$

Latent Dirichlet Allocation vs PCA

- Recall our PC model:

$$E(x_i) = \delta_{i1}F_1 + \cdots + \delta_{iK}F_K \tag{2}$$

- The analogous topic model representation, implied by the above equation, is

$$E(\frac{x_i}{m_i}) = \omega_{i1}\theta_1 + \cdots + \omega_{iK}\theta_K \tag{3}$$

- such that topic score ω_{ik} is like PC score δ_{ik} and
- θ_k topic probabilities are like rotations F_k.
- The distinction is that the multinomial in implies a different loss function (from a multinomial) rather than the sums of squared errors that PCA minimizes.
- Note that we condition on document length here so that topics are driven by relative rather than absolute term usage.

Word Embeddings

Word Embedding

- This is a “new” method that have come out of work in deep learning.
- Word embedding was originally motivated as a technique for dimension reduction on the inputs to a deep neural network.
- However, it imposes a spatial structure on words,
  - Allowing to get meanings from distance among words
  - Consider the algebra behind combinations of words in documents.

Word Embedding

- In the original deep learning context, embedding layers replace each word with a vector value
  - for example, hotdog becomes the location [1,–5, 0.25] in a three-dimensional embedding space
- Compare this to the standard bag-of-words representation, where hotdog would be represented as a binary vector that is as long as there are words in the vocabulary, say, p.
- This binary vector will have p–1 zeros and a one in the hotdog dimension.
- The word embedding has translated the language representation from a large binary space to a smaller real-valued (and much richer) space.

Word Embedding

- There are a variety of different embedding algorithms—as many as there are different architectures for deep neural networks.
- The most common and general embeddings are built around word co-occurrence matrices.
- This includes the popular `Glove` and `Word2Vec` frameworks.
- What is co-occurrence?
  - Two words co-occur if they appear within the same sentence and within b words of each other. Where b is the "window size"
  - For a vocabulary size p, this leads to a sparse p × p co-occurrence matrix where each [i, j] entry is the number of times that words i and j co-occur. Call this matrix C.
  - A word embedding algorithm seeks to approximate C as the product of two lower-dimensional matrices

Word Embedding

- A word embedding algorithm seeks to approximate C as the product of two lower-dimensional matrices

$$C \approx UV' \tag{4}$$

- Here, U and V are each $p \times K$ dimensional dense and real valued matrices.
- K is the dimension of the embedding space; hence, $K << p$ and both U and V are very tall and thin matrices.
- Each row of U and of V, u_j and v_j is then a K-dimensional embedding of the jth word.
- The implication is that these embeddings summarize the meaning of words as their inner product defines how much you expect them to co-occur.

  Recall that the inner product is a standard measure of distance in linear algebra (e.g. $e'e$)

Word Embedding

- One way to find U and V is to solve $C \approx UV'$ through the singular value decomposition (SVD).

- These locations were originally viewed as an intermediate output—as a processing step for inputs to a deep neural network.

- However, social scientists and linguists have discovered that the space of word locations contains rich information about the language of the documents used to train the embedding.

Word Embedding

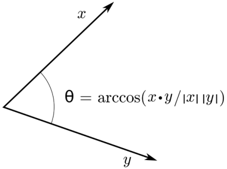
- Word embeddings preserve semantic relationships.
  - Words with similar meaning have similar representations.
  - Dimensions induced by word differences can be used to identify cultural concepts
- For example, the vector difference `man - woman` isolates a gender dimension in the space.
- The dimensions are useful because they produce quantitative measures of similarity between the associated concepts and specific words in the corpus.

Word Embedding

- In this case, we can understand the gender connotation of a given word by taking the cosine of the angle between the vector representation of the word and the differenced vector representing the gender dimension

- This is because the cosine of the angle, can be interpreted as a similarity measure.

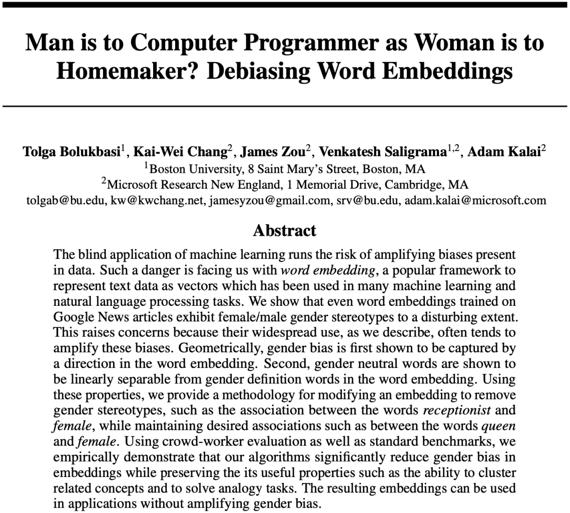
- The similarity ranges from -1 meaning exactly opposite, to 1 meaning exactly the same, with 0 indicating orthogonality or decorrelation, while in-between values indicate intermediate similarity or dissimilarity.



Word Embedding

- Words with male connotations – e.g. male first names – are going to be positively correlated with `man - woman`.
- Female words, in turn, will be negatively correlated with the dimension.
- This framework provides an intuitive approach to measuring stereotypical associations in a given corpus.
- Bolukbasi et al (2016) is a nice example

Word Embedding: Example 1



Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings

Tolga Bolukbasi[1], **Kai-Wei Chang**[2], **James Zou**[2], **Venkatesh Saligrama**[1,2], **Adam Kalai**[2]
[1]Boston University, 8 Saint Mary's Street, Boston, MA
[2]Microsoft Research New England, 1 Memorial Drive, Cambridge, MA
tolgab@bu.edu, kw@kwchang.net, jamesyzou@gmail.com, srv@bu.edu, adam.kalai@microsoft.com

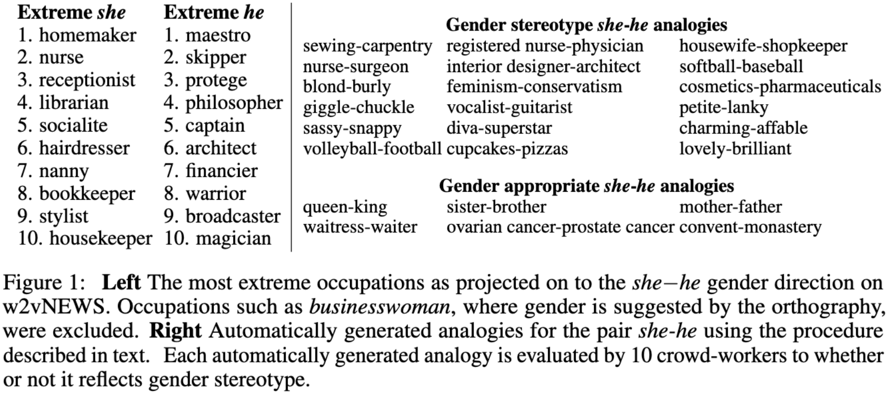
Abstract

The blind application of machine learning runs the risk of amplifying biases present in data. Such a danger is facing us with *word embedding*, a popular framework to represent text data as vectors which has been used in many machine learning and natural language processing tasks. We show that even word embeddings trained on Google News articles exhibit female/male gender stereotypes to a disturbing extent. This raises concerns because their widespread use, as we describe, often tends to amplify these biases. Geometrically, gender bias is first shown to be captured by a direction in the word embedding. Second, gender neutral words are shown to be linearly separable from gender definition words in the word embedding. Using these properties, we provide a methodology for modifying an embedding to remove gender stereotypes, such as the association between the words *receptionist* and *female*, while maintaining desired associations such as between the words *queen* and *female*. Using crowd-worker evaluation as well as standard benchmarks, we empirically demonstrate that our algorithms significantly reduce gender bias in embeddings while preserving the its useful properties such as the ability to cluster related concepts and to solve analogy tasks. The resulting embeddings can be used in applications without amplifying gender bias.

Word Embedding: Example 1

- They trained a standard word2vec embedding algorithm on the Google News corpora of news articles.
- Then look at the differences between established gender words (for example, the vector for man minus the vector for woman, or father minus mother) to establish an axis in the embedding space that spans from masculinity to femininity.
- They then calculate the location along this axis for a large number of terms that should be gender-neutral.
- The embedding space has learned—from how the words are used in news articles—that these professions are stereotypically viewed as female and male occupations.

Word Embedding: Example 1

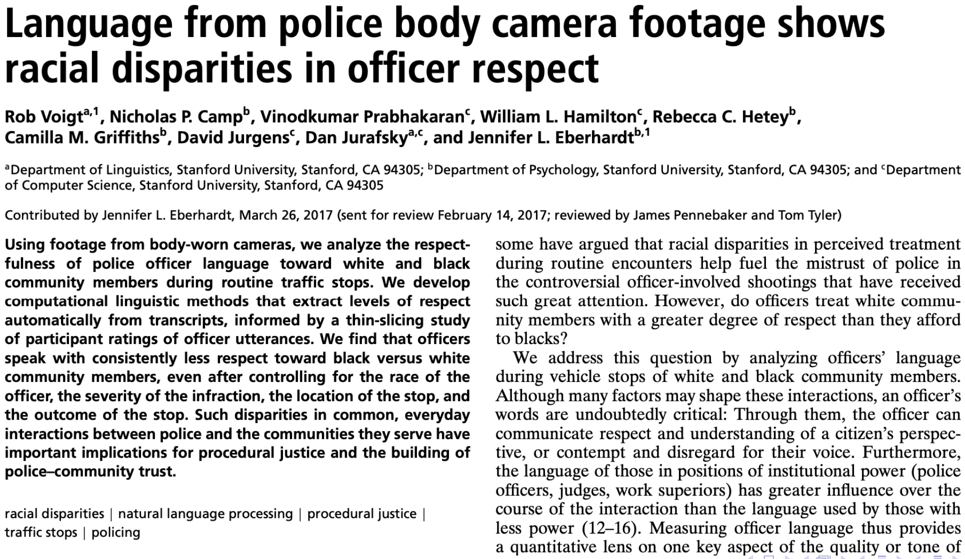


Extreme *she*	**Extreme *he***
1. homemaker	1. maestro
2. nurse	2. skipper
3. receptionist	3. protege
4. librarian	4. philosopher
5. socialite	5. captain
6. hairdresser	6. architect
7. nanny	7. financier
8. bookkeeper	8. warrior
9. stylist	9. broadcaster
10. housekeeper	10. magician

	Gender stereotype *she-he* analogies	
sewing-carpentry	registered nurse-physician	housewife-shopkeeper
nurse-surgeon	interior designer-architect	softball-baseball
blond-burly	feminism-conservatism	cosmetics-pharmaceuticals
giggle-chuckle	vocalist-guitarist	petite-lanky
sassy-snappy	diva-superstar	charming-affable
volleyball-football	cupcakes-pizzas	lovely-brilliant
	Gender appropriate *she-he* analogies	
queen-king	sister-brother	mother-father
waitress-waiter	ovarian cancer-prostate cancer	convent-monastery

Figure 1: **Left** The most extreme occupations as projected on to the *she*−*he* gender direction on w2vNEWS. Occupations such as *businesswoman*, where gender is suggested by the orthography, were excluded. **Right** Automatically generated analogies for the pair *she-he* using the procedure described in text. Each automatically generated analogy is evaluated by 10 crowd-workers to whether or not it reflects gender stereotype.

Word Embedding: Example 2



Language from police body camera footage shows racial disparities in officer respect

Rob Voigt[a,1], Nicholas P. Camp[b], Vinodkumar Prabhakaran[c], William L. Hamilton[c], Rebecca C. Hetey[b], Camilla M. Griffiths[b], David Jurgens[c], Dan Jurafsky[a,c], and Jennifer L. Eberhardt[b,1]

[a]Department of Linguistics, Stanford University, Stanford, CA 94305; [b]Department of Psychology, Stanford University, Stanford, CA 94305; and [c]Department of Computer Science, Stanford University, Stanford, CA 94305

Contributed by Jennifer L. Eberhardt, March 26, 2017 (sent for review February 14, 2017; reviewed by James Pennebaker and Tom Tyler)

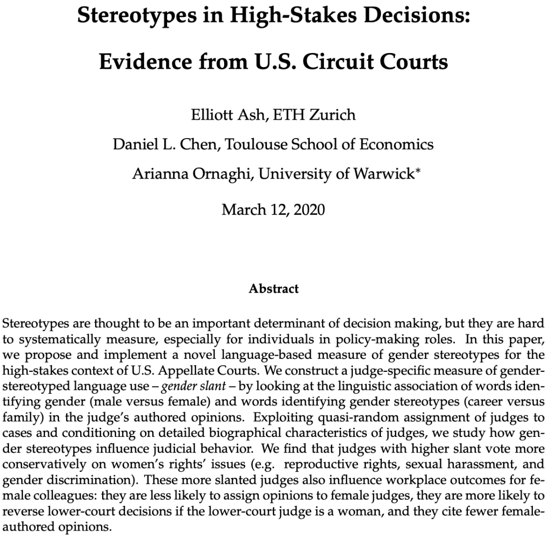
Using footage from body-worn cameras, we analyze the respectfulness of police officer language toward white and black community members during routine traffic stops. We develop computational linguistic methods that extract levels of respect automatically from transcripts, informed by a thin-slicing study of participant ratings of officer utterances. We find that officers speak with consistently less respect toward black versus white community members, even after controlling for the race of the officer, the severity of the infraction, the location of the stop, and the outcome of the stop. Such disparities in common, everyday interactions between police and the communities they serve have important implications for procedural justice and the building of police–community trust.

racial disparities | natural language processing | procedural justice | traffic stops | policing

some have argued that racial disparities in perceived treatment during routine encounters help fuel the mistrust of police in the controversial officer-involved shootings that have received such great attention. However, do officers treat white community members with a greater degree of respect than they afford to blacks?

We address this question by analyzing officers' language during vehicle stops of white and black community members. Although many factors may shape these interactions, an officer's words are undoubtedly critical: Through them, the officer can communicate respect and understanding of a citizen's perspective, or contempt and disregard for their voice. Furthermore, the language of those in positions of institutional power (police officers, judges, work superiors) has greater influence over the course of the interaction than the language used by those with less power (12–16). Measuring officer language thus provides a quantitative lens on one key aspect of the quality or tone of

Word Embedding: Example 3



Stereotypes in High-Stakes Decisions:

Evidence from U.S. Circuit Courts

Elliott Ash, ETH Zurich

Daniel L. Chen, Toulouse School of Economics

Arianna Ornaghi, University of Warwick*

March 12, 2020

Abstract

Stereotypes are thought to be an important determinant of decision making, but they are hard to systematically measure, especially for individuals in policy-making roles. In this paper, we propose and implement a novel language-based measure of gender stereotypes for the high-stakes context of U.S. Appellate Courts. We construct a judge-specific measure of gender-stereotyped language use – *gender slant* – by looking at the linguistic association of words identifying gender (male versus female) and words identifying gender stereotypes (career versus family) in the judge's authored opinions. Exploiting quasi-random assignment of judges to cases and conditioning on detailed biographical characteristics of judges, we study how gender stereotypes influence judicial behavior. We find that judges with higher slant vote more conservatively on women's rights' issues (e.g. reproductive rights, sexual harassment, and gender discrimination). These more slanted judges also influence workplace outcomes for female colleagues: they are less likely to assign opinions to female judges, they are more likely to reverse lower-court decisions if the lower-court judge is a woman, and they cite fewer female-authored opinions.

Word Embedding: Demo

```r
library(text2vec)
load('shakes_words_df_4text2vec.RData')
head(shakes_words)
```

```
##                     id        word
## 1 A_Lover_s_Complaint          nor
## 2 A_Lover_s_Complaint        gives
## 3 A_Lover_s_Complaint           it
## 4 A_Lover_s_Complaint satisfaction
## 5 A_Lover_s_Complaint           to
```

```r
shakes_words_ls <- list(shakes_words$word)
it <- itoken(shakes_words_ls, progressbar = FALSE)
shakes_vocab <- create_vocabulary(it)
shakes_vocab <- prune_vocabulary(shakes_vocab, term_count_min= 5)
head(shakes_vocab)
```

```
## Number of docs: 1
## 0 stopwords:  ...
## ngram_min = 1; ngram_max = 1
## Vocabulary:
##          term term_count doc_count
## 1:     abbess          5         1
## 2: abilities          5         1
## 3: accessary          5         1
## 4:        ace          5         1
## 5:     adders          5         1
```

Word Embedding: Demo

- The next step is to create the token co-occurrence matrix (TCM).
- The definition of whether two words occur together is arbitrary.

```
# maps words to indices
vectorizer <- vocab_vectorizer(shakes_vocab)

# use window of 10 for context words
shakes_tcm <- create_tcm(it, vectorizer, skip_grams_window = 10)
```

- Now we are ready to create the word vectors based on the GloVe model.

```
glove <- GlobalVectors$new(rank = 50, x_max = 10)
shakes_wv_main = glove$fit_transform(shakes_tcm, n_iter = 10, convergence_tol = 0.01, n_threads = 8)
```

```
## INFO  [16:55:06.317] epoch 1, loss 0.1242
## INFO  [16:55:08.764] epoch 2, loss 0.0844
## INFO  [16:55:11.249] epoch 3, loss 0.0762
## INFO  [16:55:13.680] epoch 4, loss 0.0707
## INFO  [16:55:16.109] epoch 5, loss 0.0666
## INFO  [16:55:18.540] epoch 6, loss 0.0634
## INFO  [16:55:20.980] epoch 7, loss 0.0609
## INFO  [16:55:23.419] epoch 8, loss 0.0589
## INFO  [16:55:25.849] epoch 9, loss 0.0572
## INFO  [16:55:28.288] epoch 10, loss 0.0558
```

Word Embedding: Demo

```r
dim(shakes_wv_main)
```

```
## [1] 9094   50
```

```r
shakes_wv_context <- glove$components

dim(shakes_wv_context)
```

```
## [1]   50 9094
```

```r
# Either word-vectors matrices could work, but the developers of the technique
# suggest the sum/mean may work better
shakes_word_vectors <- shakes_wv_main + t(shakes_wv_context)

rom <- shakes_word_vectors["romeo", , drop = F]

cos_sim_rom <- sim2(x =shakes_word_vectors, y = rom, method = "cosine", norm = "l2")
# head(sort(cos_sim_rom[,1], decreasing <- T), 10)
```

```
##     romeo    juliet    tybalt     nurse  benvolio  banished
## 1.0000000 0.7712391 0.7575977 0.6697068 0.6517349 0.6436404
```

Word Embedding: Demo

```
test <- shakes_word_vectors["romeo", , drop = F] -
  shakes_word_vectors["mercutio", , drop = F] +
  shakes_word_vectors["nurse", , drop = F]

cos_sim_test <- sim2(x = shakes_word_vectors, y = test, method = "cosine", norm = "l2")
head(sort(cos_sim_test[,1], decreasing = T), 10)
```

```
##     nurse    juliet     romeo      lady    mother       bed         o      wife
## 0.8904362 0.7584004 0.7179267 0.6440354 0.6374490 0.5880860 0.5756074 0.5638571
##   capulet    dromio
## 0.5520459 0.5507196
```

Review & Next Steps

- LDA
- Word Embedding
- Word Embedding: Demo
- Next class: Deep Learning: Intro
- Questions? Questions about software?

Further Readings

- Bolukbasi, T., Chang, K. W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In Advances in neural information processing systems (pp. 4349-4357).
- Clark, M (2018). An Introduction to Text Processing and Analysis with R. https://m-clark.github.io/text-analysis-with-R/ Rstudio (2020). Tutorial TensorFlow https://tensorflow.rstudio.com/tutorials/beginners/basic-ml/tutorial_basic_classification/
- Taddy, M. (2019). Business data science: Combining machine learning and economics to optimize, automate, and accelerate business decisions. McGraw Hill Professional.
- Voigt, R., Camp, N. P., Prabhakaran, V., Hamilton, W. L., Hetey, R. C., Griffiths, C. M., ... & Eberhardt, J. L. (2017). Language from police body camera footage shows racial disparities in officer respect. Proceedings of the National Academy of Sciences, 114(25), 6521-6526.