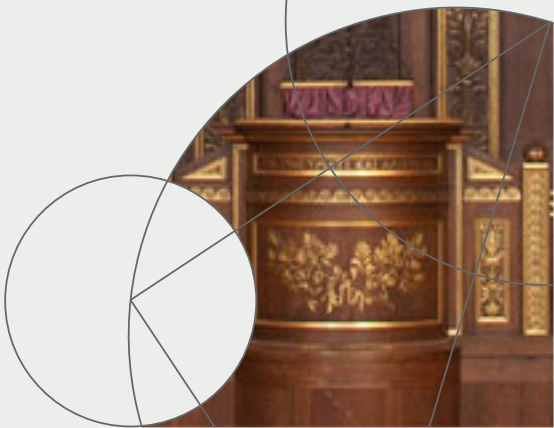
Word Embeddings for NLP

Rensheng Wang,
https://sit.instructure.com/courses/43729



Embedding vs. Word2Vec

- The concept of embeddings to be one of the most fascinating ideas in machine learning.
- If youve ever used Siri, Google Assistant, Alexa, Google Translate, or even smartphone keyboard with next-word prediction, then chances are youve benefitted from this idea that has become central to Natural Language Processing models.
- There has been quite a development over the last couple of decades in using embeddings for neural models (Recent developments include contextualized word embeddings leading to cutting-edge models like BERT and GPT2).
- Word2vec is a method to efficiently create word embeddings and has been around since 2013.
- In addition to its utility as a word-embedding method, some of its concepts have been shown to be effective in creating recommendation engines and making sense of sequential data even in commercial, non-language tasks.

Personality Embeddings: What are you like?

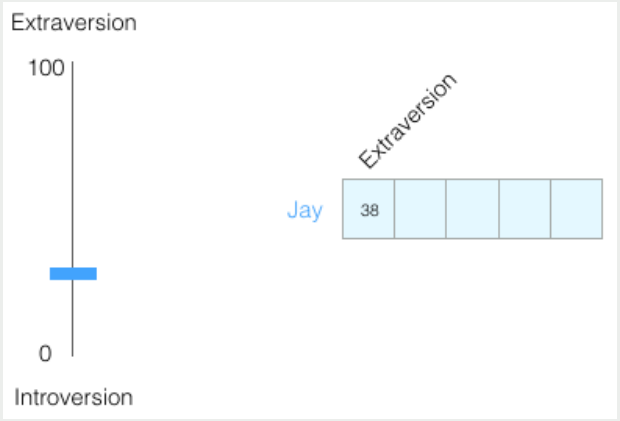
- On a scale of 0 to 100, how introverted/extraverted are you (where 0 is the most introverted, and 100 is the most extraverted)?
- Have you ever taken a personality test like MBTI or even better, the Big Five Personality Traits test?
- If you havent, these are tests that ask you a list of questions, then score you on a number of axes, introversion/extraversion being one of them.

Openness to experience	79 out of 100
Agreeableness	75 out of 100
Conscientiousness	42 out of 100
Negative emotionality	50 out of 100
Extraversion	58 out of 100

Example of the result of a Big Five Personality Trait test.

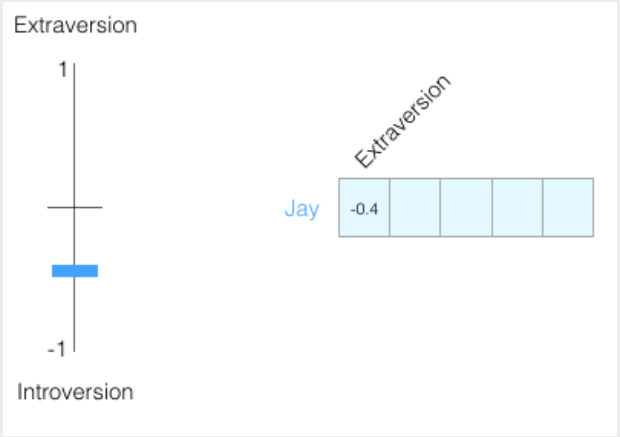
Personality Embeddings: What are you like?

- Imagine I have scored 38/100 as my introversion/extraversion score. we can plot that as:



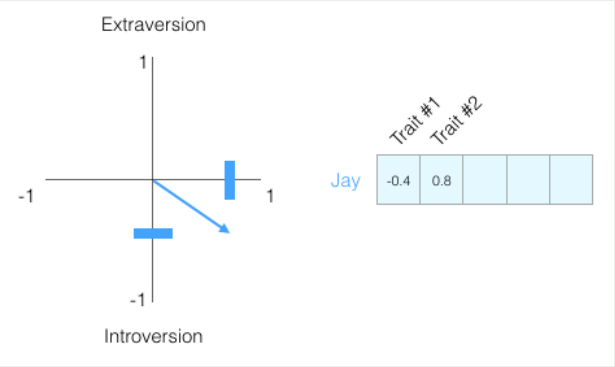
Personality Embeddings: What are you like?

- Lets switch the range to be from -1 to 1:



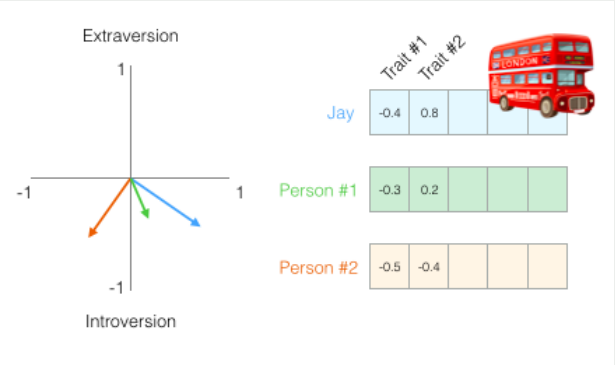
Personality Embeddings: What are you like?

- How well do you feel you know a person knowing only this one piece of information about them? Not much. People are complex. So lets add another dimension the score of one other trait from the test.



Personality Embeddings: What are you like?

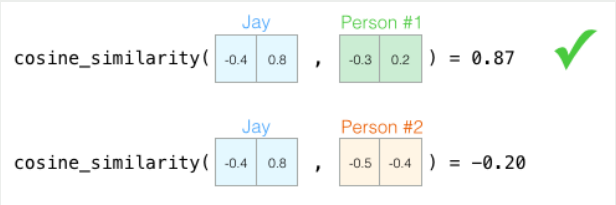
- We can now say that this vector partially represents my personality. The usefulness of such representation comes when you want to compare two other people to me. Say I get hit by a bus and I need to be replaced by someone with a similar personality. In the following figure, which of the two people is more similar to me?



→ which vector is similar to the other?

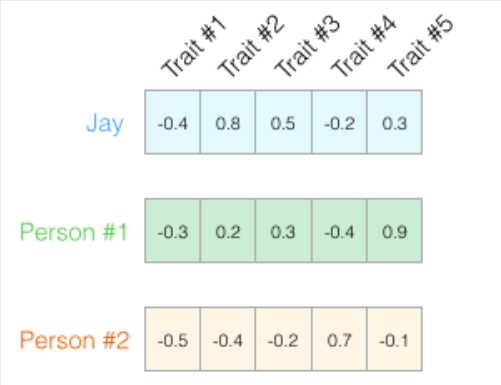
Personality Embeddings: Who is more silimar to you?

- When dealing with vectors, a common way to calculate a similarity score is `cosine_similarity`:



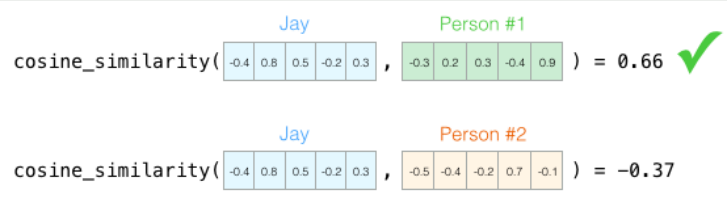
Personality Embeddings: Who is more silimar to you?

- Yet again, two dimensions arent enough to capture enough information about how different people are. Decades of psychology research have led to five major traits (and plenty of sub-traits). So lets use all five dimensions in our comparison:



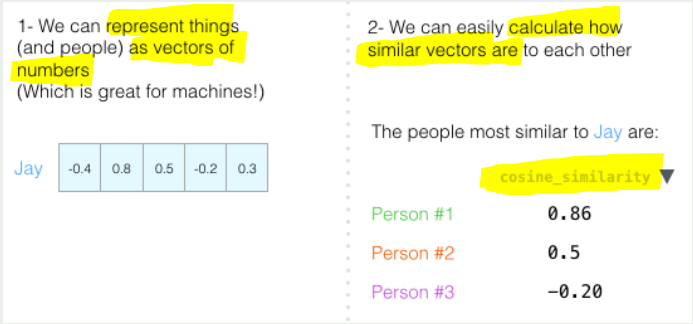
Personality Embeddings: Who is more silimar to you?

- Yet again, two dimensions arent enough to capture enough information about how different people are. Decades of psychology research have led to five major traits (and plenty of sub-traits). So lets use all five dimensions in our comparison:



Personality Embeddings: What we learn from them?

- At the end of this section, we come out with two central ideas:
  - We can represent people (and things) as vectors of numbers (which is great for machines!).
  - We can easily calculate how similar vectors are to each other.



Word Embeddings: What do they look like?

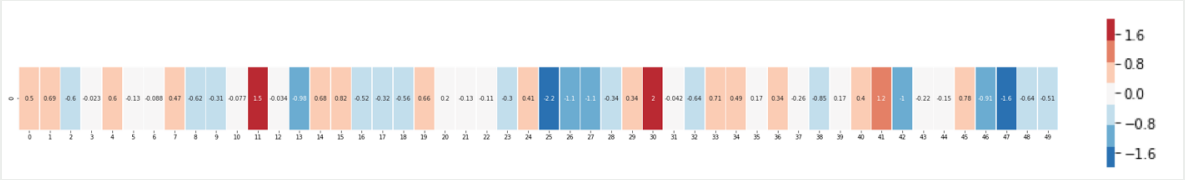
- With this understanding, we can proceed to look at trained word-vector examples (also called word embeddings) and start looking at some of their interesting properties.

  This is a word embedding for the word king (GloVe vector trained on Wikipedia):

```
[ 0.50451 , 0.68607 , -0.59517 , -0.022801, 0.60046 , -0.13498 , -0.08813
, 0.47377 , -0.61798 , -0.31012 , -0.076666, 1.493 , -0.034189, -0.98173 ,
0.68229 , 0.81722 , -0.51874 , -0.31503 , -0.55809 , 0.66421 , 0.1961 ,
-0.13495 , -0.11476 , -0.30344 , 0.41177 , -2.223 , -1.0756 , -1.0783 ,
-0.34354 , 0.33505 , 1.9927 , -0.04234 , -0.64319 , 0.71125 , 0.49159 ,
0.16754 , 0.34344 , -0.25663 , -0.8523 , 0.1661 , 0.40102 , 1.1685 ,
-1.0137 , -0.21585 , -0.15155 , 0.78321 , -0.91241 , -1.6106 , -0.64426 ,
-0.51042 ]
```

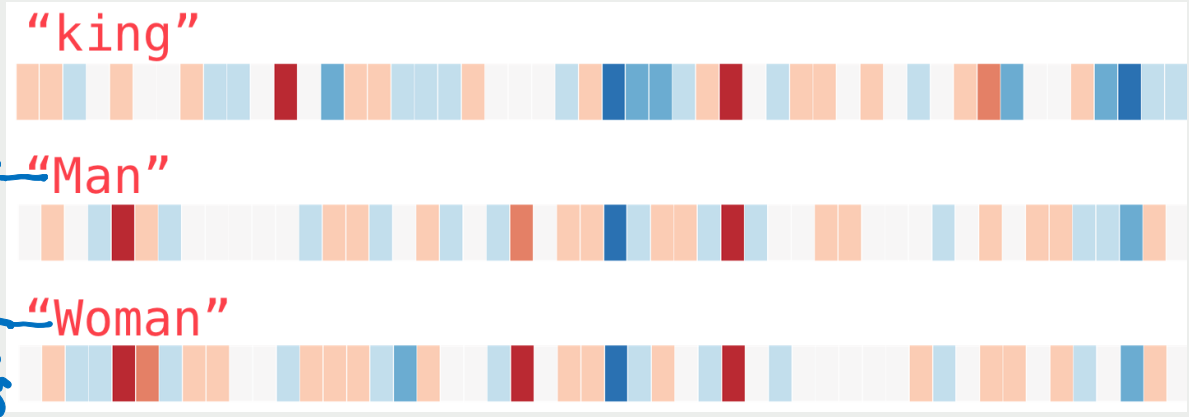
Word Embeddings: What "King" looks like?

- Lets color code the cells based on their values (red if theyre close to 2, white if theyre close to 0, blue if theyre close to -2):



"King" vs. Other Words

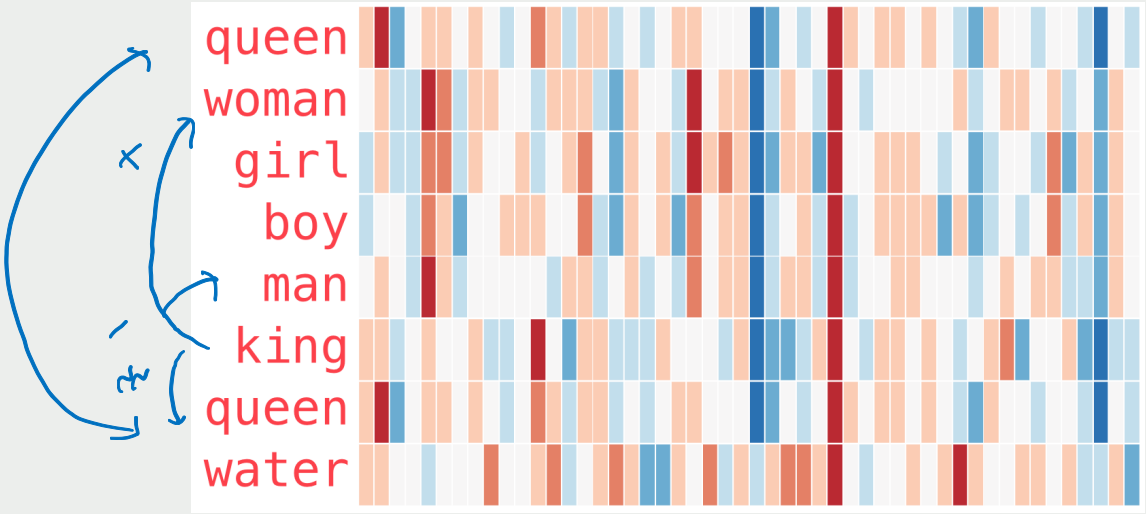
- Well proceed by ignoring the numbers and only looking at the colors to indicate the values of the cells. Lets now contrast "King" against other words:



the similarity in words can be seen visually through feature vectors

"King" vs. Other Words

- Heres another list of examples (compare by vertically scanning the columns looking for columns with similar colors):

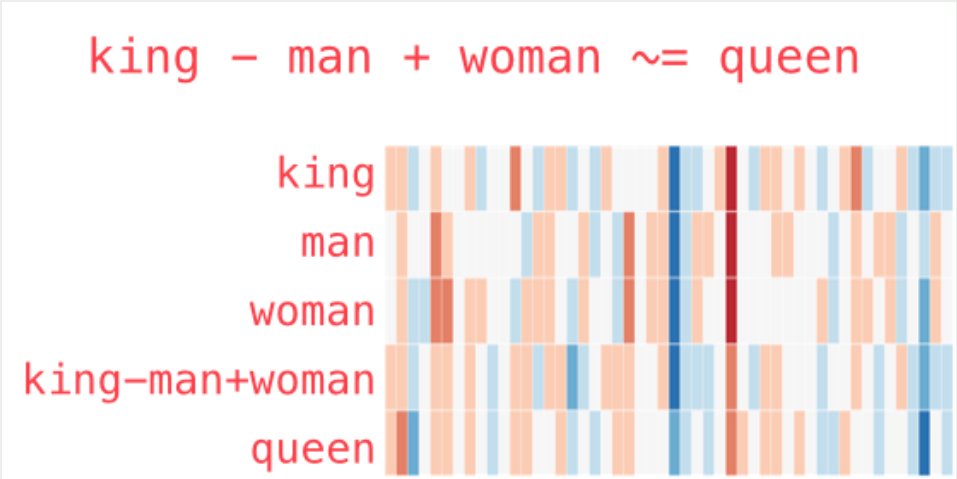


Word Embeddings

- Theres a straight red column through all of these different words. They are similar along that dimension (and we do not know what each dimensions codes for)
- You can see how "woman" and "girl" are similar to each other in a lot of places. The same with "man" and "boy" "boy" and "girl" also have places where they are similar to each other, but different from "woman" or "man". Could these be coding for a vague conception of youth? possible.
- All but the last word are words representing people. I added an object (water) to show the differences between categories. You can, for example, see that blue column going all the way down and stopping before the embedding for water.
- There are clear places where king and queen are similar to each other and distinct from all the others. Could these be coding for a vague concept of royalty?

Word Embeddings Analogies

- The famous examples that show an incredible property of embeddings is the concept of analogies. We can add and subtract word embeddings and arrive at interesting results. The most famous example is the formula: "king" - "man" + "woman":



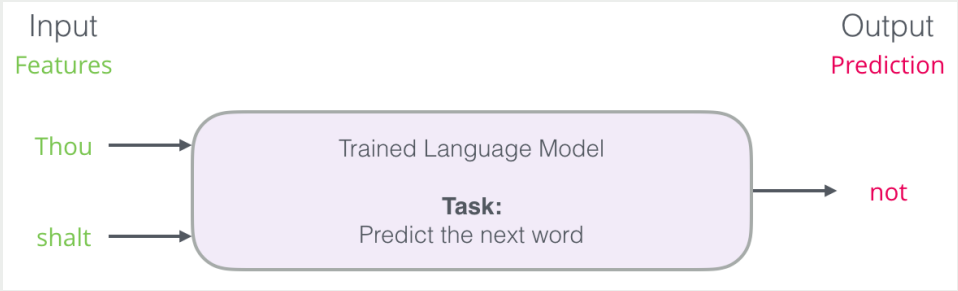
Language Modeling

- One of the best NLP examples would be the next-word prediction feature of a smartphone keyboard. Its a feature that billions of people use hundreds of times every day.

- Next-word prediction is a language model. A language model can take a list of words (lets say two words), and attempt to predict the word that follows them.

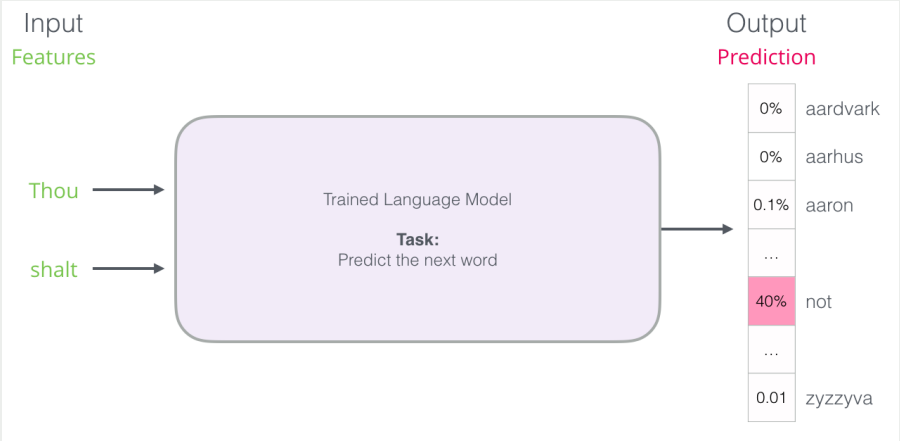
Language Modeling

- In the previous page, we can think of the model as one that took in these two words (thou shalt) and returned a list of suggestions ("not" being the one with the highest probability):
- We can think of the model as looking like this black box:



Language Modeling

- In practice, the model does not output only one word. It outputs a probability score for all the words it knows (the model's "vocabulary", from a few thousand to over a million words).
- The keyboard application then has to find the words with the highest scores, and present those to the user.



Language Modeling

- The first step is the most relevant for us as we discuss embeddings. One of the results of the training process was this matrix that contains an embedding for each word in our vocabulary.
- During prediction time, we just look up the embeddings of the input word, and use them to calculate the prediction: