SENTENCE VECTORS:

Sometimes successful attempts to capture meaning

Joel Adams, 10 March, 2020

Center for Spoken Language Understanding, Oregon Health & Science University

INTRODUCTION

- PhD candidate at the Center for Spoken Language Understanding, OHSU. Graduating in August.
- Previously worked in educational software.
  - Data analytics with SQL and Python
  - Evaluating collected metrics and correlation with long term measures of success.

OVERVIEW

- Vectors as a way to represent word meanings in a computationally tractable way
- What might we want to do with word vectors?
- But what about sentences?
- Some examples from my own research.
- Resources for building and applying sentence vectors in Python.
- Will NOT discuss machine learning except at a high level.

WHAT DOES MEANING MEAN?

- In Natural Language Processing (NLP) it is useful to have a way to approximate meaning.
  - How similar are these two words? Sentences? Documents?
- Which of these words have more similar meaning in everyday usage?
  - "cat", "feline", "dog", "computer"
- How would a computer know?

DISTRIBUTIONAL SEMANTICS

- “**Cat**”, “**Feline**”, “**Dog**”, “**Computer**” and two reasonable sentences.
  - “A **cat** has four legs and purrs”
  - “A **dog** has four legs.”
- **Distributional Hypothesis**: “A word is characterized by the company it keeps.” - JR Firth? Zellig Harris?
- How do we operationalize this?

CO-OCCURENCE MATRIX

	A	cat	has	four	legs	and	purrs	dog
A	0	1	2	2	2	1	1	1
cat	1	0	1	1	1	1	1	0
has	2	1	0	2	2	1	1	1
four	2	1	2	0	2	1	1	1
legs	2	1	2	2	0	1	1	1
and	1	1	1	1	1	0	1	0
purrs	1	1	1	1	1	1	0	0
dog	1	0	1	1	1	0	0	0

- Our Corpus:
  - “A **cat** has four legs and purrs”
  - “A **dog** has four legs.”
- Context is treated as “Bag of Words”

VECTOR SPACE MODEL (VSM)

- How can we operationalize the distributional hypothesis?
  - Vectors! Capturing the context that each word appears in.
  - In a VSM, words with similar meaning should be near each other in vectors space.
- Measure the angle between the vectors
- Lots of extensions:
  - weighing the co-occurences (e.g. tf-idf)
  - dimension reduction (e.g. Latent Semantic Analysis)

WORD EMBEDDINGS

- But what about word embeddings? Are they different?
  - Just a different application for the same idea.
  - Fixed size input for NLP tasks.
  - Represents words in a way that captures some sort of meaning.

WORD2VEC

- In 2013, Thomas Mikolov, et al presented word2vec
- Part of a family methods that use neural networks to directly estimate dense word representations without needing to first compute a full term-context matrix.
- A neural network is trained to use each word in a training corpus to predict its context. “__ cat __”
- Successes in several applications.
- Interesting linear compositionality: <madrid> - <spain> + <france> is clossest to <Paris>

BUT WHAT ABOUT SENTENCES?

- Simple solution: Averaging over Bag of Vectors
  - Seems reasonable given word2vec composition.
  - Does it work?
    - Surprisingly well!
  - Limitation: word order lost.
  - Feels a bit unprincipled. What does it mean if we add several sentences?

MORE SOPHISTICATED SENTENCE MODELS

- Doc2Vec (Le and Mikolov, 2014)
  - Word2Vec extended to include an additional variable for “paragraph”, “sentence” or “document”
- Skip Thought (Kiros, et al 2015)
  - tries to reconstruct the surrounding sentences from surrounded one

EVALUATING SENTENCE VECTORS

What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties

Alexis Conneau
Facebook AI Research
Université Le Mans
aconneau@fb.com

German Kruszewski
Facebook AI Research
germank@fb.com

Guillaume Lample
Facebook AI Research
Sorbonne Universités
glample@fb.com

- Downstream tasks
    - Paraphrase Detection
    - Entailment (whether on sentences requires the truth of another)
- Realization that these tasks require complex inference
    - Not always clear which information captured in the vector is useful.

EVALUATION: PROBING TASKS

- Sentence embeddings can capture a large number of features
  - Surface features: sentences length, word content
  - Syntactic information: reversed word order
  - Semantic information: verb tense, odd man out
- The information represented in the sentence vector depends on several factors
  - Architecture, Training task, Evaluation task

ANOTHER SENTENCE EMBEDDING

- Bidirectional Encoder Representations from Transformers (BERT), (Devlin et al 2018)
- Pre-trained on two tasks:
  - Randomly masks words in the sentence and then it tries to predict them
  - Next sentence prediction: Given two sentences, predicts if the second one is the following sentence in the original data set.
- Allows for a sentence representation that is sensitive to word order.
- Fine-tunable. Can adapt it to your own task.

MY OWN RESEARCH

- My research is funded by a grant focused on Language and Autism.
- Currently modeling the flow of topics in a conversation from Autism Diagnostic Observation Schedule (ADOS).
  - Non-contingent speech.
  - “Tell me about your school day.” “My cat is also very large!”
- Requires a way to represent the sentences that includes some sense of their meaning.

SOME EXAMPLES OF APPLIED SENTENCE VECTORS

- ADOS conversations are partially scripted:
  - "What are some things that make you happy?"
  - "What makes you angry?"
  - Created an examiner "Topic Tagger" to identify when a variation of one of these sentences is used.
- Working on a model that can determine whether the subject's speech follows the examiner topics.

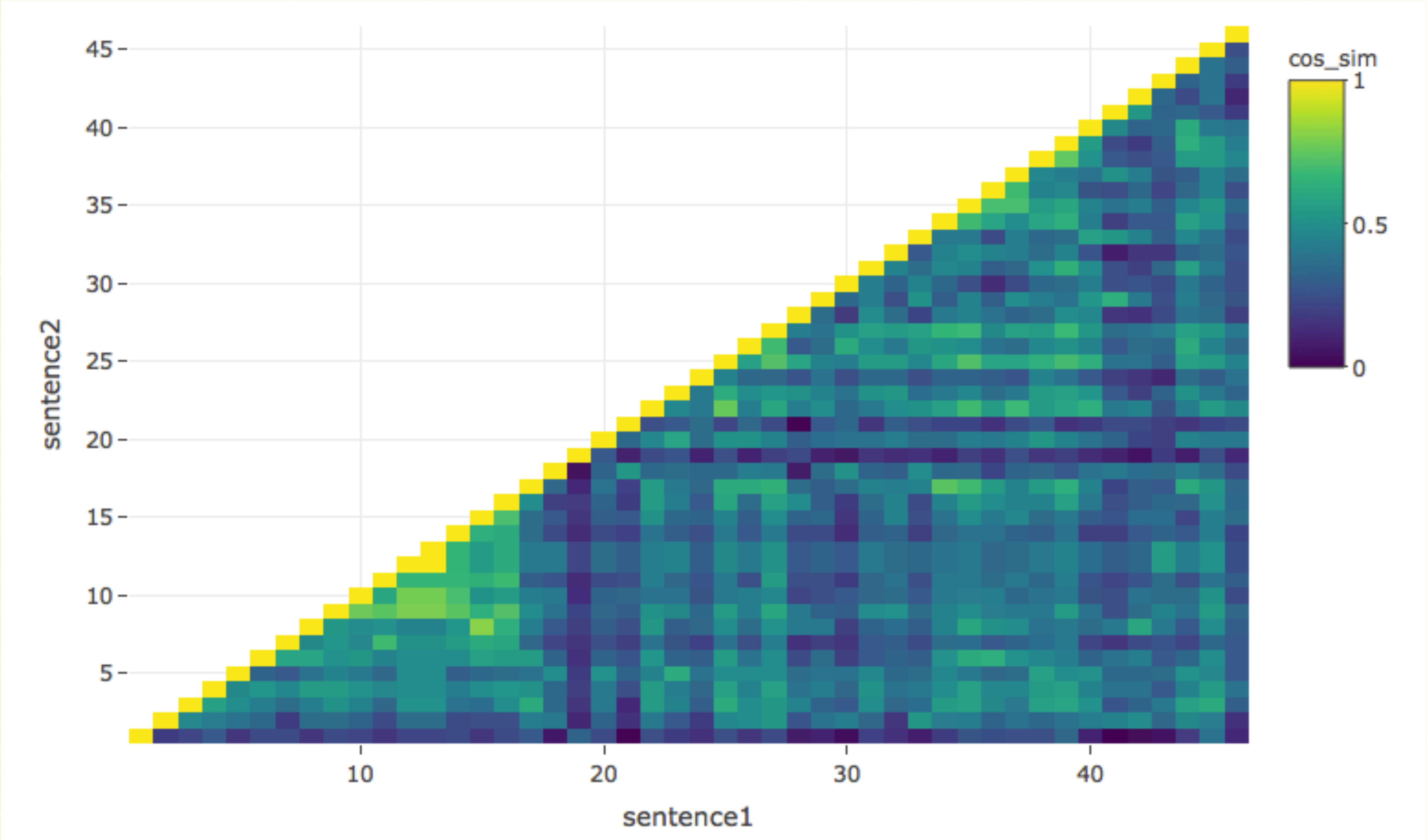
BERT APPLIED TO MY DATA

- For this discussion, I am using the Large Pre-trained BERT model
- Using second to last hidden layer, reduce mean pooling.
- Not fine tuned.
- What task could I use to fine tune it?

SIMILARITY TO SCRIPTED QUESTIONS

- **Scripted sentence**: "What do you like to do that makes you feel happy and cheerful?"
- **Positive Example**: "what are the things that you like to do that make you feel cheerful?" similarity: **.94**
- **Positive Example:** "She jumped up!" **.45**
- **Negative Example:** "what are the kinds of things that make you feel sad?" similarity: **.88**

EXAMPLES FROM SUBJECT SPEECH



RESOURCES IN PYTHON

- Word2Vec:
  - https://radimrehurek.com/gensim/
- Pre-trained models for BERT:
  - https://github.com/google-research/bert

WORKING WITH BERT

- Two fairly easy ways to get started working with BERT:
  - bert-as-a-service:
    - https://bert-as-service.readthedocs.io/en/latest/.
  - PyTorch via the huggingface transformers library:
    - https://huggingface.co/transformers/
    - Hands on guide on fine tuning and text classification:
    - https://towardsdatascience.com/https-medium-com-chaturangarajapakshe-text-classification-with-transformer-models-d370944b50ca

THANK YOU!

- Feel free to email with questions and comments.
  - adamjo@ohsu.edu