Natural Language Proces

Alex Olson

Outline

Intro to NLP & Embeddings

Outline

Intro to NLP & Embeddings

Attention, Transformers and GPT

Outline

Intro to NLP & Embeddings

Attention, Transformers and GPT

Applications of LLMs

What is NLP?

- Subfield of AI that focuses on reading, deciphering and produci

What is NLP?

- Subfield of AI that focuses on reading, deciphering and produci
- Combines computational linguistics (e.g. rule-based modelling
  statistical, ML, and deep learning approaches

What is NLP?

- Subfield of AI that focuses on reading, deciphering and produci
- Combines computational linguistics (e.g. rule-based modelling
  statistical, ML, and deep learning approaches
- Through NLP, machines can understand, analyze and generate
  are meaningful and contextually appropriate

What is NLP?

- Subfield of AI that focuses on reading, deciphering and produci
- Combines computational linguistics (e.g. rule-based modelling statistical, ML, and deep learning approaches
- Through NLP, machines can understand, analyze and generate are meaningful and contextually appropriate
- While LLMs have driven an explosion in interest, there are man which paved the way

Common NLP Tasks

- Text Classification, e.g. spam detection
- Named Entity Recognition, e.g. identifying people, places, orga
- Sentiment Analysis, e.g. positive or negative sentiment
- Machine Translation, e.g. Google Translate

Challenges in NLP

- Ambiguity: words can have multiple meanings

Challenges in NLP

- Ambiguity: words can have multiple meanings
- Context: the meaning of a word can change depending on the co

Challenges in NLP

- Ambiguity: words can have multiple meanings
- Context: the meaning of a word can change depending on the c
- Nuance: language is full of subtleties and nuances

Challenges in NLP

- Ambiguity: words can have multiple meanings
- Context: the meaning of a word can change depending on the co
- Nuance: language is full of subtleties and nuances
- Syntax vs Semantics: A phrase can be grammatical but nonsens
  ungrammatical but meaningful

Modern NLP

- LLMs have revolutionized NLP

Modern NLP

- LLMs have revolutionized NLP
- Unlike earlier rule-based or statistical models, LLMs learn to g
  training on large amounts of data

Modern NLP

- LLMs have revolutionized NLP
- Unlike earlier rule-based or statistical models, LLMs learn to ge
  training on large amounts of data
- They can generate text that is coherent and contextually approp

Modern NLP

- LLMs have revolutionized NLP
- Unlike earlier rule-based or statistical models, LLMs learn to ge
  training on large amounts of data
- They can generate text that is coherent and contextually appro
- They can be fine-tuned for specific tasks

The Encoder-Decoder View o
Learning

- Most modern deep learning architectures can be divided into a

The Encoder-Decoder View o
Learning

- Most modern deep learning architectures can be divided into a
- In the first step (feature extraction), a model encodes some inp
  representation

The Encoder-Decoder View o
Learning

- Most modern deep learning architectures can be divided into a
- In the first step (feature extraction), a model encodes some inp
  representation
- This *latent representation* is meaningful but not directly interp

The Encoder-Decoder View o Learning

- Most modern deep learning architectures can be divided into a
- In the first step (feature extraction), a model encodes some inp representation
- This *latent representation* is meaningful but not directly interp
- In the second step (decoding), the model generates an output b representation

The Encoder-Decoder View o
Learning

- Most modern deep learning architectures can be divided into a
- In the first step (feature extraction), a model encodes some inp
  representation
- This *latent representation* is meaningful but not directly interp
- In the second step (decoding), the model generates an output b
  representation
- We will first focus on the encoder part of the model

Embeddings

From Real to Symbolic

- Often, machine learning deals with *real-valued* input: data that
  or can be easily converted to a number and thus contains its ow
- Examples: pixel values in an image, audio samples in a sound fi
  readings
- But what if the input is a symbol?

Symbolic variable

- Text: characters, words, bigrams...
- Recommender Systems: item ids, user ids
- Any categorical descriptor: tags, movie genres, visite
  resume, product categories...

Symbolic variable

- Text: characters, words, bigrams...
- Recommender Systems: item ids, user ids
- Any categorical descriptor: tags, movie genres, visite
  resume, product categories...

Symbolic variable

- Text: characters, words, bigrams...
- Recommender Systems: item ids, user ids
- Any categorical descriptor: tags, movie genres, visite
  resume, product categories...

Symbolic variable

- Text: characters, words, bigrams...
- Recommender Systems: item ids, user ids
- Any categorical descriptor: tags, movie genres, visite
  resume, product categories...

Notation:

$$\text{Symbol } s \text{ in vocabulary } V$$

One-hot representation

$$onehot(\text{'salad'}) = [0, 0, 1, \ldots, 0] \in \{0, 1\}^{|V}$$

One-hot representation

$$onehot(\text{'salad'}) = [0, 0, 1, \ldots, 0] \in \{0, 1\}^{|V|}$$

- Sparse, discrete, large dimension $|V|$
- Each axis has a meaning
- Symbols are equidistant from each other:

euclidean distance = $\sqrt{2}$

Embedding

$$embedding(\text{'salad'}) = [3.28, -0.45, \ldots 7.1$$

Embedding

$$embedding(\text{'salad'}) = [3.28, -0.45, \ldots 7.1$$

- Continuous and dense
- Can represent a huge vocabulary in low dimension, typically: $d \in \{16, 32, \ldots, 4096\}$
- Axis have no meaning *a priori*
- Embedding metric can capture semantic distance

Embedding

$$embedding(\text{'salad'}) = [3.28, -0.45, \ldots 7.1$$

- Continuous and dense
- Can represent a huge vocabulary in low dimension, typically: $d \in \{16, 32, \ldots, 4096\}$
- Axis have no meaning *a priori*
- Embedding metric can capture semantic distance

Neural Networks compute transformations on continuous vecto

Implementation with Keras

Size of vocabulary $n = |V|$, size of embedding d

```
# input: batch of integers
Embedding(output_dim=d, input_dim=n, input_length=1)
# output: batch of float vectors
```

Implementation with Keras

Size of vocabulary $n = |V|$, size of embedding d

```
# input: batch of integers
Embedding(output_dim=d, input_dim=n, input_length=1)
# output: batch of float vectors
```

- Equivalent to one-hot encoding multiplied by a weight matrix $\mathbf{W}$

$$embedding(x) = onehot(x).\,\mathbf{W}$$

Implementation with Keras

Size of vocabulary $n = |V|$, size of embedding d

```
# input: batch of integers
Embedding(output_dim=d, input_dim=n, input_length=1)
# output: batch of float vectors
```

- Equivalent to one-hot encoding multiplied by a weight matrix

$$embedding(x) = onehot(x). \mathbf{W}$$

- $\mathbf{W}$ is typically **randomly initialized**, then **tuned by backprop**

Implementation with Keras

Size of vocabulary $n = |V|$, size of embedding d

```
# input: batch of integers
Embedding(output_dim=d, input_dim=n, input_length=1)
# output: batch of float vectors
```

- Equivalent to one-hot encoding multiplied by a weight matrix $\mathbf{W}$

$$embedding(x) = onehot(x). \mathbf{W}$$

- $\mathbf{W}$ is typically **randomly initialized**, then **tuned by backprop**
- $\mathbf{W}$ are trainable parameters of the model

Distance and similarity in Em
space

Euclidean distance

$$d(x, y) = ||x - y||_2$$

- Simple with good properties
- Dependent on norm (embeddings usually unconstrained)

Distance and similarity in Em
space

Euclidean distance

$d(x, y) = ||x - y||_2$

- Simple with good properties
- Dependent on norm (embeddings usually unconstrained)

Cosine similari

$cosine(x, y) = \frac{x \cdot y}{||x|| \cdot ||}$

- Angle between poi
norm
- $cosine(x, y) \in (-$
- Expected cosine si
pairs of vectors is (

Visualizing Embeddings

- Visualizing requires a projection in 2 or 3 dimensions
- Objective: visualize which embedded symbols are similar

Visualizing Embeddings

- Visualizing requires a projection in 2 or 3 dimensions
- Objective: visualize which embedded symbols are similar

PCA

- Limited by linear projection, embeddings usually have complex structure

Visualizing Embeddings

- Visualizing requires a projection in 2 or 3 dimensions
- Objective: visualize which embedded symbols are similar

PCA

- Limited by linear projection, embeddings usually have complex structure

t-SNE

Visualizing data using t-SNE, L van der Maaten, G Hinton, *The Journal of Machi*
2008

t-Distributed Stochastic Neig
Embedding

- Unsupervised, low-dimension, non-linear projection
- Optimized to preserve relative distances between nearest neigh
- Global layout is not necessarily meaningful

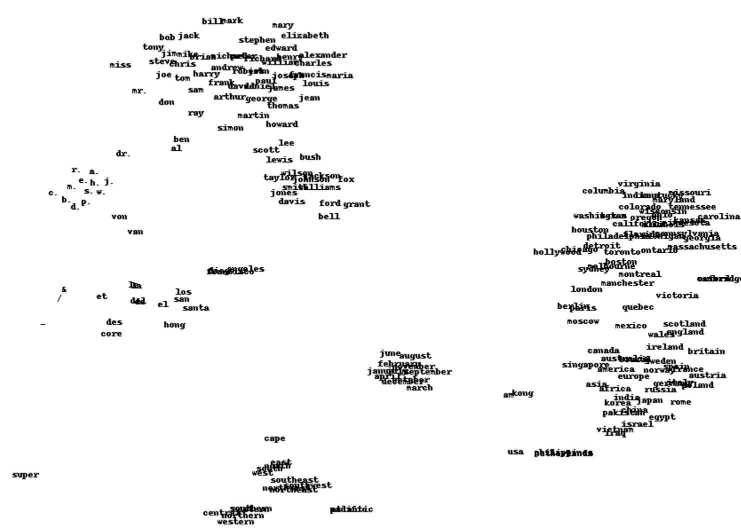
t-Distributed Stochastic Neig
Embedding

- Unsupervised, low-dimension, non-linear projection
- Optimized to preserve relative distances between nearest neigh
- Global layout is not necessarily meaningful

t-SNE projection is non deterministic (depen
initialization)

- Critical parameter: perplexity, usually set to 20, 30
- See http://distill.pub/2016/misread-tsne/

Example word vectors



excerpt from work by J. Turian on a model trained by R. Collobert et al