L3HARRIS
FAST. FORWARD.

GOING BEYOND COUNT-BASED METHODOLOGIES WITH SEMANTIC VECTOR EMBEDDING

Trevor Lax

2024/02/20

Introduction – Meaningful Vector Representations of Text in NLP

Tired Old Adages:

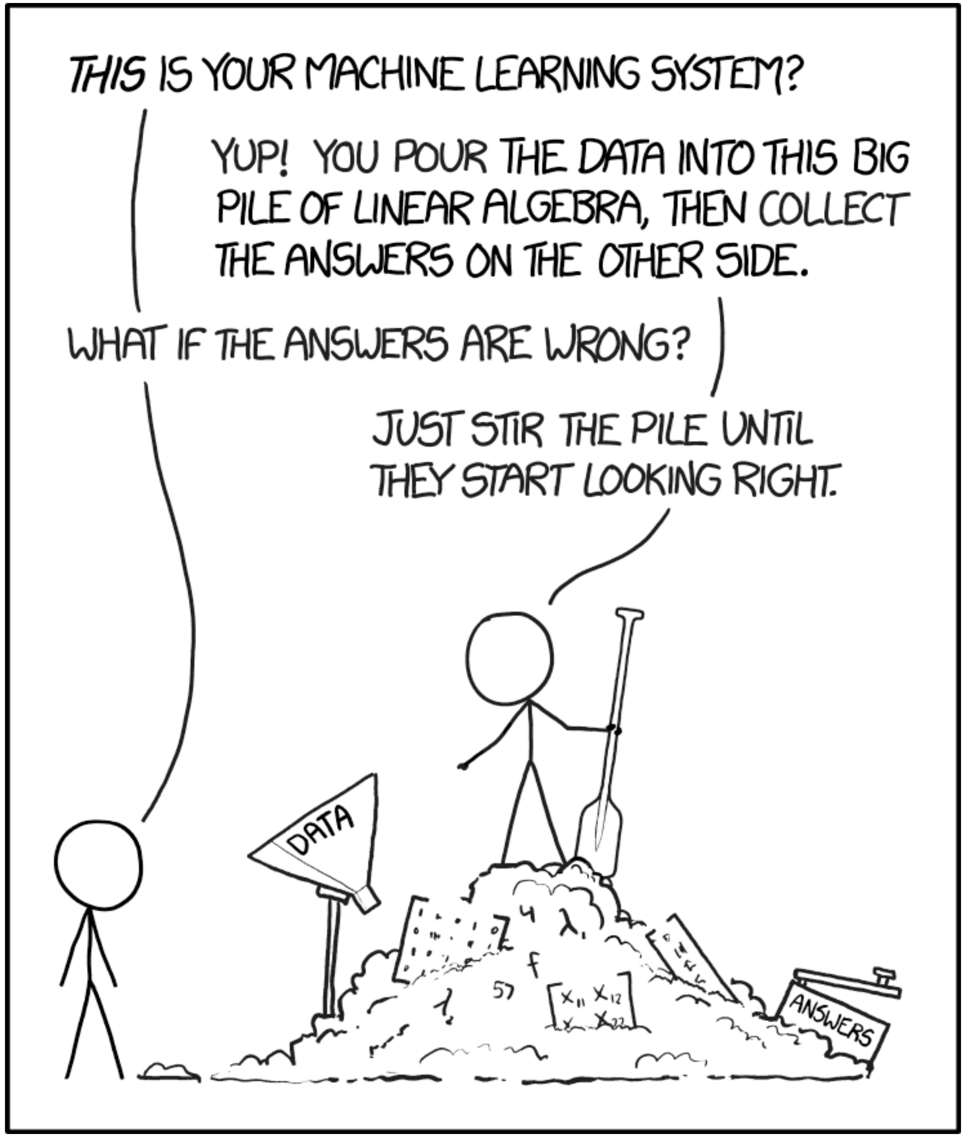
"Familiarity breeds complacency"

"Going back to the Basics"

A Reminder, A Perspective, A Method

New Types of Data

- **"Physical" data, and associated methods, is the most common type, especially in classes and examples**
- **Are often given the data and focused on output**
- **Methods/machines are often ever more complicated**
- **Text is becoming increasingly important**
- **How do we work with text inputs?**
- **What is the input to our methods/machines**
- **And how do we manipulate it?**

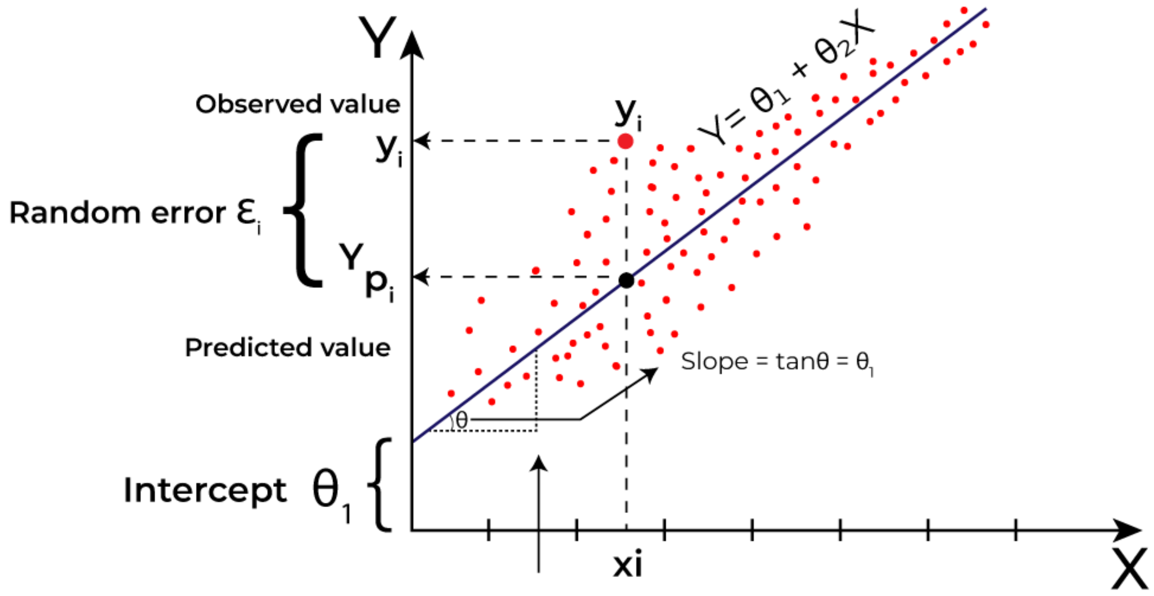


Basic Models

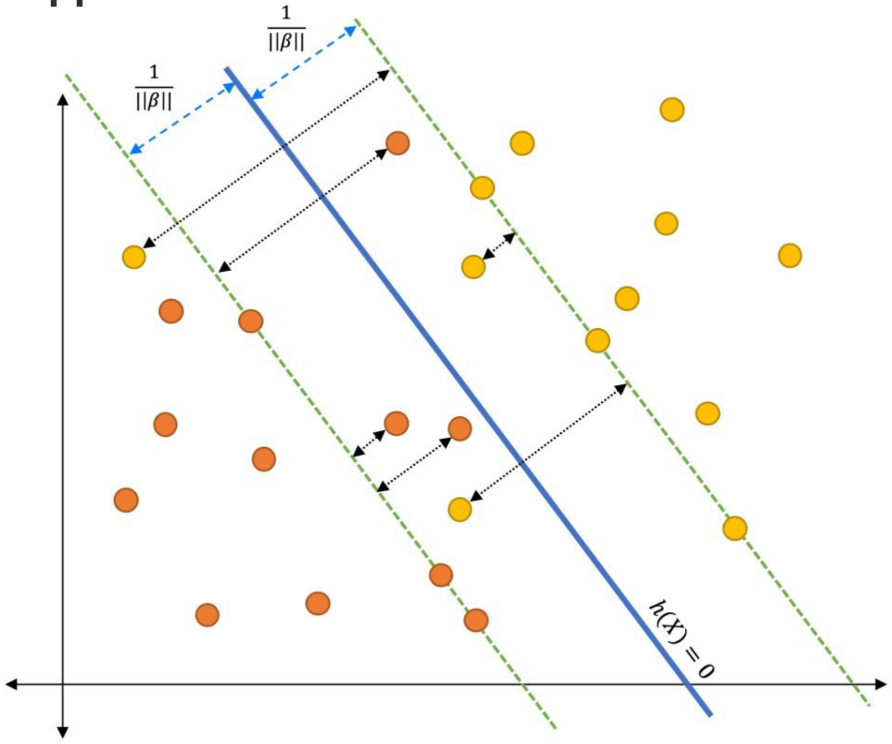
- Fitting a regression line or creating a separating boundary
- At most basic, input is "pre-generated" for us

- Data w/ n variables: {x_1,x_2,…,x_n}
- Vector input: [x_1i, x_2i,…,x_ni]

Regression



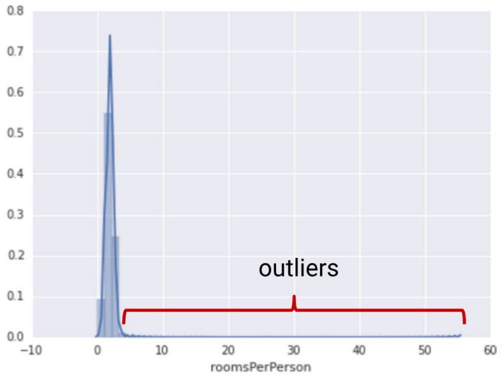
Support Vector Machine



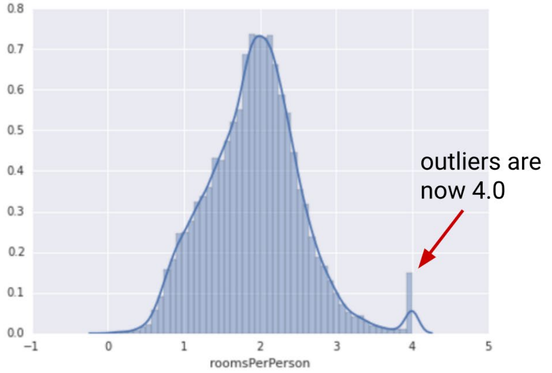
https://www.geeksforgeeks.org/ml-linear-regression/ , Support Vector Machines for Beginners – Linear SVM by Abhisek Jana

Data Transformations

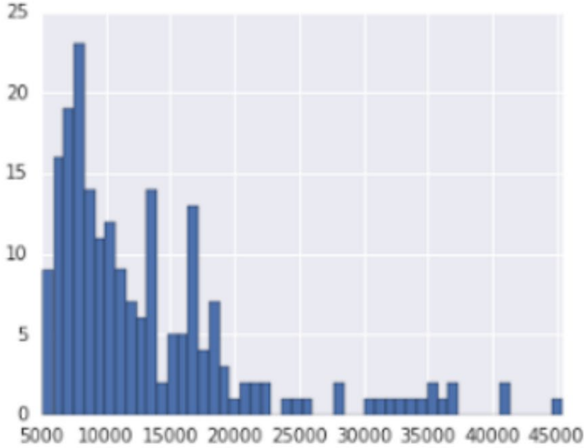
- **Standardize: (x – mean(x))/(std. dev.) = z-score, if normally distributed, becomes standard normal**
- **Normalize: (x – min(x))/(max(x) – min(x)) → [0,1]**
- **Log transform – make it closer to a normal distribution**



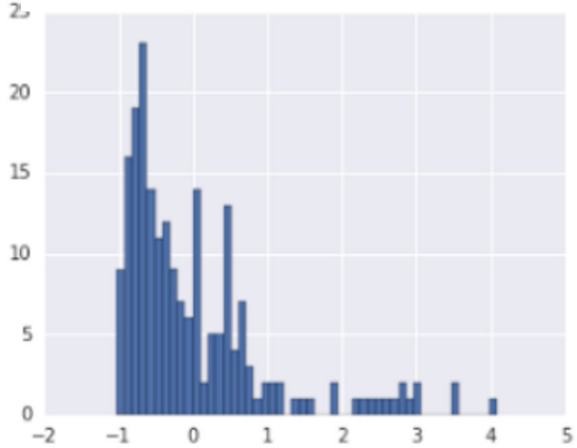
Same feature, capped to a max of 4.0



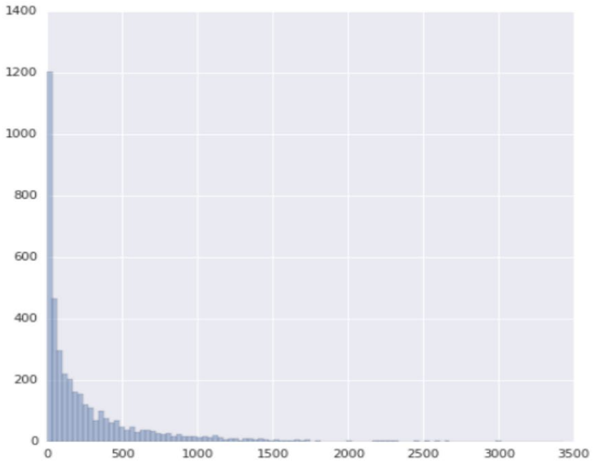
price (raw feature)



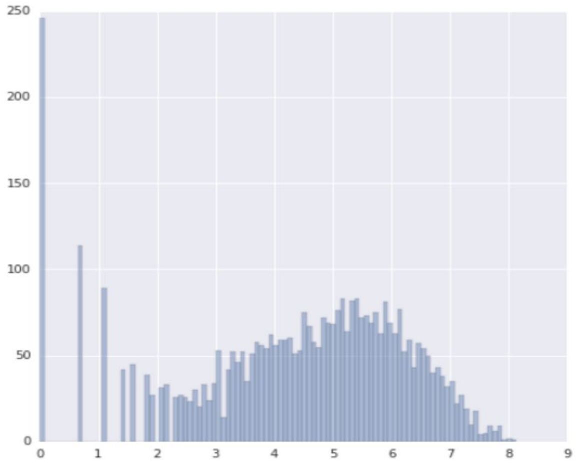
normalized (z-score)



Ratings per movie



Log ratings per movie



None of these transformations change the STRUCTURE of the INPUT

https://developers.google.com/machine-learning/data-prep/transform/normalization

Categorical variables

- Most numerical data is measurements, regardless of whether real-valued or continuous
- Categorical variables are not measurements and cannot be directly plugged into the input
- Require a transformation
- One-Hot and Dummy encoding are popular methods
- Creates a vector representation

$$b = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_k \end{bmatrix} = (X^{'}X)^{-1}X^{'}Y$$

Color
Red
Green
Blue

Dummy encoding

d1	d2
1	0
0	1
0	0

One-hot encoding

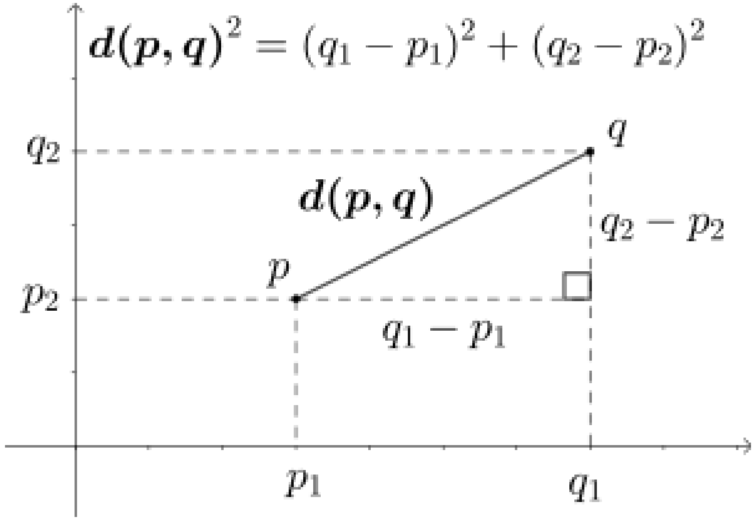
		cat	mat	on	sat	the
the	=>	0	0	0	0	1
cat	=>	1	0	0	0	0
sat	=>	0	0	0	1	0
...		...				

Encoding Categorical Variables: One-hot vs Dummy Encoding by Rukshan Pramoditha :: https://www.tensorflow.org/text/guide/word_embeddings

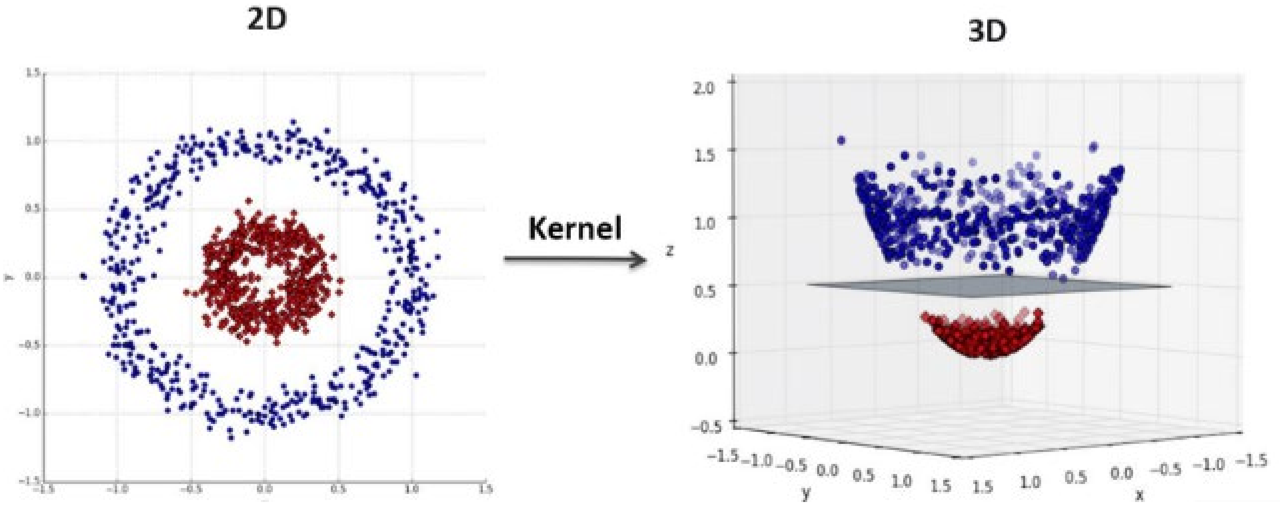
Similarity measures and Distance

- **Gaussian kernel and Euclidean distance are common "similarity" measures**
- **In either creating the model/machine or in the error calculations**
- **Euclidean distance: Sum of Square Errors**
- **Kernels: similarity in *implicit* higher dimensional space**
- **Closer means more similar**

$$\|x - y\| = \sum_{i=1}^{d} (x_i - y_i)^2$$



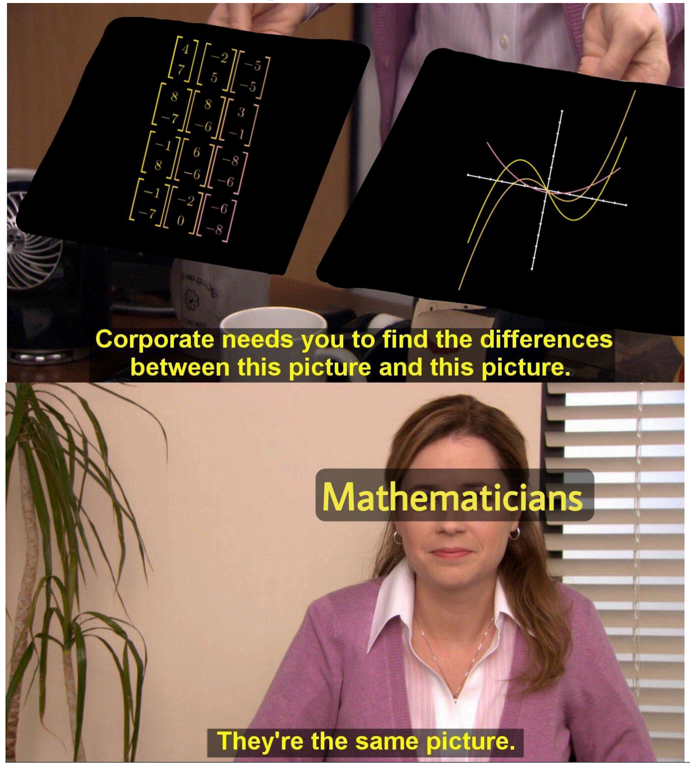
$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$



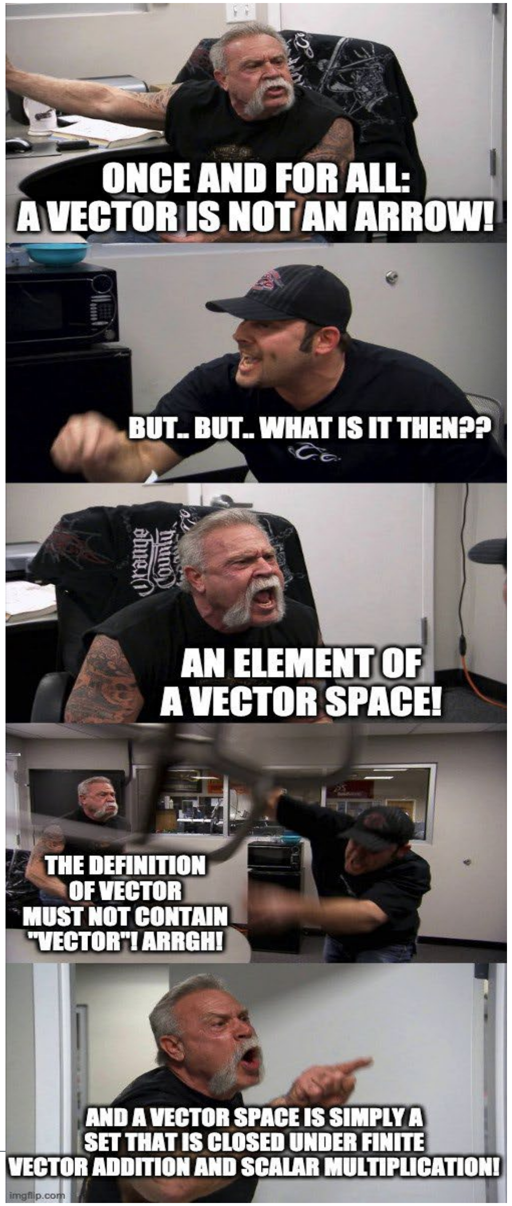
What is Kernel Trick in SVM ? Interview questions related to Kernel Trick by Suraj Yadav

Vector Space

- …close in what?
- Separating boundary in what?
- Placing the data in the vector space is usually trivial
- Focus on working with data that is already placed in a vector space

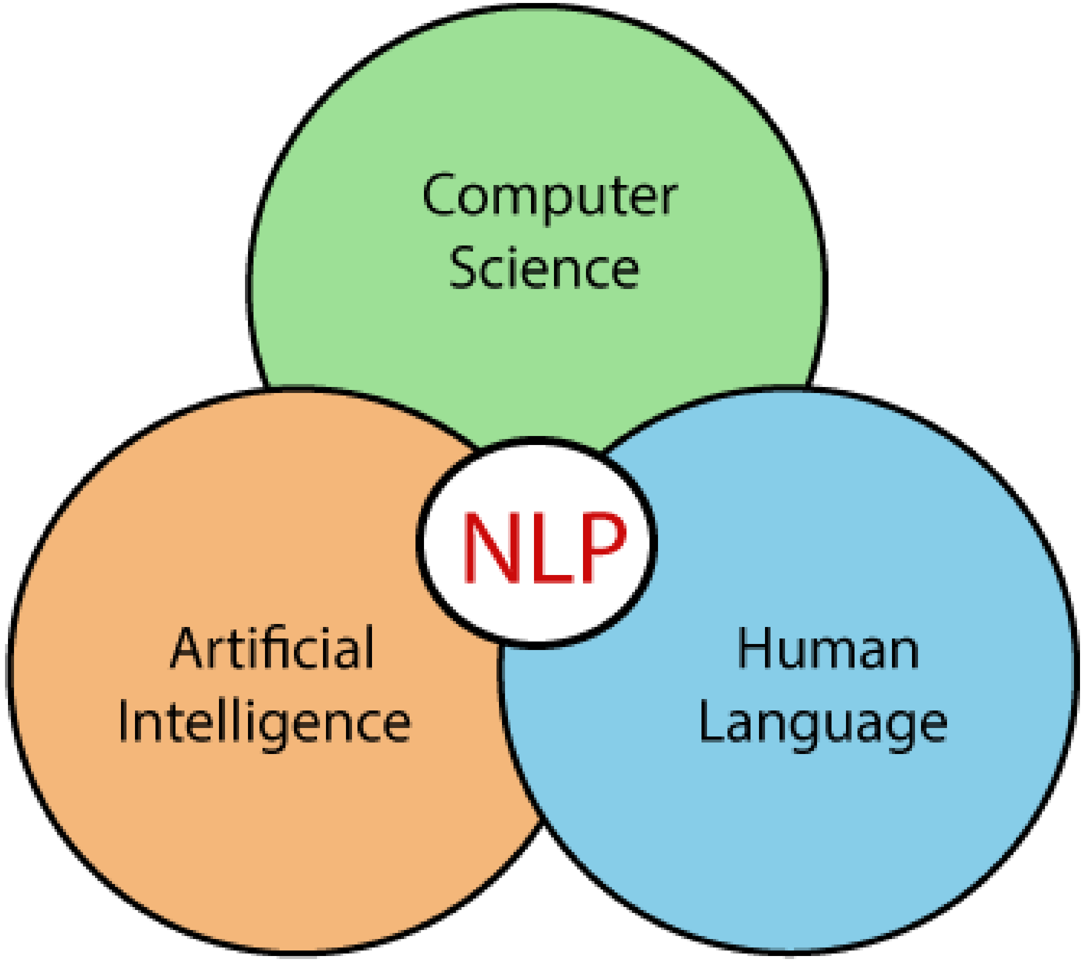


Physical data is always already embedded



Natural Language Processing (NLP)

- 1940s – started with a lot of expert systems hunting for syntax
- 1950s – Turing test (end users really are that bad)
- Several Kuhn-ian Paradigm shifts and AI winters later
- Late 1980's - the rise of statistical methods
- 1997 - Long Short-Term Memory (type of NN)
- 2011 - IBM's Watson won Jeopardy!
- 2011 – Personal assistants
- 2018 – BERT (Bidirectional Encoder Representations from Transformations)

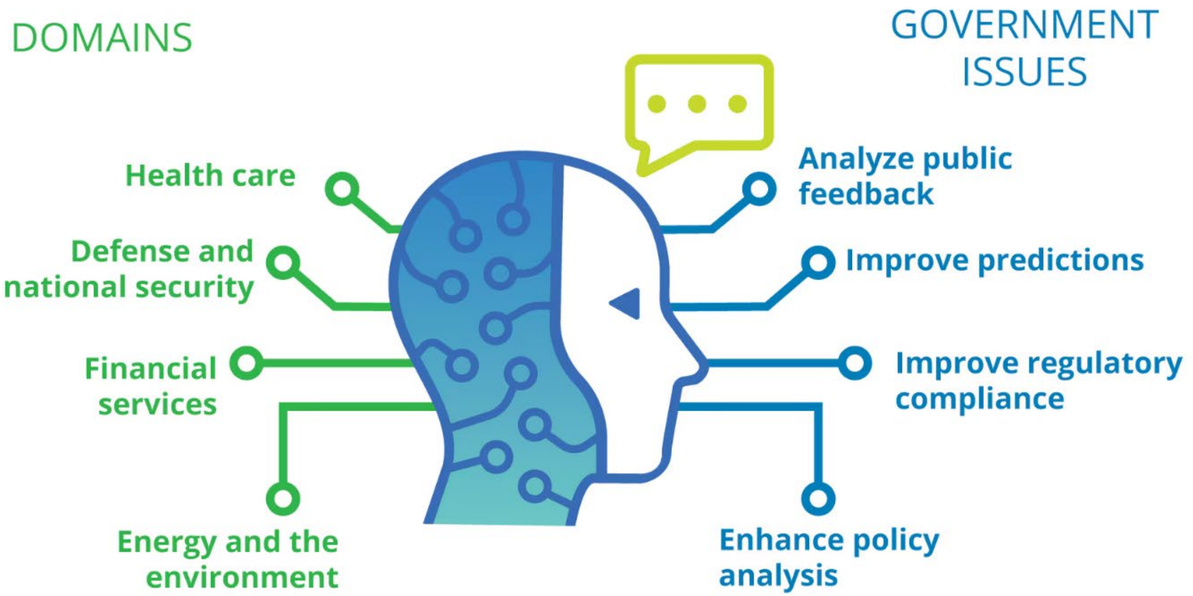


NLP – importance

- Automated assistants
- Google searches
- Autocorrect
- Translation

- Enables use of massive amounts of new data
- Data classification
- Sentiment Analysis
- Actuals – automated standard Work Breakdown Structure (WBS) assignment

NLP can address important issues across government domains

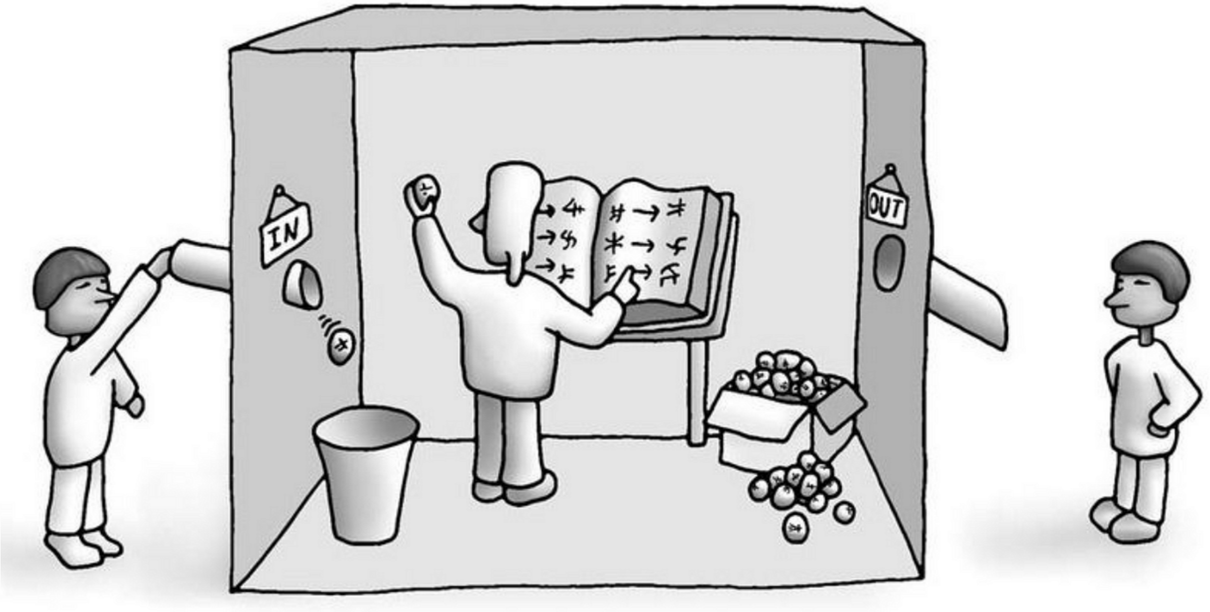


Source: Deloitte analysis.

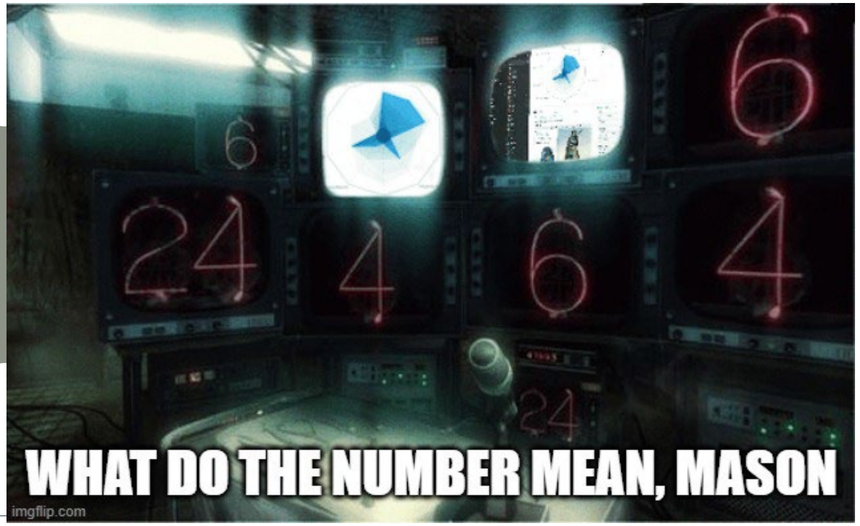
Deloitte Insights | deloitte.com/insights

Mapping and Embedding – Semantics!

- Need to work with text
- Data is no longer inherently in vector form.
- Physical data and measurements ARE
- Story from data, not meaning of measurement
- “Meaning” is not a necessary question
- How do we map Textual to Numerical?
- What level do we make our representation?
- Word, sentence, document?

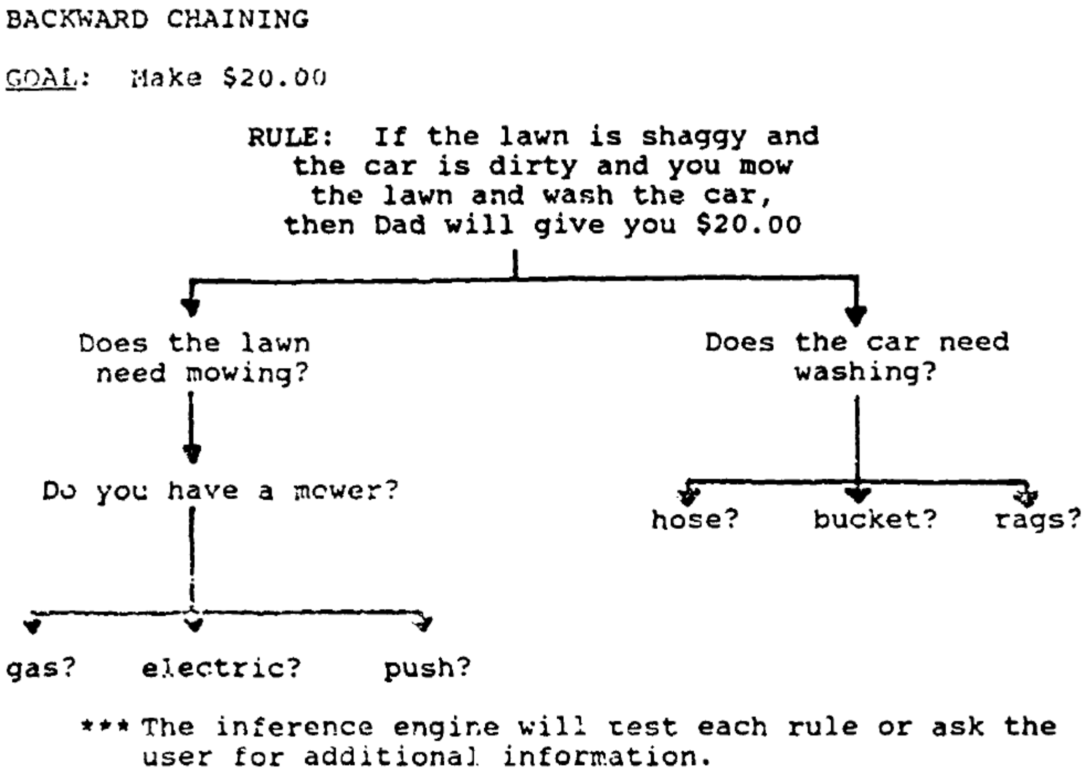


How do we create “meaningful” numerical representations of text?



Expert Systems

- First attempts at AI
- Rules based systems – giant collections of IF-THEN statements
- Translation - databases of words, meanings, and various grammatical rules
- Early AI researchers quickly realized that consciousness contains a fair few rules, most of which are unknown
- Some past success: R1/XCON ordered computer parts based on consumer specifications, DENDRAL helped with identifying unknown organic molecules
- Current: ESSecA: An automated expert system for threat modelling and penetration testing for IoT ecosystems (Rak et. Al.)
- Fuzzy logic and cyclical with data analysis



The vodka is strong, but the meat is rotten.

- Apocryphal

Bag of Words (BoW)

- **Essentially assigning a number to each word, but usually one hot Encoding**

```
It was the best of times,
it was the worst of times,
it was the age of wisdom,
it was the age of foolishness

“it” = 1
“was” = 1
“the” = 1
“best” = 1
“of” = 1
“times” = 1
“worst” = 0
“age” = 0
“wisdom” = 0
“foolishness” = 0

"it was the worst of times" = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]
"it was the age of wisdom" = [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]
"it was the age of foolishness" = [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]
```

- **Can add n-grams – “it was” counts as a word**

- **All words are equally important: Binary – is there or is not**
- **Similar to one-hot encoding**
- **Input length changes with each new word, so the model input structure changes**
- **Does not touch the question of meaning**
- **“took” vs “take”**
- **“fine dinner” vs “got a fine”**

Term Frequency – Inverse Document Frequency (TF-IDF)

- **Possibly the most common measure - tries to correct for lack of weighting in BoW**
- **Words have different values in different documents**
- **Still have vectors with the same dimension as the number of words, similar to BoW**
- **Term Frequency: How often a words occurs in a corpus**
- $tf(t,d) = \frac{f_{t,d}}{\Sigma_{t' \in d} f_{f',d}}$ **(relative tf)**
- **Inverse Document Frequency: 1 / how many documents the word appears in**
- $idf(t) = log\left(\frac{N}{n_t}\right)$
- **The relative number of times the word has occurred times the log of how frequently it appears in a document**
- $\mathrm{TfIdf}(\mathrm{t},\mathrm{d}) \ = \frac{f_{t,d}}{\Sigma_{f_{t\prime,d}} f_{t',d}} \log\left(\frac{\mathrm{N}}{\mathrm{n_t}}\right)$
- **Helps to increase the importance of rare words and temper the influence of very common words**
- **All terms have to be recalculated when new words or documents are added**
- **Still does not attempt to address the issue of “meaning”**

TF-IDF example

Documents
Sphinx of black quartz judge my vow
It is not going to rain today.
It is and it will continue to rain.

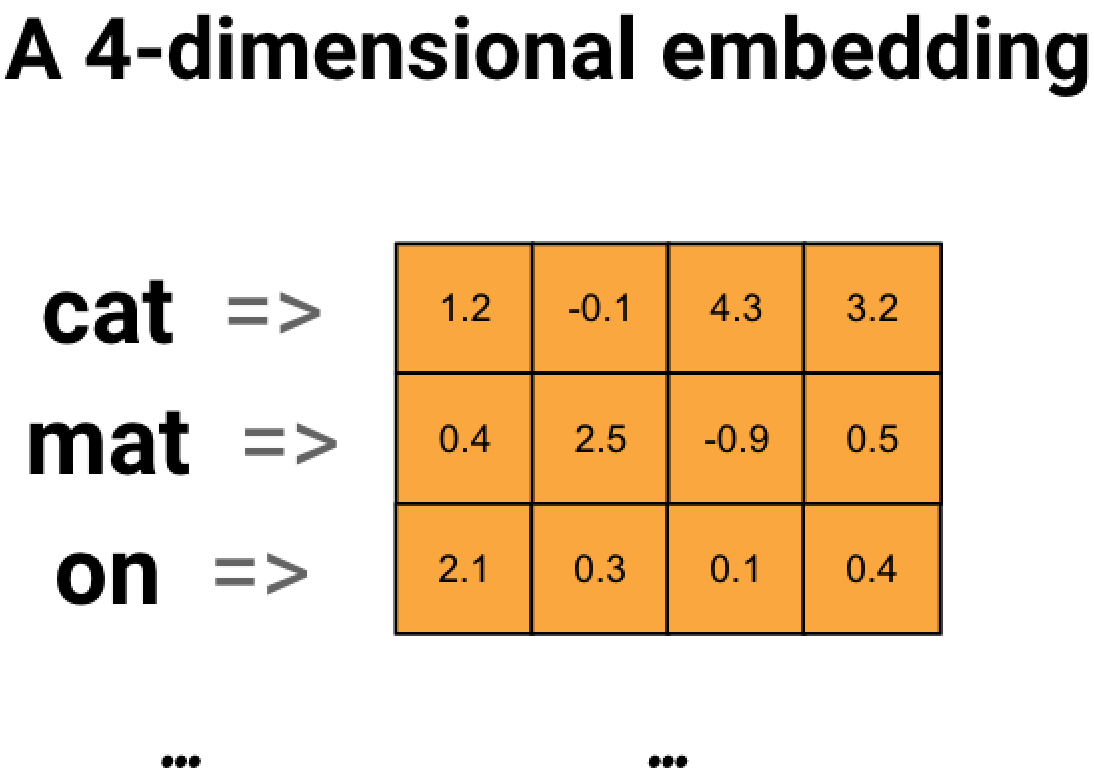
Word	TF – doc 1	TF – doc 2	TF – doc 3
Sphinx	1/7	0	¼
My	1/7	0	0
It	0	1/7	2/8

Word	IDF
Sphinx	Log(3/1)
My	Log(3/2)
It	Log(3/2)

Word	Doc1	Doc2	Doc3
Sphinx	(1/7)*(log(3/1))	0	(1/4)*(log(3/1))
My	(1/7)*(log(3/2))	0	0
It	0	(1/7)*(log(3/2))	(1/4*(log(3/2))

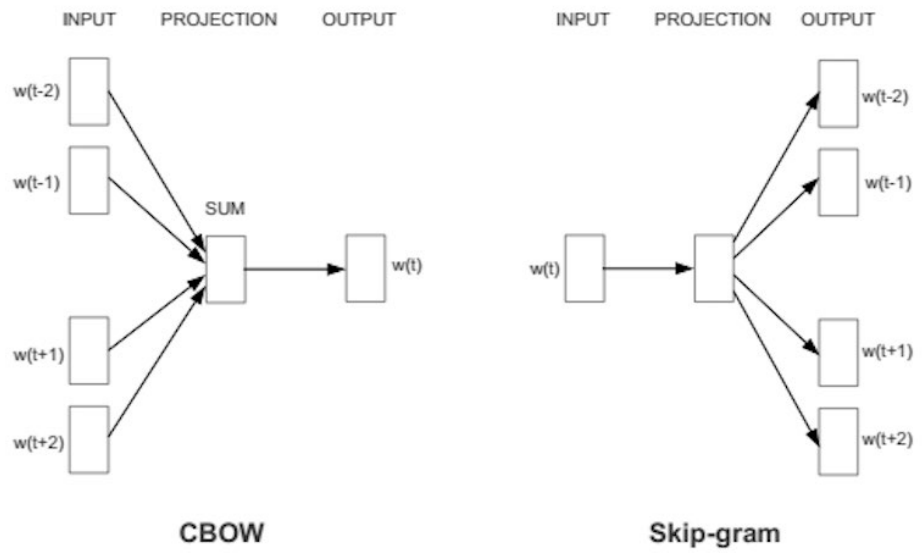
Word2Vec

- Create vector embeddings of words
- Similar to auto-encoders used in translation
- Trained against nearby words, instead of desired translated output, learns context and meaning
- Words (and (later) sentences) that are similar will be near each other in vector space
- Vectors are of a set dimension, instead of growing with vocabulary size
- Since the vectors are independent of document and corpus, they do not have to be recalculated for each new document like TF-IDF

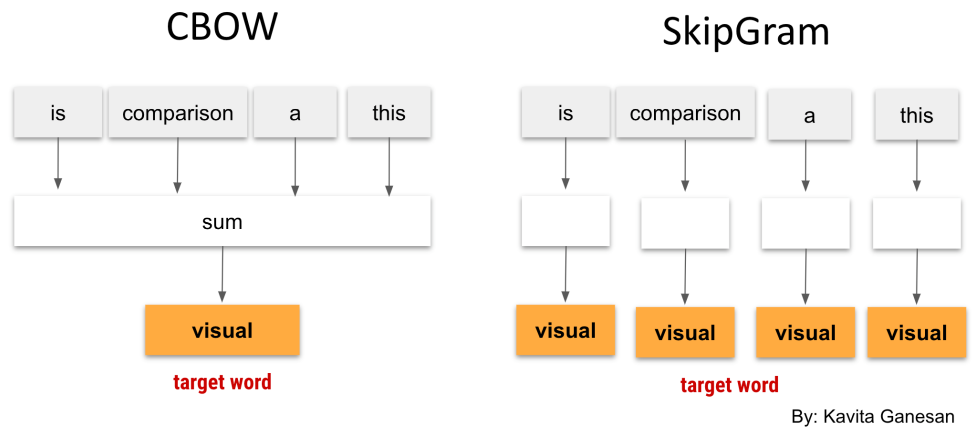


https://www.tensorflow.org/text/guide/word_embeddings

Skip-gram vs CBOW



- CBOW: fill in the blank
- SkipGram: Guess the context from the word
- Generated at the "sentence" level – where "sentence" is a user-defined unit of text
- Shallow Neural Network to predict context or word to learn similar meanings

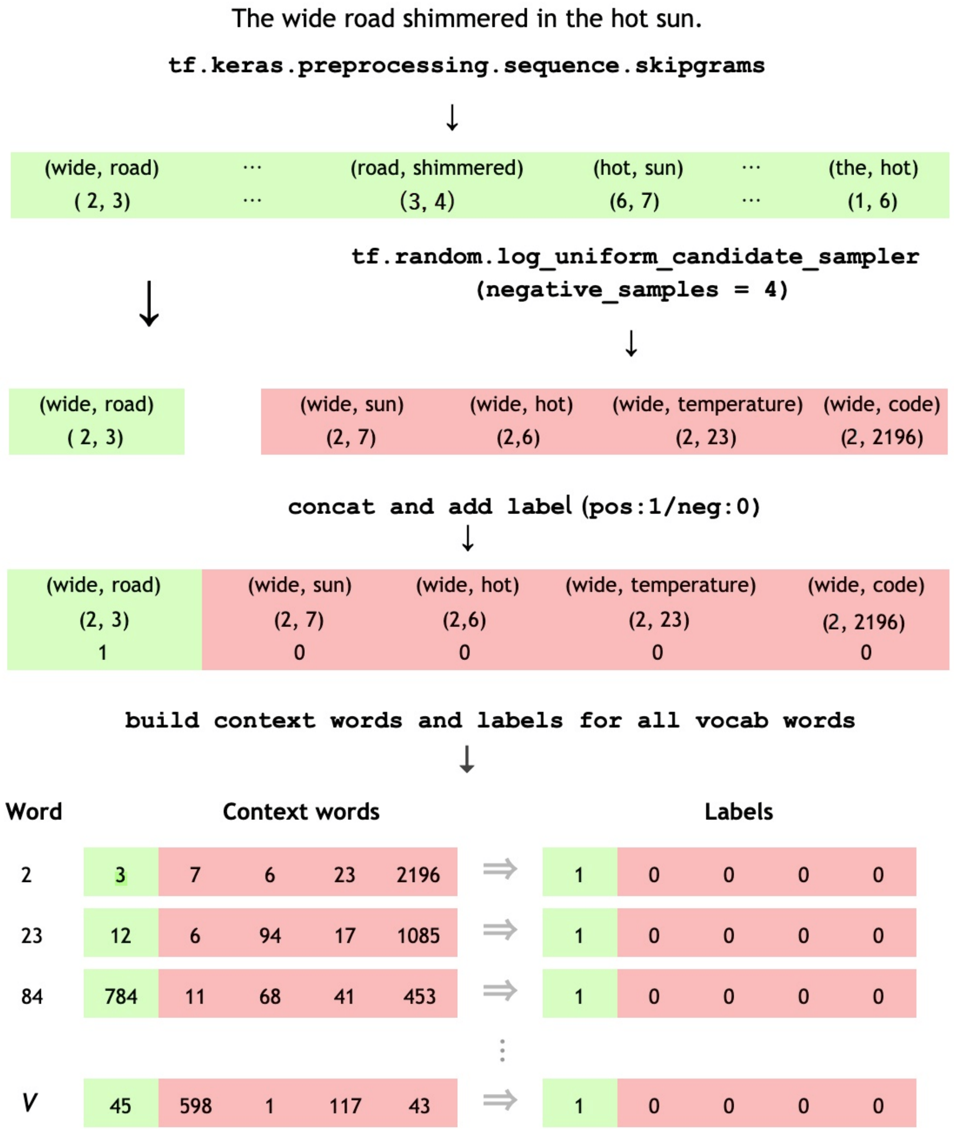


This is a visual comparison

Window Size	Text	Skip-grams
2	[The **wide** road shimmered] in the hot sun.	wide, the wide, road wide, shimmered
	The [wide road **shimmered** in the] hot sun.	shimmered, wide shimmered, road shimmered, in shimmered, the
	The wide road shimmered in [the hot **sun**].	sun, the sun, hot
3	[The **wide** road shimmered in] the hot sun.	wide, the wide, road wide, shimmered wide, in
	[The wide road **shimmered** in the hot] sun.	shimmered, the shimmered, wide shimmered, road shimmered, in shimmered, the shimmered, hot
	The wide road shimmered [in the hot **sun**].	sun, in sun, the sun, hot

Vector creation

- Words are often "vectorized" – converted to numerical representations
- Embeddings is the *actual* vector
- Number of parameters in first layer is vocab_size*embedding_dim
- Loss function: categorical cross entropy (softmax (0,1)) plus log-loss because outputs are one-hot
- Optimizer and hyper-parameters (usually two-layer) for the NN
- Trained on CBOW or Skip-Gram samples



Sentence Embedding

- How do we move from individual words to “sentences” (WBS definitions…)
- Could simply average the words, but have already seen how proper weighting can improve results
- This algorithm does just that, and then subtract off the projection onto the first singular vector
- See Principal Components Analysis (PCA)

Algorithm 1 Sentence Embedding

Input: Word embeddings $\{v_w : w \in \mathcal{V}\}$, a set of sentences $\mathcal{S}$, parameter a and estimated probabilities $\{p(w) : w \in \mathcal{V}\}$ of the words.

Output: Sentence embeddings $\{v_s : s \in \mathcal{S}\}$

for all sentence s in $\mathcal{S}$ **do**

$\quad v_s \leftarrow \frac{1}{|s|} \sum_{w \in s} \frac{a}{a+p(w)} v_w$

end for

Form a matrix X whose columns are $\{v_s : s \in \mathcal{S}\}$, and let u be its first singular vector

for all sentence s in $\mathcal{S}$ **do**

$\quad v_s \leftarrow v_s - uu^\top v_s$

end for

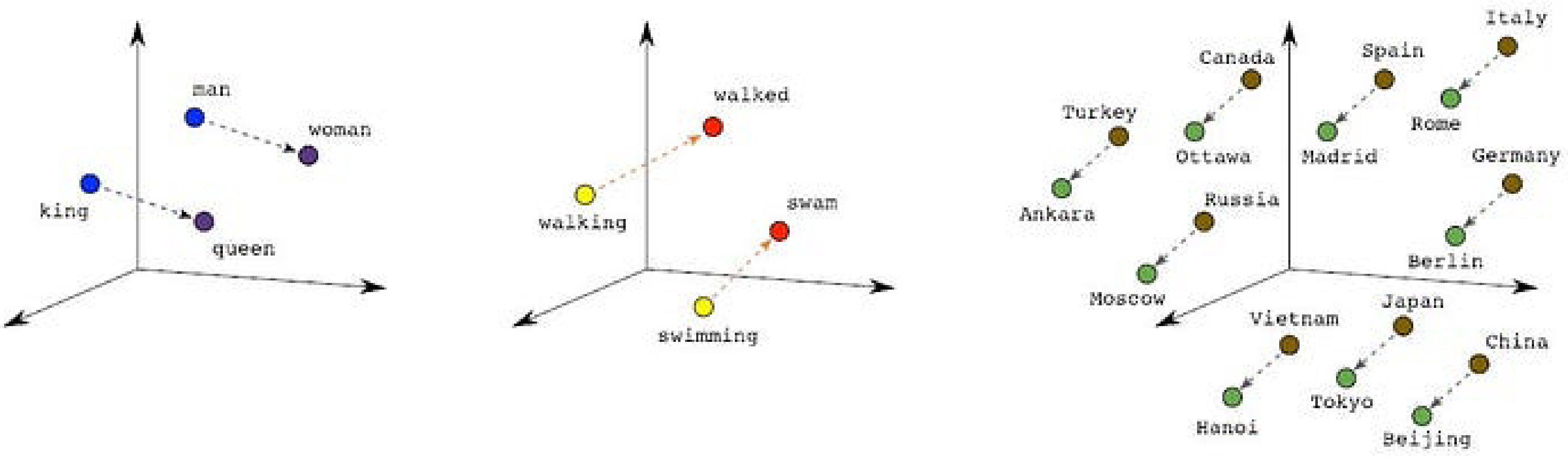
A Simple But Tough-To Beat Baseline For Sentence Embeddings by Sanjeev Arora, Yingyu Liang, and Tengyu Ma

Presented at the ICEAA 2024 Professional Development & Training Workshop - www.iceaaonline.com/min2024

Nearness

- Semantically similar words and those that are connected should be close or connected
- Same concepts as "physical" data can now be applied
- Sentence clustering – which means that SENTENCES that have similar meaning will also be close in space

- Sentence: WBS dictionary definition (user defined)
- Distance now has the same meaning as with physical data
- King + Woman = Queen, in vector space

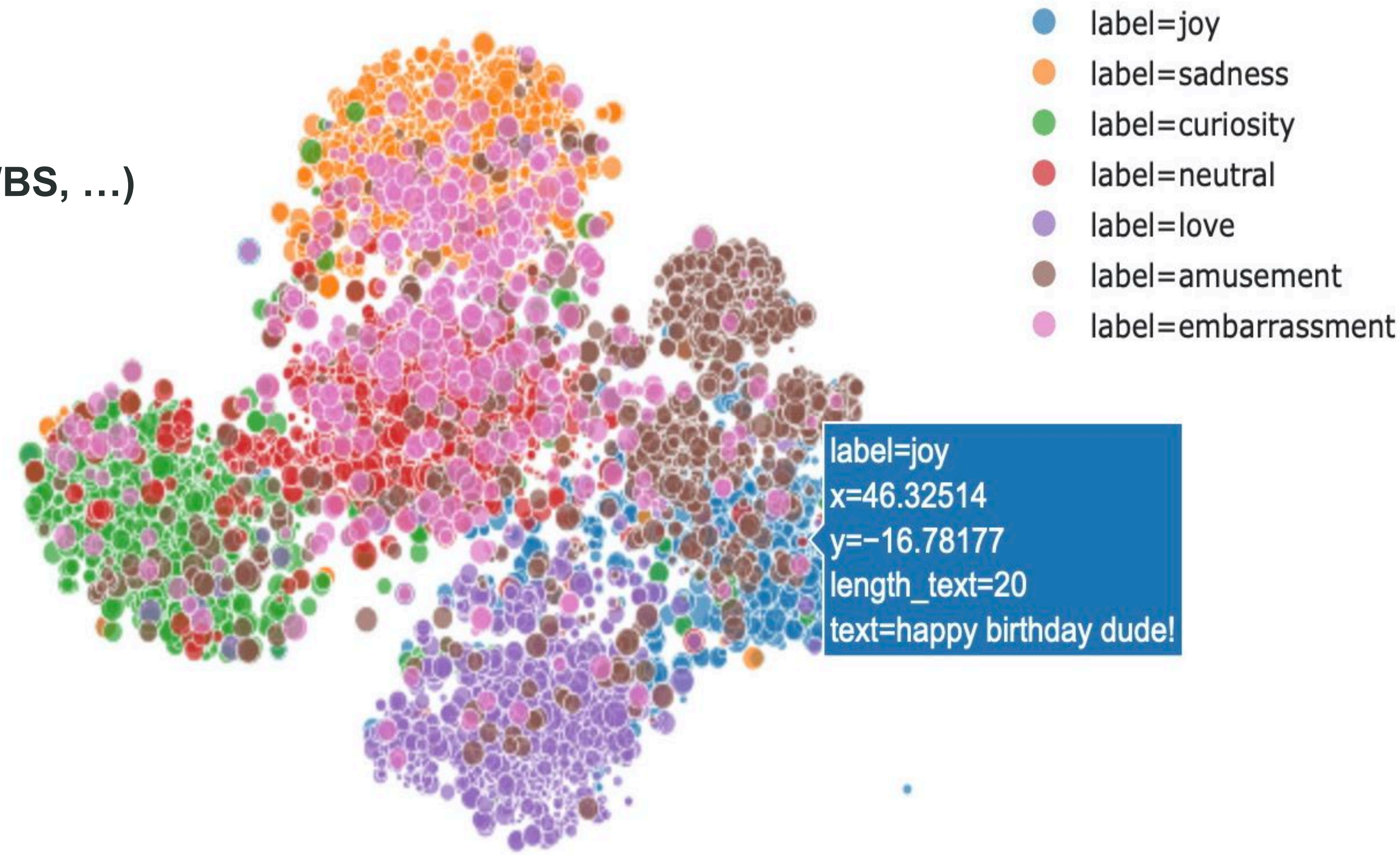


Male-Female Verb Tense Country-Capital

A Guide to Word Embedding What is it? How is it more useful than a Bag-of-Words model? By Shraddha Anala

Clustering

- Similar words (or whatever "unit" of text is) should now be near each other in "space"
- Standard tools work the same with embedded text now that they are embedded
- Clustering analysis
- Common methods (e.g., SVM) – separating boundaries
- Categorization via text (sentiment analysis, WBS, …)



Sentences Embedding Visualization – How to do it the Best Way – Tom Keldenich

Conclusions and Continuations

- Physical data, the type most often worked with, is always already embedded in a vector space
- Most transformations do not change the STRUCTURE of the input or transform the data into a vector
- Text, Natural Language Processing, is becoming increasingly important and accessible
- Text requires vector embedding
- Vector embeddings raise the issue of SEMANTICS
- Word2Vec, and similar methods, possess many of the properties we are used to with physical data
- LSTM (Long Short-Term Memory), order matters
- ATTENTION: very important concept in cognitive science and machine learning methods
- BERT (Bidirectional Encoder Representations from Transformations): can be used for sentiment analysis, text prediction/generation, summarization
- Smaller versions of BERT that can run on smaller computational devices
- ChatGPT (or internally developed alternatives) also uses transformers
- SPACy and TensorFlow: open-source options

This text was placed here intentionally.

Thank you

Questions?