03 – WORD REPRESENTATION

MACHINE LEARNING FOR NATURAL LANGUAGE PROCESSING, AIMS 2024

Dr. Elvis Ndah

OVERVIEW

1. Word meaning
2. Sparse vector representation
3. Frequency based representation
4. Distributional semantics (dense) vector representation
5. Word2vec models

MACHINE LEARNING ON TEXT DATA!!!

Machine learning algorithms work with numeric values and natural language generate data in the form of text.

To solve NLP task with ML we need to represent text data in a way that ML algorithms or statistical techniques can understand.

What should word representation capture?

WHAT IS MEANING (SEMANTICS)?

Meaning in natural language is conveyed by words

- A words is an atomic symbol
- Meaning
  - Idea that is represented by a word, phrase, sentence, etc.
  - Idea expressed in a work of writing, speech or art, etc.

In linguistics a *meaning* maps *a word(s) to an idea*

Signifier (symbol) ⇔ signified (idea or thing)

WHAT IS MEANING (SEMANTICS)?

Corpus:

1. *About the bird, the bird bird bird bird*
2. *You heard about the bird*
3. *the bird is the word*

Vocabulary:

V = {about, bird, heard, is, the, word, you}

	about	bird	heard	is	the	word	you
About the bird, the bird, bird bird bird		?			?		
You heard about the bird		?		?	?		
The bird is the word		?			?		

WHAT IS MEANING (SEMANTICS)?

Lexicographic meaning of a word.

...bar...

What is the context of bar ….?

- is *bar* a place to have a drink?
- is *bar* a long rod?
- is *bar* to block it?

...bank...

What is the context of bank ….?

- is *bank* a financial institution?
- is *bank* a river bank?

REPRESENTING WORDS AS DISCRETE SYMBOLS

- Traditionally NLP regarded words as discrete symbols or localist representation.
- Vector dimension = number of words in a vocabulary (e.g., 10,000+)
- Words are onehot vectors: position (index) of the word in vocabulary is

$bird$	car	$length$	$long$
0	0	0	0
0	0	0	0
0	0	0	1
0	0	0	0
1	0	0	0
0	0	1	0
0	0	0	0
0	1	0	0
0	0	0	0

LIMITATIONS OF SPARSE VECTOR REPRESENTATION

- Computationally expensive because one-hot vectors are very high-dimensional and **sparse vectors**
- have no concept of similarity - each word vector is orthogonal to all others:
  - *long/length* are as different as *long/car*

$$v_{long}^{T} \cdot v_{length} = v_{long}^{T} \cdot v_{car} = 0$$

- This representation does not account for semantics, word usage, etc..

FREQUENCY BASED REPRESENTATION

Bag-of-words is a collection of all the words that are present in the document along with their frequencies.

- Corpus C of D documents
- Vocabulary of N unique tokens (features).
- Count vector (frequency matrix) is a DxN matrix M.

	about	bird	heard	is	the	word	you
About the bird, the bird, bird bird bird	1	5	0	0	2	0	0
You heard about the bird	1	1	1	0	1	0	1
The bird is the word	0	1	0	1	2	1	0

TERM FREQUENCY INVERSE DOCUMENT FREQUENCY (TF-IDF)

- A 2D matrix where each term denotes the relative frequency of a particular term (word) in a particular document as compared to other documents.
- Main objective of *tf-idf* is to penalize very frequent words and boost influence of rare words
- **Term Frequency (*tf*):** number of times a term *t* appears in a document *d*, where $d \in D$
- **Inverse document frequency (*idf*):** inverse count of occurances of term *t* in the document set D.

$$tfidf = tf(t,d) * idf(t,D)$$

where,

$$tf(t,d) = \frac{count\ of\ t\ in\ document\ d}{number\ of\ words\ in\ d}$$

$$df(t) = \#\ of\ documents\ with\ t$$

$$idf(t) = \log(\frac{N(D)}{df(t) + \boxed{1}})$$

add one to prevent division by zero. This can occur when a word (feature) in our vocabulary (feature set) is not present in text at prediction time.

EXAMPLE OF TF-IDF

document A : The car is driven on the road.
document B: The truck is driven on the highway.

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	2/7	2/7	Log(2/2) = 0	0	0
Car	1/7	0	Log(2/1) = 0.3	0.043	0
Truck	0	1/7	Log(2/1) = 0.3	0	0.043
Is	1/7	1/7	Log(2/2) = 0	0	0
Driven	1/7	1/7	Log(2/2) = 0	0	0
On	1/7	1/7	Log(2/2) = 0	0	0
Road	1/7	0	Log(2/1) = 0.3	0.043	0
Highway	0	1/7	Log(2/1) = 0.3	0	0.043

- The word *is* and *driven* appears in both documents, it is not as informative.
- The word *car* is more informative because it appears in only one document.

LIMITATIONS OF FREQUENCY-BASED REPRESENTATION

- Ignore the meaning or semantics of the word
- Ignores word context
- Does not capture co-occurrences of words in different documents
- Loses the ordering of the words

"My name is John" is the same as *"Is my name John?"*

NOTE:

- *advantage of TF-IDF have some over raw counts is that rare words are not penalized.*

DISTRIBUTIONAL SEMANTICS

A word's meaning is given by the words that frequently appears close-by *"You shall know a word by the company it keeps"* John R. Firth 1957

When a word w appears in a text, its context is the set of words that appear nearby (within a fixed window).

Use the many context of word *w* to build a representation of *w.*

The ***contexts*** in which a word appears tells us a lot about what it means.

Words that appear in similar contexts have similar meanings

DISTRIBUTIONAL SEMANTICS – WORD EMBEDDINGS

A **word embedding** is a function that maps each word type to a single vector.

- These vectors are typically **dense** and have much **lower dimensionality** than the size of the vocabulary.
  - Dense vector dimension: 50 – 1024

- This mapping function typically ignores that the same string of letters may have **different senses or parts of speech**
  - *a table* in *a table of contents*
  - *to table in table a motion*

- This mapping function typically assumes a **fixed size vocabulary**

DISTRIBUTIONAL SEMANTICS

Distributional semantics: Word embeddings — dense vectors

$$long = \begin{pmatrix} 0.02 \\ -0.11 \\ 0.18 \\ 0.12 \\ -0.22 \\ -0.10 \\ 0.23 \end{pmatrix} \qquad lenght = \begin{pmatrix} 0.02 \\ -0.18 \\ 0.18 \\ 0.12 \\ -0.18 \\ -0.10 \\ 0.20 \end{pmatrix} \qquad \mathrm{car} = \begin{pmatrix} -0.72 \\ 0.81 \\ -0.71 \\ 0.92 \\ 0.78 \\ -0.10 \\ -0.01 \end{pmatrix}$$

We want to capture concept of similarity such that, *similarity(long, length)* > *similarity(long, car)*

$$v_{long}^{T} \cdot v_{length} \gg v_{long}^{T} \cdot v_{car}$$

GENERATING DENSE VECTORS

Word2Vec (Mikolov et al. 2013): The first really influential dense word embeddings

Two ways to think about Word2Vec: a binary classifier

Variants of Word2Vec

- Two different context representations: CBOW or Skip-Gram
- Two different optimization objectives: Negative sampling (NS) or hierarchical softmax

GENERATING DENSE VECTORS - SKIP GRAM

Main idea:

- Use a binary classifier to predict which words appear in the context of a ***target word***.
  - Word in the context of target words are the **positive** examples
  - Words not in context are **negative** examples

$$w_t = \theta X + e_t$$

- The parameter θ of that classifier provide a dense vector representation of the target word (embedding).

- **Main assumption:** words that appear in similar contexts (that have high distributional similarity) will have very similar vector representations.

SKIP GRAM TRAINING DATA

Training sentence

.. lemon, debt problems turning into banking crises as ...

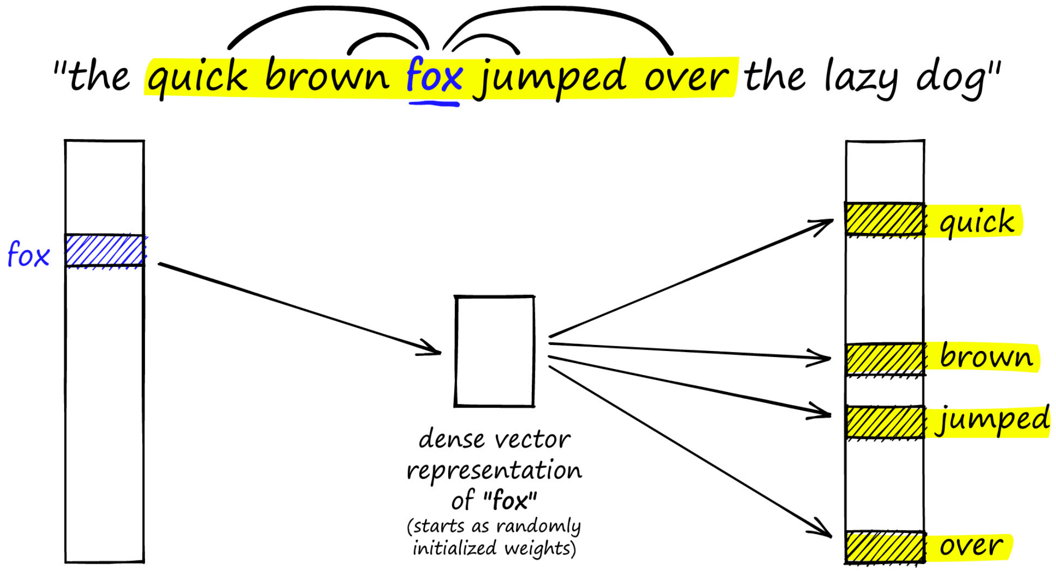
c1 c2 t c3 c4

Training data: input/output pairs centring on apricot (assume a +/-2 window)

- **Positive examples (D+):**
  - *(into, problems), (into, turning), (into, banking), (into, crises)*
- **Negative examples (D-):**
  - *(into, aardvark), (into, puddle)*

GENERATING DENSE VECTORS – SKIP GRAM

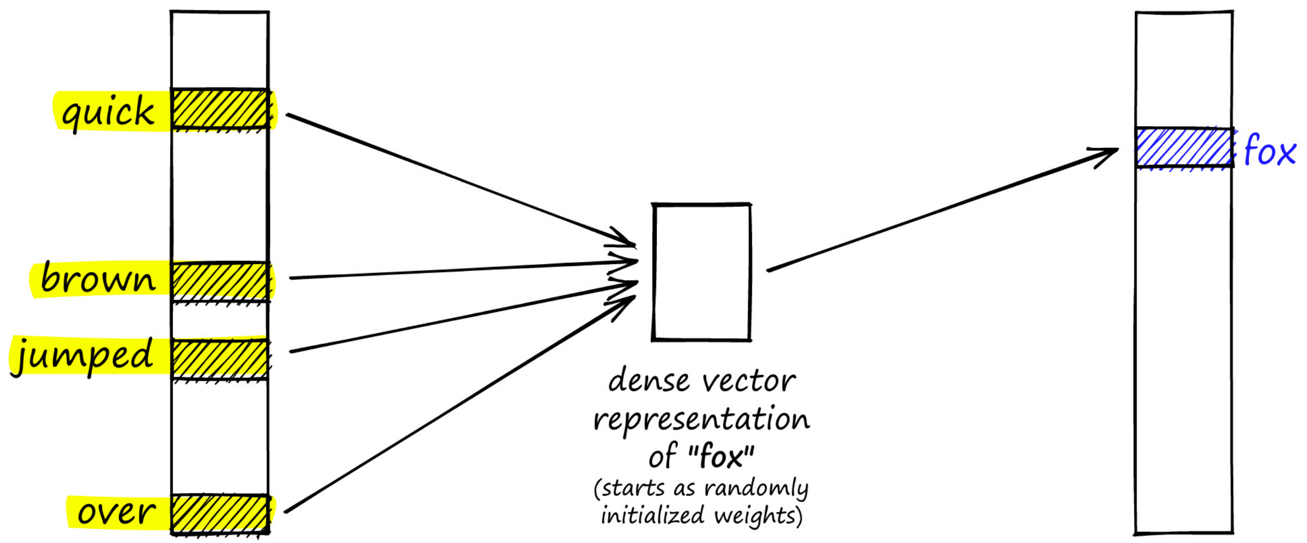
- Given a word fox, train a model to attempts to predict surrounding words (its context).
- After training discard the left and right blocks, keeping only the middle dense vector.



- This vector represents can be used to embed this word for downstream language models.

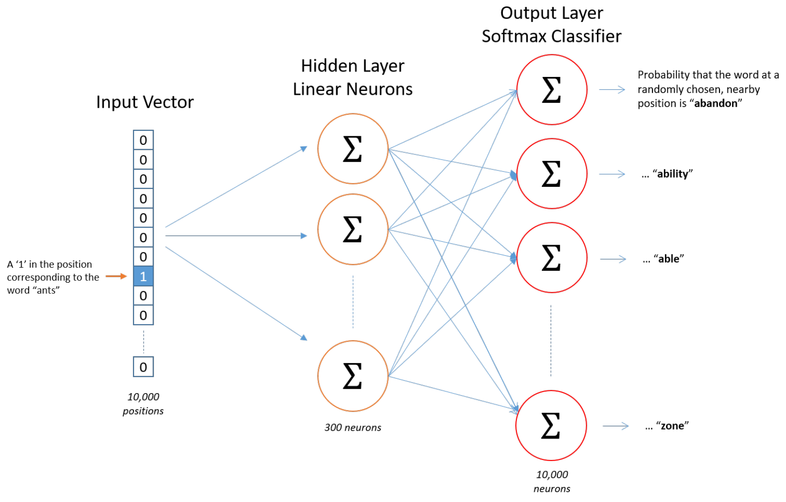
GENERATING DENSE VECTORS – CBOW

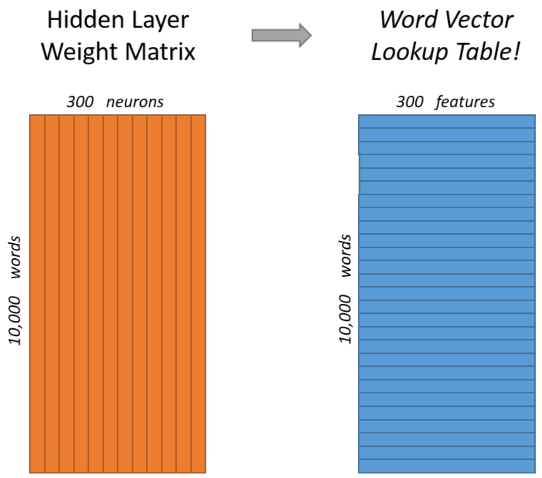
- Continuous bag of words *(CBOW)* aims to predict a word based on its context.



GENERATING DENSE VECTORS – SKIP GRAM

- Single vector representation
  - Vocabulary size of 10,000 words
  - Embedding dimension of 300

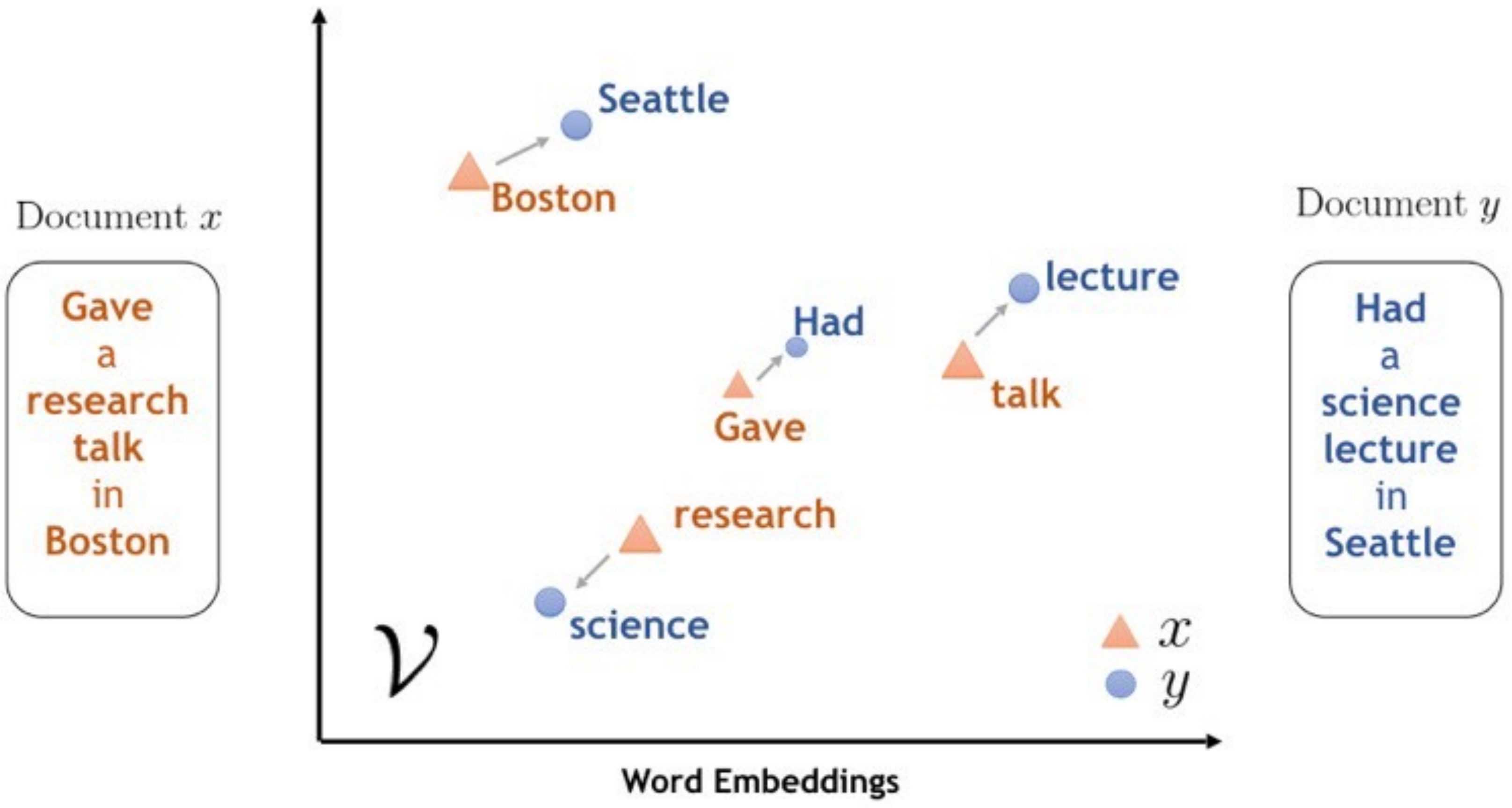




$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

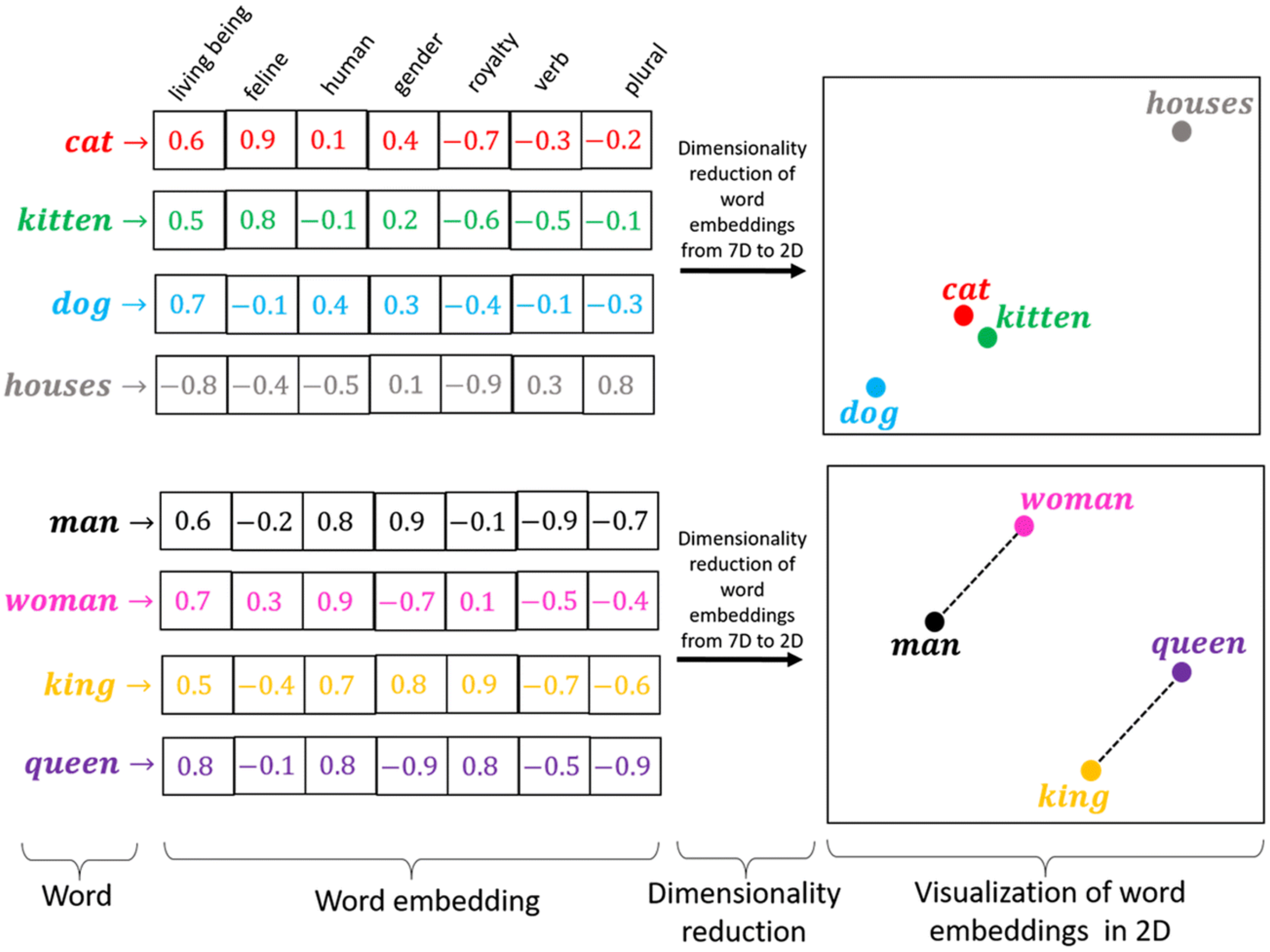
http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

DISTRIBUTIONAL SEMANTICS



www.ibm.com/blogs/research/2018/11/word-movers-embedding/

Analogy: embeddings capture relational meaning



CHARACTERISTICS OF GOOD EMBEDDING MODELS

Non conflation: the embedding model should identify differences in the context and encode them into a meaningful representation e.g., domain, plural, singular, tense

Robust against lexical ambiguity: model should capture meaning of the word and find appropriate embeddings e.g., "the **bow** of a ship" and "**bow** and arrow" should have different embeddings

Demonstrate a multifaceted representation: all properties of word e.g., morphology, syntax, phonetic etc. should contribute to the word representation e.g., representation should change when a prefix is added, or tense is changed

Reliable and consistent: when trained on same dataset even with random initialization the performance of various representation should score consistently.

EVALUATING THE WORD EMBEDDING MODEL

Intrinsic: evaluate quality of representation irrespective of NLP task

- word similarity measures the distance between word vectors and human perceived semantic similarity.
- Word analogy: given a pair of words a and a* and a third word *b* can the analogy relationship between *a* and *a** can be used to find the corresponding word *b**
- Concept categorization: goal is to evaluate if the representations is such that words with similar concepts can be easily clustered together.

Extrinsic

- Evaluation on a real task
- Can take along time to compute accuracy
- Unclear if the subsystem is the problem or its interaction or other subsystems
- If replacing exactly one subsystem with another improves accuracy

COUNT BASED VS PREDICTION-BASED WORD VECTORS

- Count based vectors
  - Faster to train and it efficiently use word statistics
  - Primarily used to capture word similarity
  - Disproportionate importance given to large counts

- Word2vec
  - Scales with corpus size but with Inefficient usage of statistics
  - Generate improved performance on other tasks
  - Can capture complex patterns beyond word similarity

PRE-TRAINED WORD EMBEDDINGS

- Word2vec: https://code.google.com/archive/p/word2vec/
- Fasttext: http://www.fasttext.cc/
- Glove: http://nlp.stanford.edu/projects/glove/
- Gensim: https://radimrehurek.com/gensim/

READING MATERIALS

1. Efficient Estimation of Word Representations in Vector Space (https://arxiv.org/abs/1301.3781)
2. Distributed Representations of Words and Phrases and their Compositionality (https://arxiv.org/abs/1310.4546)
3. A Survey of Word Embeddings Evaluation Methods (https://www.semanticscholar.org/reader/fcf816d9e7b804f4201e4cbf5437e62d683c8a8e)