Vector Semantics

Dr. Demetrios Glinos
University of Central Florida

CAP6640 – Computer Understanding of Natural Language

Today

- Distributional Models of Meaning
- Vectors and Documents
- Words as Vectors
- Pointwise Mutual Information
- Alternatives to PPMI for Measuring Association
- Cosine Similarity
- Alternative Similarity Metrics
- Using Syntax to Define Context
- Evaluating Vector Models

Meaning From Context

- Words that occur in similar contexts tend to have similar meanings
- We can use this to determine the meaning or sense of a word
  - e.g., a thesaurus, which contains lists of synonyms
- Example:

A bottle of raki is on the table.
Everybody likes raki.
Raki makes you drunk.
We make raki from grapes.

  - Suppose you do not already know what "raki" is.
  - What can we conclude about it from this context?
  - How do we reach this conclusion?

Vector Semantics

- **Vector space models (vector semantics)**
  - The name we use for ***distributional models*** of meaning
  - Meaning of word is computed from the *distribution* of words around it
    - e.g., determining what "raki" is from words like "bottle" and "drunk" in close proximity to it
  - Surrounding words are generally represented as a *vector* related to counts
  - Vectors tend to be very long and also sparse
- When we represent a word as a vector, we are ***embedding*** it in a vector space model

Uses of Vector Models of Meaning

- Long history in NLP
  - named entity recognition
  - parsing
  - semantic role labeling
  - relation extraction

- The most common method for computing semantic similarity
  - of words, sentences, and documents
  - question answering
  - summarization
  - automatic essay grading

Today

- Distributional Models of Meaning
- Vectors and Documents
- Words as Vectors
- Pointwise Mutual Information
- Alternatives to PPMI for Measuring Association
- Cosine Similarity
- Alternative Similarity Metrics
- Using Syntax to Define Context
- Evaluating Vector Models

Term-Document Matrices

- **Term-document matrix**
  - each row represents a word in the vocabulary
  - each column represents a document in some collection
  - each cell represents the number of times the row word occurs in a document

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	5	117	0	0

Figure 15.1 The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.

source: J&M (3d Ed. draft)

Document Vector

- **Document vector**
  - identifies a point in |V|- dimensional space

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	5	117	0	0

Figure 15.2 The term-document matrix for four words in four Shakespeare plays. The red boxes show that each document is represented as a column vector of length four.

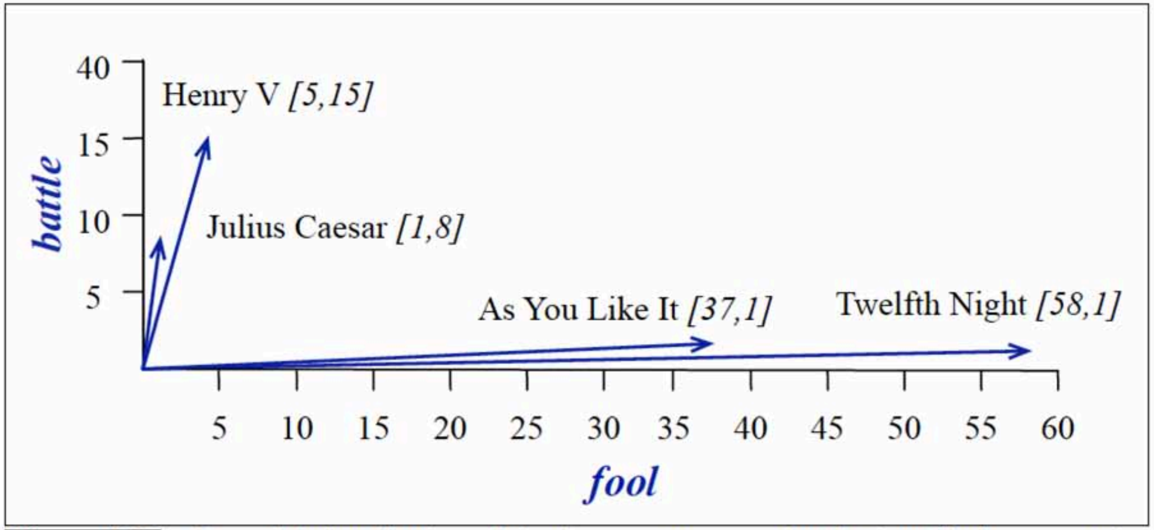


Figure 15.3 A spatial visualization of the document vectors for the four Shakespeare play documents, showing just two of the dimensions, corresponding to the words *battle* and *fool*. The comedies have high values for the *fool* dimension and low values for the *battle* dimension.

source: J&M (3d Ed. draft)

Document Vector Characteristics

- Similar documents tend to have similar document vectors
- Document vector size is |V|
- |V| is typically between 10,000 and 50,000 words
- Including words less frequent than the top 50,000 is generally not helpful
- Vectors are sparse, since most values are zero
- Need to use efficient algorithms for storing and computing with sparse matrices

Today

- Distributional Models of Meaning
- Vectors and Documents
- Words as Vectors
- Pointwise Mutual Information
- Alternatives to PPMI for Measuring Association
- Cosine Similarity
- Alternative Similarity Metrics
- Using Syntax to Define Context
- Evaluating Vector Models

Words as Vectors

- **Term-term (word-word) matrix**
  - matrix is size |V| x |V|
  - more fine-grained than using rows from term-document matrix
  - each cell counts # times the row word and the column word co-occur in corpus in some context
    - could be an entire document
    - more typically, within a window around the row word
    - e.g., column word is within 4 words to left and right of row word
  - window size generally between 1 and 8 words on each side
    - total context from 3 to 17 words
    - small window represents more syntactic relationship
    - larger window represents more semantic relationship

Example: Term Vectors

- 7-word windows, from Brown corpus:

	aardvark	...	computer	data	pinch	result	sugar	...
apricot	0	...	0	0	1	0	1	
pineapple	0	...	0	0	1	0	1	
digital	0	...	2	1	0	1	0	
information	0	...	1	6	0	4	0	

Figure 15.4 Co-occurrence vectors for four words, computed from the Brown corpus, showing only six of the dimensions (hand-picked for pedagogical purposes). The vector for the word *digital* is outlined in red. Note that a real vector would have vastly more dimensions and thus be much sparser.

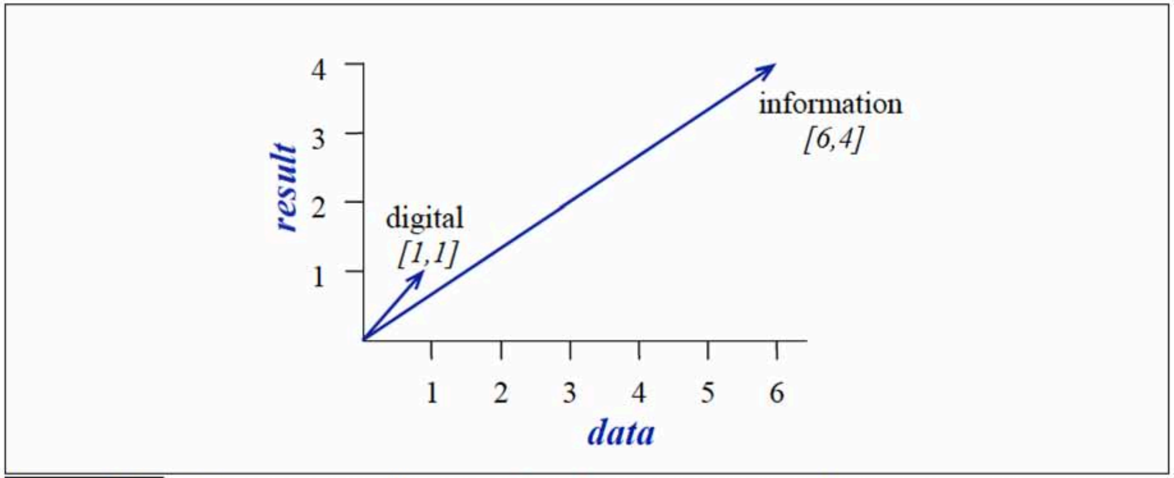


Figure 15.5 A spatial visualization of word vectors for *digital* and *information*, showing just two of the dimensions, corresponding to the words *data* and *result*.

source: J&M (3d Ed. draft)

Types of Word Co-occurrence

- First-order co-occurrence
  - also called syntagmatic association
  - the words themselves are typically found near each other

- Second-order co-occurrence
  - also called paradigmatic association
  - the words typically have similar *neighbors*

Today

- Distributional Models of Meaning
- Vectors and Documents
- Words as Vectors
- Pointwise Mutual Information
- Alternatives to PPMI for Measuring Association
- Cosine Similarity
- Alternative Similarity Metrics
- Using Syntax to Define Context
- Evaluating Vector Models

Pointwise Mutual Information (PMI)

- Not all context words are equally informative
  - *the, it, they* occur frequently in many contexts
  - simple *frequency* (counts) are not the best measure of association between words

- **Mutual information** between random variables X and Y

$$I(X,Y) = \sum_x \sum_y P(x,y) log_2 \frac{P(x,y)}{P(x)P(y)}$$

- **Pointwise mutual information** between events x and y

$$I(x,y) = log_2 \frac{P(x,y)}{P(x)P(y)}$$

PPMI for Words

- PMI measures how often two events occur, compared with what we would expect if they were independent events

- **PMI between target word w and context word c**

$$PMI(w,c) = log_2 \frac{P(w,c)}{P(w)P(c)}$$

- PMI ranges from $-\infty$ to $+\infty$
- Negative values tend to be unreliable unless we use extremely large corpora
- So, instead we use PPMI, which clamps negative values to zero

- **Positive PMI between target word w and context word c**

$$PPMI(w,c) = \max(log_2 \frac{P(w,c)}{P(w)P(c)}, 0)$$

Computing PPMI

- Given co-occurrence matrix F with W rows (words) and C columns (contexts), where f_{ij} is count of time word w_i occurs in context c_j

- We generate a PPMI matrix as follows, where $ppmi_{ij}$ is the PPMI value for w_i and c_j

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W} \sum_{j=1}^{C} f_{ij}}$$

$$p_{i*} = \frac{\sum_{j=1}^{C} f_{ij}}{\sum_{i=1}^{W} \sum_{j=1}^{C} f_{ij}}$$

$$p_{*j} = \frac{\sum_{i=1}^{W} f_{ij}}{\sum_{i=1}^{W} \sum_{j=1}^{C} f_{ij}}$$

$$PPMI_{ij} = \max(log_2 \frac{p_{ij}}{p_{i*} p_{*j}}, 0)$$

Example: PPMI

	aardvark	...	computer	data	pinch	result	sugar	...
apricot	0	...	0	0	1	0	1	
pineapple	0	...	0	0	1	0	1	
digital	0	...	2	1	0	1	0	
information	0	...	1	6	0	4	0	

Figure 15.4 Co-occurrence vectors for four words, computed from the Brown corpus, showing only six of the dimensions (hand-picked for pedagogical purposes). The vector for the word *digital* is outlined in red. Note that a real vector would have vastly more dimensions and thus be much sparser.

source: J&M (3d Ed. draft)

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

$$p_{i*} = \frac{\sum_{j=1}^{C} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

$$p_{*j} = \frac{\sum_{i=1}^{W} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

$$PPMI_{ij} = \max(log_2 \frac{p_{ij}}{p_{i*}p_{*j}}, 0)$$

Assuming the information above represents all relevant data for the entire corpus:

$$p(w = information, c = data) = \frac{6}{19} = .316$$

$$p(w = information) = \frac{11}{19} = .579$$

$$p(c = data) = \frac{7}{19} = .368$$

$$ppmi(information, data) = \max\left(log_2\left(\frac{.316}{.579 \cdot .368}\right), 0\right) = .568$$

Infrequent Event Bias

- PPMI is biased toward infrequent events
  - very rare words tend to have high PPMI values

- Solutions
  - Laplace smoothing (with typical values ranging from 0.1 to 3)
  - Modify computation of P(c) to be a power of α (0.75 found to be useful), which effectively raises the probability of rare events

$$PPMI_{\alpha}\,(w,c) = \max(log_2 \frac{P(w,c)}{P(w)P_{\alpha}(c)}\,, 0)$$

$$P_{\alpha}(c) = \frac{count(c)^{\alpha}}{\sum_{c} count(c)^{\alpha}}$$

Today

- Distributional Models of Meaning
- Vectors and Documents
- Words as Vectors
- Pointwise Mutual Information
- Alternatives to PPMI for Measuring Association
- Cosine Similarity
- Alternative Similarity Metrics
- Using Syntax to Define Context
- Evaluating Vector Models

tf-idf

- **tf-idf (term frequency – indirect document frequency)**
  - used mainly for IR and also in summarization
  - PPMI and t-test are preferred for word similarity

  - tf component
    - simply the frequency (count) of the term in the document
    - can also use log frequency or other function of frequency

  - idf component: $$idf_i = log\left(\frac{N}{df_i}\right)$$

    where N = # docs in corpus, and df_i= # docs in which term *i* occurs

- tf-idf weight for word i in document j: $$w_{ij} = tf_{ij} \cdot idf_i = tf_{ij} \cdot log\left(\frac{N}{df_i}\right)$$

- tf-idf prefers words that are frequent in the current document, but rare overall in the collection

t-test Statistic

- ***t*-test Statistic**
  - a statistical hypothesis test in which the test statistic follows Student's t-distribution under the null hypothesis
  - the t-distribution arises from estimating the mean of a normally distributed population where the sample size is small and standard deviation is unknown
  - computes the difference between observed and expected means, normalized by the variance
  - the higher the value of t, the greater the likelihood that we can reject the null hypothesis that the observed and expected means are equal

t-test Statistic

- Computing the value of t

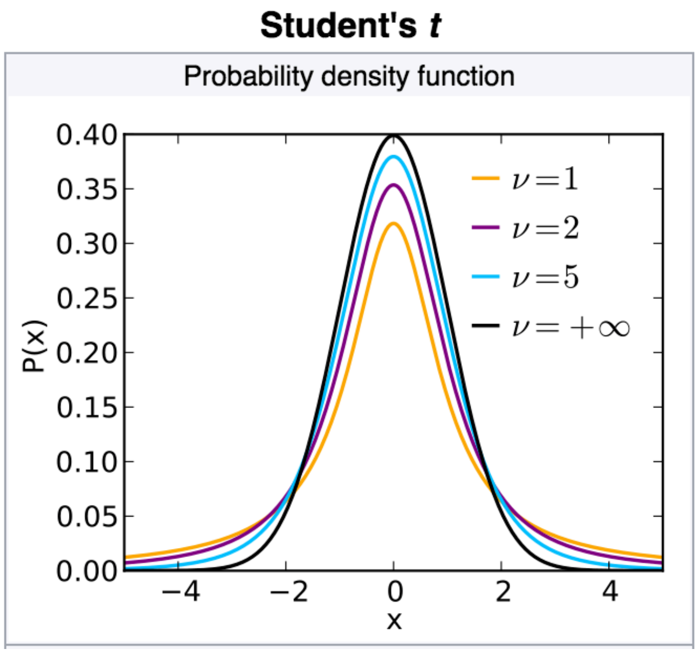
$$t = \frac{\bar{x} - \mu}{\sqrt{\frac{s^2}{N}}}$$

- For words, the null hypothesis is that the words are independent, i.e.,

  P(a,b) = P(a) P(b)

- Ignoring N, since it is constant, we have

$$t\text{-}test(a,b) = \frac{P(a,b) - P(a)P(b)}{\sqrt{P(a)P(b)}}$$



source: Wikipedia

- compute value, then look up confidence in table (degrees of freedom = # samples -1)

Today

- Distributional Models of Meaning
- Vectors and Documents
- Words as Vectors
- Pointwise Mutual Information
- Alternatives to PPMI for Measuring Association
- Cosine Similarity
- Alternative Similarity Metrics
- Using Syntax to Define Context
- Evaluating Vector Models

Cosine Similarity

- **Cosine similarity**
  - by far, the preferred method for determining similarity of vectors
  - computes the angle between two unit vectors in any size vector space
  - compute using normalized dot product

$$cosine(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} {v_i}^2}\ \sqrt{\sum_{i=1}^{N} {w_i}^2}}$$

  - Note: since raw frequency values are non-negative, cosine values will be positive for all word vectors

Example: Cosine Similarity

- Consider these raw frequency values

	large	data	computer
apricot	2	0	0
digital	0	1	2
information	1	6	1

- We compute

$$\cos(apricot, information) = \frac{2+0+0}{\sqrt{4+0+0}\sqrt{1+36+1}} = \frac{2}{2\sqrt{38}} = .16$$

$$\cos(digital, information) = \frac{1+6+1}{\sqrt{0+1+4}\sqrt{1+36+1}} = \frac{8}{\sqrt{5}\sqrt{38}} = .58$$

Today

- Distributional Models of Meaning
- Vectors and Documents
- Words as Vectors
- Pointwise Mutual Information
- Alternatives to PPMI for Measuring Association
- Cosine Similarity
- Alternative Similarity Metrics
- Using Syntax to Define Context
- Evaluating Vector Models

Jaccard Index

- Originally for binary vectors
- Extended to weighted vectors

$$similarity_{Jaccard}(\vec{v},\vec{w}) = \frac{\sum_{i=1}^{N} \min(v_i, w_i)}{\sum_{i=1}^{N} max(v_i, w_i)}$$

- numerator computes the weighted number of overlapping features
- denominator serves as a normalizing value

Dice Coefficient

- Also originally for binary vectors and subsequently extended to weighted vectors

$$similarity_{Dice}(\vec{v},\vec{w}) = \frac{2\times\sum_{i=1}^{N}\min(v_i,w_i)}{\sum_{i=1}^{N}(v_i+w_i)}$$

- same numerator as for Jaccard
- denominator is sum of average weights of both vectors

Information-Theoretic Divergence Measures

- Basic idea: each word vector represents a probability distribution, so they are similar to the extent that these distributions are similar

- KL divergence

$$D(P||Q) = \sum_x P(x) log \frac{P(x)}{Q(x)}$$

- Jensen-Shannon divergence
  - solves problem where KL is undefined for Q(x)=0, which occurs often for sparse word vectors

$$JS(P||Q) = D\left(P|\frac{P+Q}{2}\right) + D\left(Q|\frac{P+Q}{2}\right)$$

i.e.,

$$sim_{JS}(\vec{v}||\vec{w}) = D\left(\vec{v}\,|\frac{\vec{v}+\vec{w}}{2}\right) + D\left(\vec{w}|\frac{\vec{v}+\vec{w}}{2}\right)$$

Today

- Distributional Models of Meaning
- Vectors and Documents
- Words as Vectors
- Pointwise Mutual Information
- Alternatives to PPMI for Measuring Association
- Cosine Similarity
- Alternative Similarity Metrics
- Using Syntax to Define Context
- Evaluating Vector Models

Using Syntax for Context

- Basic idea:

  The meaning of entities, and the meaning of grammatical relations among then, is related to the restriction of combinations of these entities relative to other entities.

  -- Harris, Z.S., *Mathematical Structures of Language* (1968)

- Feature space is expanded to include not only each possible word, but also each possible grammatical relation for each word

- Size of feature space becomes |V|x R, where R is the number of possible relations

Example: Syntax-Based Feature Set

	subj-of, absorb	*subj-of*, adapt	*subj-of*, behave	...	*pobj-of*, inside	*pobj-of*, into	...	*nmod-of*, abnormality	*nmod-of*, anemia	*nmod-of*, architecture	...	*obj-of*, attack	*obj-of*, call	*obj-of*, come from	*obj-of*, decorate	...	*nmod*, bacteria	*nmod*, body	*nmod*, bone marrow
cell	1	1	1		16	30		3	8	1		6	11	3	2		3	2	2

Figure 15.13 Co-occurrence vector for the word *cell*, from Lin (1998), showing grammatical function (dependency) features. Values for each attribute are frequency counts from a 64-million word corpus, parsed by an early version of MINIPAR.

source: J&M (3d Ed. draft)

Alternative Use of Syntax

- Instead of augmenting feature space:
    - Count words in window (same as before)
    - Provided they are in a syntactical dependency relationship with the target word
    - Can also restrict the types of dependency relations that are counted
    - Can also weight the counts based on the length of the dependency path
    - Once we have the counts, we can use PPMI or any other chosen weighting scheme instead of raw frequency counts

Today

- Distributional Models of Meaning
- Vectors and Documents
- Words as Vectors
- Pointwise Mutual Information
- Alternatives to PPMI for Measuring Association
- Cosine Similarity
- Alternative Similarity Metrics
- Using Syntax to Define Context
- Evaluating Vector Models

Evaluating Vector Model Performance

- Extrinsic evaluation
  - adding vector modeling as a component of an NLP task and determining whether this improves performance
- Intrinsic evaluation
  - comparing algorithm similarity scores to scores assigned manually by humans
  - available human judgment datasets
    - WordSim-353 – 0 to 10 ratings on 353 noun pairs (e.g., *plane* and *car*)
    - SimLex-999 – includes both concrete and abstrat adjective, noun and verb pairs
    - TOEFL dataset of 80 questions with 4 choices for each target word
    - Stanford Contextual Word Similarity (SCWS) dataset – 2,003 pairs of words in context