NLP – Vector Semantics and Embeddings

Winter Semester 2023/2024

Oct 26, 2023

Prof. Dr.-Ing. Christian Bergler, Prof. Dr. Patrick Levi | OTH Amberg-Weiden

BoW, TF-IDF, and N-Gram

Short Recap...

Bag-of-Words (BoW)

- Determination of the vocabulary and associated word frequencies across a set of documents (text pieces of varying size, e.g. paragraphs, single pages), which results in a matrix representation (documents × vocabulary size) denoting individual word counts

BoW, TF-IDF, and N-Gram

Short Recap...

Bag-of-Words (BoW)

- Determination of the vocabulary and associated word frequencies across a set of documents (text pieces of varying size, e.g. paragraphs, single pages), which results in a matrix representation (documents × vocabulary size) denoting individual word counts

Term Frequency-Inverse Document Frequency (TF-IDF)

- Identification of all the word-specific frequencies, referred to as "term frequency" (TF), in addition to the number of occurrences per word across all the documents, while the word importance decreases with an increasing cross-document appearance

BoW, TF-IDF, and N-Gram

Short Recap...

Bag-of-Words (BoW)

- Determination of the vocabulary and associated word frequencies across a set of documents (text pieces of varying size, e.g. paragraphs, single pages), which results in a matrix representation (documents × vocabulary size) denoting individual word counts

Term Frequency-Inverse Document Frequency (TF-IDF)

- Identification of all the word-specific frequencies, referred to as "term frequency" (TF), in addition to the number of occurrences per word across all the documents, while the word importance decreases with an increasing cross-document appearance
- Inverse Document Frequency (IDF) $= log(\frac{1+N}{1+df(word)}) + 1$ with N as the number of documents and $df(word)$ as the word-specific document frequency $\rightarrow$ TF × IDF

BoW, TF-IDF, and N-Gram

Short Recap...

Probabilistic Language Modeling – N-Gram

- Probability extended (<s>, </s>) word sequence → $P(\vec{w}) = P(w_0, w_1, ..., w_m, w_{m+1})$

BoW, TF-IDF, and N-Gram

Short Recap...

Probabilistic Language Modeling – N-Gram

- Probability extended (<s>, </s>) word sequence → $P(\vec{w}) = P(w_0, w_1, ..., w_m, w_{m+1})$
- Markov Assumption: $P(w_0, w_1, w_2, w_3, ..., w_m, w_{m+1}) \approx P(w_0) \prod_{i=0}^{m} P(w_{i+1} | w_{i-k+1}^{i})$

BoW, TF-IDF, and N-Gram

Short Recap...

Probabilistic Language Modeling – N-Gram

- Probability extended (<s>, </s>) word sequence → $P(\vec{w}) = P(w_0, w_1, ..., w_m, w_{m+1})$
- Markov Assumption: $P(w_0, w_1, w_2, w_3, ..., w_m, w_{m+1}) \approx P(w_0) \prod_{i=0}^{m} P(w_{i+1} | w_{i-k+1}^{i})$
- Maximum-Likelihood Estimation (MLE): Single word ($\vec{w}_{1\times1}$) or word-phrase/sentence ($\vec{w}_{1\times M}$) via $P(\vec{w}) = P(w_i | w_{i-k}, \ldots, w_{i-1}) = \frac{C(\vec{w})}{C(\vec{w})_{ref}} = \frac{C(w_{i-k},...,w_{i-1},w_i)}{C(w_{i-k},...,w_{i-1})}$ with $C(\vec{w})$ and $C(\vec{w})_{ref}$ as total word/sequence and reference count

BoW, TF-IDF, and N-Gram

Short Recap...

Probabilistic Language Modeling – N-Gram

- Probability extended (<s>, </s>) word sequence → $P(\vec{w}) = P(w_0, w_1, ..., w_m, w_{m+1})$
- Markov Assumption: $P(w_0, w_1, w_2, w_3, ..., w_m, w_{m+1}) \approx P(w_0) \prod_{i=0}^{m} P(w_{i+1} | w_{i-k+1}^{i})$
- Maximum-Likelihood Estimation (MLE): Single word ($\vec{w}_{1\times 1}$) or word-phrase/sentence ($\vec{w}_{1\times M}$) via $P(\vec{w}) = P(w_i | w_{i-k}, \ldots, w_{i-1}) = \frac{C(\vec{w})}{C(\vec{w})_{ref}} = \frac{C(w_{i-k},\ldots,w_{i-1},w_i)}{C(w_{i-k},\ldots,w_{i-1})}$ with $C(\vec{w})$ and $C(\vec{w})_{ref}$ as total word/sequence and reference count
- Challenges: stochastic independence assumption, curse of dimensionality ($|V|^n$), and unknown/unseen word occurrences (out of vocabulary)

BoW, TF-IDF, and N-Gram

Short Recap...

Probabilistic Language Modeling – N-Gram

- Probability extended (<s>, </s>) word sequence → $P(\vec{w}) = P(w_0, w_1, ..., w_m, w_{m+1})$
- Markov Assumption: $P(w_0, w_1, w_2, w_3, ..., w_m, w_{m+1}) \approx P(w_0) \prod_{i=0}^{m} P(w_{i+1} | w_{i-k+1}^{i})$
- Maximum-Likelihood Estimation (MLE): Single word ($\vec{w}_{1\times 1}$) or word-phrase/sentence ($\vec{w}_{1\times M}$) via $P(\vec{w}) = P(w_i | w_{i-k}, \ldots, w_{i-1}) = \frac{C(\vec{w})}{C(\vec{w})_{ref}} = \frac{C(w_{i-k},...,w_{i-1},w_i)}{C(w_{i-k},...,w_{i-1})}$ with $C(\vec{w})$ and $C(\vec{w})_{ref}$ as total word/sequence and reference count
- Challenges: stochastic independence assumption, curse of dimensionality ($|V|^n$), and unknown/unseen word occurrences (out of vocabulary)
- Avoid zero probabilities through smoothing (probability discount and redistribution)

BoW, TF-IDF, and N-Gram

Short Recap...

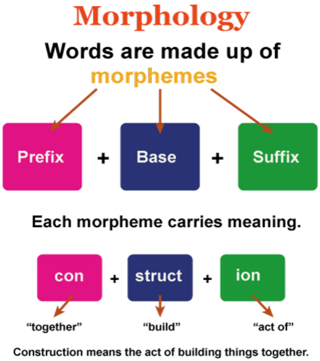
Probabilistic Language Modeling – N-Gram

- Probability extended (<s>, </s>) word sequence → $P(\vec{w}) = P(w_0, w_1, \ldots, w_m, w_{m+1})$
- Markov Assumption: $P(w_0, w_1, w_2, w_3, \ldots, w_m, w_{m+1}) \approx P(w_0) \prod_{i=0}^{m} P(w_{i+1} | w_{i-k+1}^{i})$
- Maximum-Likelihood Estimation (MLE): Single word ($\vec{w}_{1\times1}$) or word-phrase/sentence ($\vec{w}_{1\times M}$) via $P(\vec{w}) = P(w_i | w_{i-k}, \ldots, w_{i-1}) = \frac{C(\vec{w})}{C(\vec{w})_{ref}} = \frac{C(w_{i-k},\ldots,w_{i-1},w_i)}{C(w_{i-k},\ldots,w_{i-1})}$ with $C(\vec{w})$ and $C(\vec{w})_{ref}$ as total word/sequence and reference count
- Challenges: stochastic independence assumption, curse of dimensionality ($|V|^n$), and unknown/unseen word occurrences (out of vocabulary)
- Avoid zero probabilities through smoothing (probability discount and redistribution)
- Model Evaluation: intrinsic (perplexity) and extrinsic (word error rate)

Morphology & Language Modeling

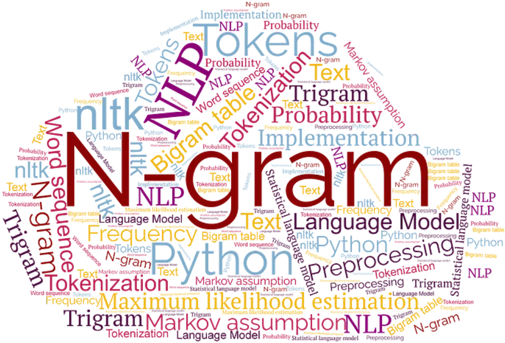
So far ...

Morphology


Morphology
Words are made up of morphemes
Prefix
+
Base
+
Suffix
Each morpheme carries meaning.
con
+
struct
+
ion
"together"
"build"
"act of"
Construction means the act of building things together.

- Describing the structure of words (smallest unit → Morphem)

Language Modeling



- Describing the distribution of words (probability of word sequence $\vec{w}$)

Source: https://www.javatpoint.com/morphology-definition

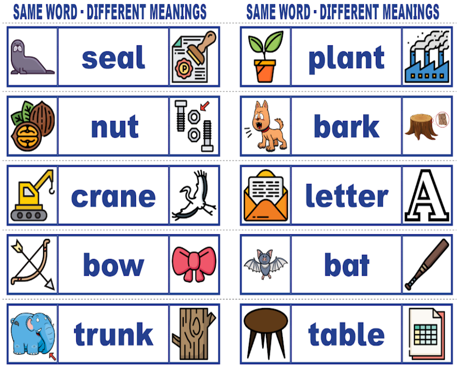
Source: https://medium.com/codex/statistical-language-model-n-gram-to-calculate-the-probability-of-word-sequence-using-python-2e54a1084250

Scope of this Lecture...

“Meaning of words!”

“Theory of word meaning – How does it look like and what do words mean/represent?”

- “Meaning of words” known as Lexical Semantics


SAME WORD - DIFFERENT MEANINGS
SAME WORD - DIFFERENT MEANINGS
seal
plant
nut
bark
crane
letter
A
bow
bat
trunk
table

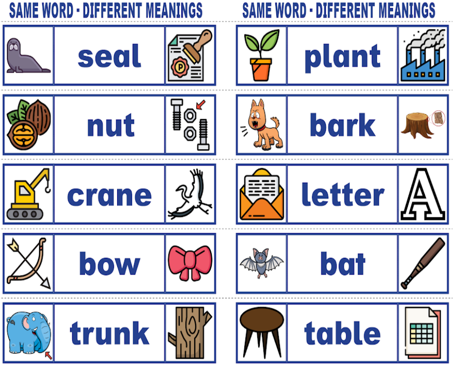
Source: https://www.englishforkidz.com/2020/01/vocabulary-cards-word-different-meanings.html

Scope of this Lecture...

"Meaning of words!"

"Theory of word meaning – How does it look like and what do words mean/represent?"

- "Meaning of words" known as Lexical Semantics
- Lemma vs. Sense:
  - Lemma: "mouse" or "bar"
  - Sense: "mouse" – a little rodent vs. hand-operated control device to move a cursor, "bar" – place to have a drink vs. barrier blocking something


SAME WORD · DIFFERENT MEANINGS
seal
nut
crane
bow
trunk
SAME WORD · DIFFERENT MEANINGS
plant
bark
letter
bat
table

Source: https://www.englishforkidz.com/2020/01/vocabulary-cards-word-different-meanings.html

Scope of this Lecture...

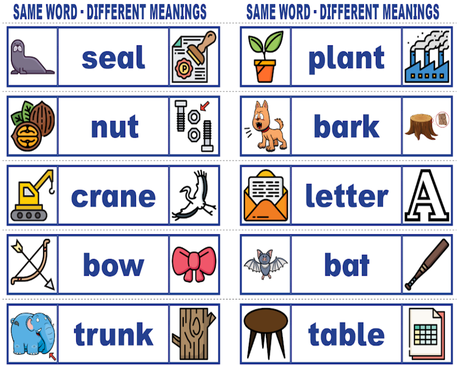
“Meaning of words!”

“Theory of word meaning – How does it look like and what do words mean/represent?”

- “Meaning of words” known as Lexical Semantics
- Lemma vs. Sense:
  - Lemma: “mouse” or “bar”
  - Sense: “mouse” – a little rodent vs. hand-operated control device to move a cursor, “bar” – place to have a drink vs. barrier blocking something
- Sense: the meaning component of a word

→ Polysemy: lemma/lexeme with multiple meanings/senses

→ Homonymy: lemma/lexeme with same spelling (homographs – see Figure), pronunciation (homophones, e.g. “no vs. know”) or both, but different meaning/sense


SAME WORD - DIFFERENT MEANINGS
SAME WORD - DIFFERENT MEANINGS
seal
plant
nut
bark
crane
letter
bow
bat
trunk
table

Source: https://www.englishforkidz.com/2020/01/vocabulary-cards-word-different-meanings.html

Lexical Semantics

“Meaning of words!”

Relation Between Different Words/Senses – Synonymy

- Present the same sense/meaning across almost the entire context
- Synonym examples: wealthy & rich, big & large, small & tiny, scared & afraid, couch & sofa, good & fine, automobile & car, garbage & trash, wheel & tire, and many more...

Lexical Semantics

“Meaning of words!”

Relation Between Different Words/Senses – Synonymy

- Present the same sense/meaning across almost the entire context
- Synonym examples: wealthy & rich, big & large, small & tiny, scared & afraid, couch & sofa, good & fine, automobile & car, garbage & trash, wheel & tire, and many more...
- However, it is very difficult to identify cases with a “perfect” synonymy, due to differences caused by slang, genre, politeness, etc.
- Real synonymy? → “my big brother” vs. “my large brother” or “the food was good” vs. “the food was fine”

Lexical Semantics

“Meaning of words!”

Principle of Contrast

- Different words or word forms possess a different meaning

Lexical Semantics

“Meaning of words!”

Principle of Contrast

- Different words or word forms possess a different meaning
- Similar to the Mutual Exclusivity (ME) Assumption, describing the assignment of exactly a single label/name to a specific object (1-by-1 reference, done by children)

  → However, multiple labels with different meanings for a specific object are possible

Lexical Semantics

“Meaning of words!”

Principle of Contrast

- Different words or word forms possess a different meaning
- Similar to the Mutual Exclusivity (ME) Assumption, describing the assignment of exactly a single label/name to a specific object (1-by-1 reference, done by children)

  → However, multiple labels with different meanings for a specific object are possible
- Deviations from conventional word meanings are based on a different and contrasting underlying meaning

Lexical Semantics

"Meaning of words!"

Principle of Contrast

- Different words or word forms possess a different meaning
- Similar to the Mutual Exclusivity (ME) Assumption, describing the assignment of exactly a single label/name to a specific object (1-by-1 reference, done by children)

  → However, multiple labels with different meanings for a specific object are possible
- Deviations from conventional word meanings are based on a different and contrasting underlying meaning

Principle of Conventionality

- The language community agreed on a defined conventional meaning for words
- Language would not be possible without conventional definitions and agreements, but rather individual-specific vocabularies → Allows communication!

Lexical Semantics

“Meaning of words!”

Relation Between Different Words/Senses – Similarity

- The closeness is not as pronounced as in the case of synonyms, however there exists a thematic similarity and relatedness

Lexical Semantics

“Meaning of words!”

Relation Between Different Words/Senses – Similarity

- The closeness is not as pronounced as in the case of synonyms, however there exists a thematic similarity and relatedness
- Word similarity examples:
  - Car & Bicycle & Motorbike & Lorry & Van
  - Monday & Tuesday & Wednesday & Thursday
  - Football & Basketball & Ice hockey & Golf & Baseball
  - Cow & Horse & Pig & Chicken
  - Artist & Actor, Vanish & Disappear, Coffee & Espresso
  - (. . .)

Lexical Semantics

“Meaning of words!”

Relation Between Different Words/Senses – Relatedness

- “Word Relatedness” is also referred to as Word Association
- Words can be related on the basis of various circumstances, such as a similar semantic field, covering a specific domain with structured semantic relations

Lexical Semantics

“Meaning of words!”

Relation Between Different Words/Senses – Relatedness

- “Word Relatedness” is also referred to as Word Association
- Words can be related on the basis of various circumstances, such as a similar semantic field, covering a specific domain with structured semantic relations
- Word relatedness examples:
  - Car & Engine & Driving & Crash & License plate & Wheel
  - Monday & June & Date & Year & Calendar
  - Football & Goal & Stadium & Ball & Penalty
  - Coffee & Cup & Drinking & Sugar & Milk & Espresso
  - Bank & Money & Loan & Interest Rate & Current Account

Lexical Semantics

“Meaning of words!”

Relation Between Different Words/Senses – Antonymy

- Present the opposite sense/meaning in terms of one particular aspect/feature, however, with a significant word similarity

Lexical Semantics

“Meaning of words!”

Relation Between Different Words/Senses – Antonymy

- Present the opposite sense/meaning in terms of one particular aspect/feature, however, with a significant word similarity
- Antonymy examples:
  - Hot & Cold, Arrival & Departure, Birth & Death, Fear & Courage, Black & White, Artificial & Natural, Accidental & Intentional

Lexical Semantics

“Meaning of words!”

Relation Between Different Words/Senses – Antonymy

- Present the opposite sense/meaning in terms of one particular aspect/feature, however, with a significant word similarity
- Antonymy examples:
  - Hot & Cold, Arrival & Departure, Birth & Death, Fear & Courage, Black & White, Artificial & Natural, Accidental & Intentional
- Antonyms represent either a binary opposition (active & passive), opposite scaling ends (short & long), or reversives (up & down)

Lexical Semantics

“Meaning of words!”

Relation Between Different Words/Senses – Connotation (Sentiment)

- Describe words which possess an affective sense/meaning – “funny” (positive) versus “mad” (negative), “wedding” (positive) versus “funeral (negative)”

Lexical Semantics

“Meaning of words!”

Relation Between Different Words/Senses – Connotation (Sentiment)

- Describe words which possess an affective sense/meaning – “funny” (positive) versus “mad” (negative), “wedding” (positive) versus “funeral (negative)”
- However, the distinction between the actual affective sense/meaning is often challenging
    - “copy” (positive) vs. “forgery” (negative)
    - “reproduction” (positive) vs. “fake” (negative)

and dependent on three different affectivess dimensions:

Lexical Semantics

“Meaning of words!”

Relation Between Different Words/Senses – Connotation (Sentiment)

- Describe words which possess an affective sense/meaning – “funny” (positive) versus “mad” (negative), “wedding” (positive) versus “funeral (negative)”
- However, the distinction between the actual affective sense/meaning is often challenging
  - “copy” (positive) vs. “forgery” (negative)
  - “reproduction” (positive) vs. “fake” (negative)

  and dependent on three different affectivess dimensions:
  - *Valence:* pleasantness of the stimulus
  - *Arousal:* emotional intensity produced by the stimulus
  - *Dominance:* level of control, provided by the stimulus

Lexical Semantics

“Meaning of words!”

Summary

- Complex many-to-many association of senses/meanings and words
  → Polysemy & Homonymy
- There exist different relations in terms of word meaning/sense, including:

Lexical Semantics

“Meaning of words!”

Summary

- Complex many-to-many association of senses/meanings and words
  → Polysemy & Homonymy
- There exist different relations in terms of word meaning/sense, including:
  - Synonymy (same sense across almost entire context, e.g. fear & anxiety)

Lexical Semantics

“Meaning of words!”

Summary

- Complex many-to-many association of senses/meanings and words
  → Polysemy & Homonymy
- There exist different relations in terms of word meaning/sense, including:
  - Synonymy (same sense across almost entire context, e.g. fear & anxiety)
  - Similarity (less close than synonyms, however with a strong thematic association, e.g. fear & scariness)

Lexical Semantics

“Meaning of words!”

Summary

- Complex many-to-many association of senses/meanings and words
  → Polysemy & Homonymy
- There exist different relations in terms of word meaning/sense, including:
  - Synonymy (same sense across almost entire context, e.g. fear & anxiety)
  - Similarity (less close than synonyms, however with a strong thematic association, e.g. fear & scariness)
  - Relatedness (domain-specific semantic relations, fear & panic)

Lexical Semantics

“Meaning of words!”

Summary

- Complex many-to-many association of senses/meanings and words
  → Polysemy & Homonymy
- There exist different relations in terms of word meaning/sense, including:
  - Synonymy (same sense across almost entire context, e.g. fear & anxiety)
  - Similarity (less close than synonyms, however with a strong thematic association, e.g. fear & scariness)
  - Relatedness (domain-specific semantic relations, fear & panic)
  - Antonymy (opposite sense/meaning regarding a specific aspect, but a strong word relatedness, e.g. fear & courage)

Lexical Semantics

“Meaning of words!”

Summary

- Complex many-to-many association of senses/meanings and words
  → Polysemy & Homonymy
- There exist different relations in terms of word meaning/sense, including:
  - Synonymy (same sense across almost entire context, e.g. fear & anxiety)
  - Similarity (less close than synonyms, however with a strong thematic association, e.g. fear & scariness)
  - Relatedness (domain-specific semantic relations, fear & panic)
  - Antonymy (opposite sense/meaning regarding a specific aspect, but a strong word relatedness, e.g. fear & courage)
  - Connotation (affective sense/meaning – sentiment analysis, e.g. fear → negative, courage → positive)

Lexical Semantics

“Meaning of words!”

Approaches in NLP to Identify the “Meaning of Words”

- Lexicographic Tradition – the entire information about word senses is derived from knowledge obtained from lexicons, thesauri, and ontologies

Lexical Semantics

"Meaning of words!"

Approaches in NLP to Identify the "Meaning of Words"

- Lexicographic Tradition – the entire information about word senses is derived from knowledge obtained from lexicons, thesauri, and ontologies
    - Explicit knowledge in terms of word senses/meanings with the assumption that particular words possess discrete word meanings (e.g. "*seal* – marine mammal which lives in the ocean" versus "*seal* – stamp for imprint on an official document")

Lexical Semantics

“Meaning of words!”

Approaches in NLP to Identify the “Meaning of Words”

- Lexicographic Tradition – the entire information about word senses is derived from knowledge obtained from lexicons, thesauri, and ontologies
  - Explicit knowledge in terms of word senses/meanings with the assumption that particular words possess discrete word meanings (e.g. “*seal* – marine mammal which lives in the ocean” versus “*seal* – stamp for imprint on an official document”)
  - Explicit relations between words (e.g. “sailor” works on a “ship”, “bakery” sells “cookies”, “house” consists of several “rooms”, “leaves” are part of a “plant”)

Lexical Semantics

“Meaning of words!”

Approaches in NLP to Identify the “Meaning of Words”

- Lexicographic Tradition – the entire information about word senses is derived from knowledge obtained from lexicons, thesauri, and ontologies
  - Explicit knowledge in terms of word senses/meanings with the assumption that particular words possess discrete word meanings (e.g. “*seal* – marine mammal which lives in the ocean” versus “*seal* – stamp for imprint on an official document”)
  - Explicit relations between words (e.g. “sailor” works on a “ship”, “bakery” sells “cookies”, “house” consists of several “rooms”, “leaves” are part of a “plant”)
- Distributional Tradition – all information about a specific word meaning is derived from the respective contextual environment, using large quantities and volumes of text

Lexical Semantics

“Meaning of words!”

Approaches in NLP to Identify the “Meaning of Words”

- Lexicographic Tradition – the entire information about word senses is derived from knowledge obtained from lexicons, thesauri, and ontologies
    - Explicit knowledge in terms of word senses/meanings with the assumption that particular words possess discrete word meanings (e.g. “*seal* – marine mammal which lives in the ocean” versus “*seal* – stamp for imprint on an official document”)
    - Explicit relations between words (e.g. “sailor” works on a “ship”, “bakery” sells “cookies”, “house” consists of several “rooms”, “leaves” are part of a “plant”)
- Distributional Tradition – all information about a specific word meaning is derived from the respective contextual environment, using large quantities and volumes of text
    - Vector Semantics: map words to a vector representation, using sparse vectors or more contemporary vectorial representations methods, such as dense vectors

Vector Semantics

Type of Vector Representations

Essential Requirement: The vectorial representation must be chosen in a way to be able to model and represent word similarities

Vector Semantics

Type of Vector Representations

Essential Requirement: The vectorial representation must be chosen in a way to be able to model and represent word similarities

- One-Hot-Vector – each word is represented as discrete symbol via a one-hot-encoding:
  - Haus $= [1, 0, 0, \ldots, 0, 0]$, Maus $= [0, 1, 0, \ldots, 0, 0]$, Hund $= [0, 0, 1, \ldots, 0, 0]$, Sonne $= [0, 0, 0, \ldots, 1, 0]$, Katze $= [0, 0, 0, \ldots, 0, 1]$ – Advisable – Yes/No?

Vector Semantics

Type of Vector Representations

Essential Requirement: The vectorial representation must be chosen in a way to be able to model and represent word similarities

- One-Hot-Vector – each word is represented as discrete symbol via a one-hot-encoding:
  - Haus $= [1, 0, 0, \ldots, 0, 0]$, Maus $= [0, 1, 0, \ldots, 0, 0]$, Hund $= [0, 0, 1, \ldots, 0, 0]$, Sonne $= [0, 0, 0, \ldots, 1, 0]$, Katze $= [0, 0, 0, \ldots, 0, 1]$ – Advisable – Yes/No?
  - No! $\rightarrow$ Vocabulary $|V| >> 0$ results in high-dimensional vectors ($|V| \times |V|$ matrix) & word distances between two vectors are identical!

Vector Semantics

Type of Vector Representations

Essential Requirement: The vectorial representation must be chosen in a way to be able to model and represent word similarities

- One-Hot-Vector – each word is represented as discrete symbol via a one-hot-encoding:
  - Haus $= [1, 0, 0, \dots, 0, 0]$, Maus $= [0, 1, 0, \dots, 0, 0]$, Hund $= [0, 0, 1, \dots, 0, 0]$, Sonne $= [0, 0, 0, \dots, 1, 0]$, Katze $= [0, 0, 0, \dots, 0, 1]$ – Advisable – Yes/No?
  - No! → Vocabulary $|V| >> 0$ results in high-dimensional vectors ($|V| \times |V|$ matrix) & word distances between two vectors are identical!
- Sparse Vector – each word is represented as a vector capturing the word-specific context, derived by word co-occurrences (BoW, TF-IDF, N-Gram), to measure word (distributional) similarities:
  - Maus $= [0, 0, 0.2, \dots, 0, 0.3]$, Katze $= [0, 0.4, 0.2, \dots, 0, 0]$, Hund $= [0.2, 0, 0, \dots, 0, 0.4]$,

Vector Semantics

Type of Vector Representations

Essential Requirement: The vectorial representation must be chosen in a way to be able to model and represent word similarities

- One-Hot-Vector – each word is represented as discrete symbol via a one-hot-encoding:
  - Haus $= [1, 0, 0, \ldots, 0, 0]$, Maus $= [0, 1, 0, \ldots, 0, 0]$, Hund $= [0, 0, 1, \ldots, 0, 0]$, Sonne $= [0, 0, 0, \ldots, 1, 0]$, Katze $= [0, 0, 0, \ldots, 0, 1]$ – Advisable – Yes/No?
  - No! → Vocabulary $|V| >> 0$ results in high-dimensional vectors ($|V| \times |V|$ matrix) & word distances between two vectors are identical!
- Sparse Vector – each word is represented as a vector capturing the word-specific context, derived by word co-occurrences (BoW, TF-IDF, N-Gram), to measure word (distributional) similarities:
  - Maus $= [0, 0, 0.2, \ldots, 0, 0.3]$, Katze $= [0, 0.4, 0.2, \ldots, 0, 0]$, Hund $= [0.2, 0, 0, \ldots, 0, 0.4]$,
  - Word matrix still includes mainly 0-values (sparse), high-dimensional vector representations

Vector Semantics

Type of Vector Representations

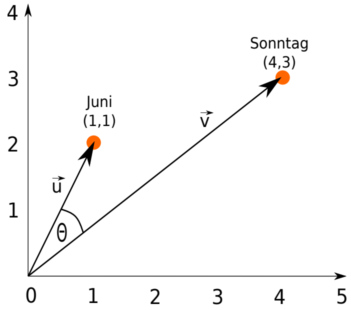
Essential Requirement: The vectorial representation must be chosen in a way to be able to model and represent word similarities

- One-Hot-Vector – each word is represented as discrete symbol via a one-hot-encoding:
  - Haus $= [1, 0, 0, \ldots, 0, 0]$, Maus $= [0, 1, 0, \ldots, 0, 0]$, Hund $= [0, 0, 1, \ldots, 0, 0]$, Sonne $= [0, 0, 0, \ldots, 1, 0]$, Katze $= [0, 0, 0, \ldots, 0, 1]$ – Advisable – Yes/No?
  - No! → Vocabulary $|V| >> 0$ results in high-dimensional vectors ($|V| \times |V|$ matrix) & word distances between two vectors are identical!
- Sparse Vector – each word is represented as a vector capturing the word-specific context, derived by word co-occurrences (BoW, TF-IDF, N-Gram), to measure word (distributional) similarities:
  - Maus $= [0, 0, 0.2, \ldots, 0, 0.3]$, Katze $= [0, 0.4, 0.2, \ldots, 0, 0]$, Hund $= [0.2, 0, 0, \ldots, 0, 0.4]$,
  - Word matrix still includes mainly 0-values (sparse), high-dimensional vector representations
- Dense Vector – compressed (dense) real-valued vector, known as word embeddings, learned by a (deep) neural network and state-of-the-art in all modern NLP algorithms

Vector Semantics

Word as a Vector, How...?

Idea: Transform a word into a vector, called embedding, which describes the "word meaning/sense", while the degree of word similarity (in terms of "meaning") is given by the closeness in the (high-dimensional) semantic space

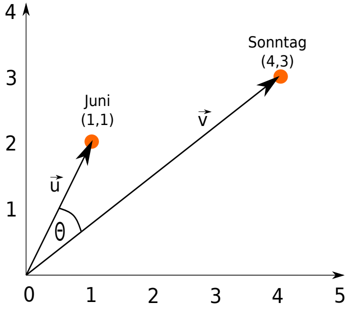


Vector Semantics

Word as a Vector, How...?

Idea: Transform a word into a vector, called embedding, which describes the "word meaning/sense", while the degree of word similarity (in terms of "meaning") is given by the closeness in the (high-dimensional) semantic space

- Similar context leads to similar meanings
  → Distributional Hypothesis

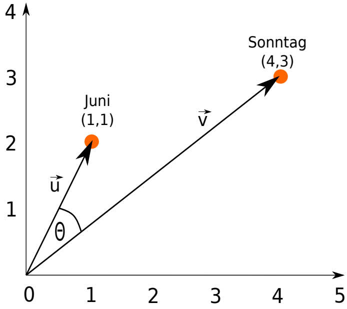


Vector Semantics

Word as a Vector, How...?

Idea: Transform a word into a vector, called embedding, which describes the "word meaning/sense", while the degree of word similarity (in terms of "meaning") is given by the closeness in the (high-dimensional) semantic space

- Similar context leads to similar meanings
  → Distributional Hypothesis
- Similar meanings possess also similar representations

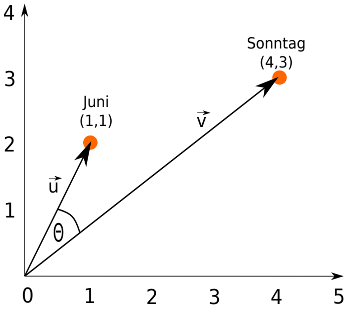


Vector Semantics

Word as a Vector, How...?

Idea: Transform a word into a vector, called embedding, which describes the "word meaning/sense", while the degree of word similarity (in terms of "meaning") is given by the closeness in the (high-dimensional) semantic space

- Similar context leads to similar meanings
  → Distributional Hypothesis
- Similar meanings possess also similar representations
- "Word Similarity" describes vectors which present related directions and magnitudes

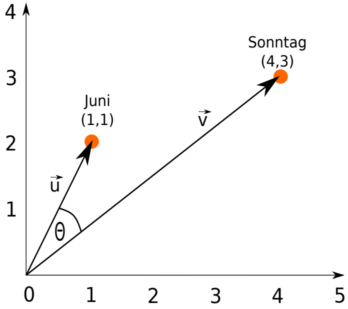


Vector Semantics

Word as a Vector, How...?

Idea: Transform a word into a vector, called embedding, which describes the "word meaning/sense", while the degree of word similarity (in terms of "meaning") is given by the closeness in the (high-dimensional) semantic space

- Similar context leads to similar meanings
  → Distributional Hypothesis
- Similar meanings possess also similar representations
- "Word Similarity" describes vectors which present related directions and magnitudes
- Vectors cover morphological or syntactic word properties (lemma/stem, POS, etc.)



Vector Semantics

Importance of Context

Is context helpful to identify words and categorize the underlying word meaning/sense?

What different words would fit in this context? What is the generic underlying meaning which all the matching words have in common?

“The ??? players skillfully passed the ball to each other.”
“The ??? players engaged in a thrilling rally, hitting the ball back and forth.”
“The ??? player carefully lined up their shot, aiming to hit the ball to a proper direction.”
“The ??? team passed the ball quickly, trying to outmaneuver the opposing defenders”
“The ??? team practiced their passing skills on the field.”

Vector Semantics

Importance of Context

Is context helpful to identify words and categorize the underlying word meaning/sense?

What different words would fit in this context? What is the generic underlying meaning which all the matching words have in common?

“The ??? players skillfully passed the ball to each other.”
“The ??? players engaged in a thrilling rally, hitting the ball back and forth.”
“The ??? player carefully lined up their shot, aiming to hit the ball to a proper direction.”
“The ??? team passed the ball quickly, trying to outmaneuver the opposing defenders”
“The ??? team practiced their passing skills on the field.”

→ Generic underlying word meaning: **“Sport activities with a ball”**

Vector Semantics

Importance of Context

Is context helpful to identify words and categorize the underlying word meaning/sense?

What different words would fit in this context? What is the generic underlying meaning which all the matching words have in common?

"The ??? players skillfully passed the ball to each other."
"The ??? players engaged in a thrilling rally, hitting the ball back and forth."
"The ??? player carefully lined up their shot, aiming to hit the ball to a proper direction."
"The ??? team passed the ball quickly, trying to outmaneuver the opposing defenders"
"The ??? team practiced their passing skills on the field."

→ Generic underlying word meaning: **"Sport activities with a ball"**

Distributional Similarities (Vector-Space Semantics)

- Contextual word information is used to characterize the sense and meaning of words
- Assumption: Similar context → Similar meaning → Similar (vector-space) representation (e.g. soccer, rugby, tennis, golf, handball, basketball, volleyball, etc.)

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Key Concept: Compute the overall word-specific semantic similarity, based on the contextual similarity in which the individual words occur

- How to build these vectors?
  - Every word vector element w_i (dimension) of $\vec{w} = [w_1, w_2, \ldots, w_N] \in \mathbb{R}^N$ is related to a specific context c_i, specifying the degree of association between a word w_N and context c_i
  - Co-occurrence counts between a specific word w_N and context c_i ($\vec{w}$ close to $\vec{u}$ = similar)

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Key Concept: Compute the overall word-specific semantic similarity, based on the contextual similarity in which the individual words occur

- How to build these vectors?
  - Every word vector element w_i (dimension) of $\vec{w} = [w_1, w_2, \ldots, w_N] \in \mathbb{R}^N$ is related to a specific context c_i, specifying the degree of association between a word w_N and context c_i
  - Co-occurrence counts between a specific word w_N and context c_i ($\vec{w}$ close to $\vec{u}$ = similar)
- Bag-of-Words (BoW)/Term Frequency-Inverse Document Frequency (TF-IDF)
  - Term-Document-Matrix (2D): cell specifies for a word w_N (term t) the frequency count f related to document (context) $d_i = c_i \rightarrow$ TF-matrix

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Key Concept: Compute the overall word-specific semantic similarity, based on the contextual similarity in which the individual words occur

- How to build these vectors?
  - Every word vector element w_i (dimension) of $\vec{w} = [w_1, w_2, \ldots, w_N] \in \mathbb{R}^N$ is related to a specific context c_i, specifying the degree of association between a word w_N and context c_i
  - Co-occurrence counts between a specific word w_N and context c_i ($\vec{w}$ close to $\vec{u}$ = similar)
- Bag-of-Words (BoW)/Term Frequency-Inverse Document Frequency (TF-IDF)
  - Term-Document-Matrix (2D): cell specifies for a word w_N (term t) the frequency count f related to document (context) $d_i = c_i \rightarrow$ TF-matrix
  - Column: "A particular document and its vector of counts across all words" $\rightarrow$ How is the word distribution in a single document?

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Key Concept: Compute the overall word-specific semantic similarity, based on the contextual similarity in which the individual words occur

- How to build these vectors?
  - Every word vector element w_i (dimension) of $\vec{w} = [w_1, w_2, \dots, w_N] \in \mathbb{R}^N$ is related to a specific context c_i, specifying the degree of association between a word w_N and context c_i
  - Co-occurrence counts between a specific word w_N and context c_i ($\vec{w}$ close to $\vec{u}$ = similar)
- Bag-of-Words (BoW)/Term Frequency-Inverse Document Frequency (TF-IDF)
  - Term-Document-Matrix (2D): cell specifies for a word w_N (term t) the frequency count f related to document (context) $d_i = c_i \rightarrow$ TF-matrix
  - Column: "A particular document and its vector of counts across all words" → How is the word distribution in a single document?
  - Row: "A particular word and its vector of counts across all documents" → In how many documents the word occurs?

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Term-Document-Matrix (2D)

	DocA	**DocB**	**DocC**	**DocD**
Wizard	123	0	83/83	2
Magic	92	7	112	4
Defend	12	79	8	89
Fight	0	180	6	101

- Similar documents possess similar vector representation (Column!) → DocA & DocC
- Similar words possess similar vector representation (Row!) → Wizard & Magic

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Term-Document-Matrix (2D)

	DocA	**DocB**	**DocC**	**DocD**
Wizard	123	0	83/83	2
Magic	92	7	112	4
Defend	12	79	8	89
Fight	0	180	6	101

- Similar documents possess similar vector representation (Column!) → DocA & DocC
- Similar words possess similar vector representation (Row!) → Wizard & Magic
- How about sparsity in terms of many words and documents?
  → Various dimensionality reduction techniques (reduce size & uninformative elements)

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Term-Document-Matrix (2D)

	DocA	**DocB**	**DocC**	**DocD**
Wizard	123	0	83/83	2
Magic	92	7	112	4
Defend	12	79	8	89
Fight	0	180	6	101

- Similar documents possess similar vector representation (Column!) → DocA & DocC
- Similar words possess similar vector representation (Row!) → Wizard & Magic
- How about sparsity in terms of many words and documents?
  → Various dimensionality reduction techniques (reduce size & uninformative elements)
- Careful when we talk about "context"? How about the contextual distances between individual words – are those important?

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Different Interpretations of “Context”

Count-based techniques rely on the “definition and interpretation of context”, next to the “way of computing the individual matrix elements”

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Different Interpretations of "Context"

Count-based techniques rely on the "definition and interpretation of context", next to the "way of computing the individual matrix elements"

- Context specified by the neighboring words ($\pm k$ words)
  - How often does a word w_N (e.g. "soccer") appear near the context c_i (e.g "ball")?
  - Different interpretations of "near": (1) "soccer" within a window of $\pm k$ words around "ball" – versus – (2) "soccer" within the same document as "ball"

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Different Interpretations of "Context"

Count-based techniques rely on the "definition and interpretation of context", next to the "way of computing the individual matrix elements"

- Context specified by the neighboring words ($\pm k$ words)
  - How often does a word w_N (e.g. "soccer") appear near the context c_i (e.g "ball")?
  - Different interpretations of "near": (1) "soccer" within a window of $\pm k$ words around "ball" – versus – (2) "soccer" within the same document as "ball"

"Yesterday I went out for ice cream after the match and ordered one ball of chocolate. (...) Overall, it was quite impressive during the soccer game, how focused every player was."

→ Look at the above document – are those similarities the same (multiple senses)?

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Different Interpretations of "Context"

Count-based techniques rely on the "definition and interpretation of context", next to the "way of computing the individual matrix elements"

- Context specified by the neighboring words ($\pm k$ words)
  - How often does a word w_N (e.g. "soccer") appear near the context c_i (e.g "ball")?
  - Different interpretations of "near": (1) "soccer" within a window of $\pm k$ words around "ball" – versus – (2) "soccer" within the same document as "ball"

  "Yesterday I went out for ice cream after the match and ordered one ball of chocolate. (...) Overall, it was quite impressive during the soccer game, how focused every player was."

  → Look at the above document – are those similarities the same (multiple senses)?
- Context specified by grammatical relations
  - How often is a context word c_i (noun) used as subject/object of a target word w_N (verb)?
  - "Ball" and "shoot" → more detailed similarities, however, a parsing procedure is required!

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

ChatGPT – Generate a paragraph about soccer and the OTH-university with a quote of Professor Bergler and a total of maximum 50 words!

ChatGPT: “Soccer is a beautiful game that unites people from all walks of life,” stated Professor Bergler, a renowned sport psychologist at OTH University. “Soccer, as a game and sport, does not only build physical strength and teamwork, but also fosters a sense of camaraderie and cultural understanding.” The OTH university recognizes the power of soccer as a beautiful game for personal growth and global integration.

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

ChatGPT – Generate a paragraph about soccer and the OTH-university with a quote of Professor Bergler and a total of maximum 50 words!

ChatGPT: “Soccer is a beautiful game that unites people from all walks of life,” stated Professor Bergler, a renowned sport psychologist at OTH University. “Soccer, as a game and sport, does not only build physical strength and teamwork, but also fosters a sense of camaraderie and cultural understanding.” The OTH university recognizes the power of soccer as a beautiful game for personal growth and global integration.

Build Word-Word Co-Occurrence Matrix

- Specify the vocabulary of **context words** $c_1, \ldots, c_i$ (frequently enough, stop words?)

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How…?

ChatGPT – Generate a paragraph about soccer and the OTH-university with a quote of Professor Bergler and a total of maximum 50 words!

ChatGPT: "Soccer is a beautiful game that unites people from all walks of life," stated Professor Bergler, a renowned sport psychologist at OTH University. "Soccer, as a game and sport, does not only build physical strength and teamwork, but also fosters a sense of camaraderie and cultural understanding." The OTH university recognizes the power of soccer as a beautiful game for personal growth and global integration.

Build Word-Word Co-Occurrence Matrix

- Specify the vocabulary of **context words** $c_1, \ldots, c_i$ (frequently enough, stop words?)
- Specify "**nearby**": "near" is given, if c_i occurs within $\pm\, k$ words around word w_N

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

ChatGPT – Generate a paragraph about soccer and the OTH-university with a quote of Professor Bergler and a total of maximum 50 words!

ChatGPT: "Soccer is a beautiful game that unites people from all walks of life," stated Professor Bergler, a renowned sport psychologist at OTH University. "Soccer, as a game and sport, does not only build physical strength and teamwork, but also fosters a sense of camaraderie and cultural understanding." The OTH university recognizes the power of soccer as a beautiful game for personal growth and global integration.

Build Word-Word Co-Occurrence Matrix

- Specify the vocabulary of **context words** $c_1, \dots, c_i$ (frequently enough, stop words?)
- Specify "**nearby**": "near" is given, if c_i occurs within $\pm k$ words around word w_N
- Specify and **compute co-occurrence** frequencies of words w_N and context words c_i

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

ChatGPT – Generate a paragraph about soccer and the OTH-university with a quote of Professor Bergler and a total of maximum 50 words!

ChatGPT: "Soccer is a beautiful game that unites people from all walks of life," stated Professor Bergler, a renowned sport psychologist at OTH University. "Soccer, as a game and sport, does not only build physical strength and teamwork, but also fosters a sense of camaraderie and cultural understanding." The OTH university recognizes the power of soccer as a beautiful game for personal growth and global integration.

Build Word-Word Co-Occurrence Matrix

- Specify the vocabulary of **context words** $c_1, \ldots, c_i$ (frequently enough, stop words?)
- Specify "**nearby**": "near" is given, if c_i occurs within $\pm k$ words around word w_N
- Specify and **compute co-occurrence** frequencies of words w_N and context words c_i
- Specify **Representation** of the **word-word co-occurrence matrix** (binary, frequency, PMI, ...)

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

ChatGPT – Generate a paragraph about soccer and the OTH-university with a quote of Professor Bergler and a total of maximum 50 words!

ChatGPT: “Soccer is a beautiful game that unites people from all walks of life,” stated Professor Bergler, a renowned sport psychologist at OTH University. “Soccer, as a game and sport, does not only build physical strength and teamwork, but also fosters a sense of camaraderie and cultural understanding.” The OTH university recognizes the power of soccer as a beautiful game for personal growth and global integration.

Build Word-Word Co-Occurrence Matrix

- Specify the vocabulary of **context words** $c_1, \ldots, c_i$ (frequently enough, stop words?)
- Specify “**nearby**”: “near” is given, if c_i occurs within $\pm k$ words around word w_N
- Specify and **compute co-occurrence** frequencies of words w_N and context words c_i
- Specify **Representation** of the **word-word co-occurrence matrix** (binary, frequency, PMI, ...)
- Specify a metric to **compute (word) vector similarity**

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

ChatGPT – Generate a paragraph about soccer and the OTH-university with a quote of Professor Bergler and a total of maximum 50 words!

ChatGPT: "Soccer is a beautiful game that unites people from all walks of life," stated Professor Bergler, a renowned sport psychologist at OTH University. "Soccer, as a game and sport, does not only build physical strength and teamwork, but also fosters a sense of camaraderie and cultural understanding." The OTH university recognizes the power of soccer as a beautiful game for personal growth and global integration.

Build Word-Word Co-Occurrence Matrix

- Specify the vocabulary of **context words** $c_1, \ldots, c_i$ (frequently enough, stop words?)
- Specify "**nearby**": "near" is given, if c_i occurs within $\pm k$ words around word w_N
- Specify and **compute co-occurrence** frequencies of words w_N and context words c_i
- Specify **Representation** of the **word-word co-occurrence matrix** (binary, frequency, PMI, ...)
- Specify a metric to **compute (word) vector similarity**

words/context ($k = \pm 7$)	$c_1 =$ **game**	**university**	**...**	**beautiful**	**sport**
$w_1 =$ **soccer**	$f(w_1, c_1) = 3$	2	...	2	2
...	...	...	...	...	...

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Define co-occurrence frequencies and the word-word co-occurrence matrix representation

- Co-occurrence frequencies
  - Fixed window size: context word c_i present in $\pm k$ words around the target word w_N
  - As part of same sentence → Consider sentence boundaries!
  - Grammatical relations: context word c_i as subject/object of the target word w_N

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Define co-occurrence frequencies and the word-word co-occurrence matrix representation

- Co-occurrence frequencies
  - Fixed window size: context word c_i present in $\pm k$ words around the target word w_N
  - As part of same sentence → Consider sentence boundaries!
  - Grammatical relations: context word c_i as subject/object of the target word w_N
- Word-Word co-occurrence matrix representation
  - $f(w, c)$ as binary value → context word c_i present $(= 1 \mid 0)$ around the target word w_N

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Define co-occurrence frequencies and the word-word co-occurrence matrix representation

- Co-occurrence frequencies
  - Fixed window size: context word c_i present in $\pm k$ words around the target word w_N
  - As part of same sentence → Consider sentence boundaries!
  - Grammatical relations: context word c_i as subject/object of the target word w_N
- Word-Word co-occurrence matrix representation
  - $f(w, c)$ as binary value → context word c_i present $(= 1 \mid 0)$ around the target word w_N
  - $f(w, c)$ as frequency values → context word c_i (N times) around the target word w_N

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Define co-occurrence frequencies and the word-word co-occurrence matrix representation

- Co-occurrence frequencies
  - Fixed window size: context word c_i present in $\pm k$ words around the target word w_N
  - As part of same sentence → Consider sentence boundaries!
  - Grammatical relations: context word c_i as subject/object of the target word w_N
- Word-Word co-occurrence matrix representation
  - $f(w, c)$ as binary value → context word c_i present $(= 1 \mid 0)$ around the target word w_N
  - $f(w, c)$ as frequency values → context word c_i (N times) around the target word w_N
  - $f(w, c)$ as probability values → probability of context word c_i around the target word w_N

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Positive Pointwise Mutual Information (PPMI)

- Co-occurrences with only few counts are traditionally more informative, compared to high co-occurrence counts
- Most of the high word co-occurrences are in combination with "stop words" and less informative regarding the word meaning
- Goal: Identification of co-occurrence counts being larger than expected by chance

$$PPMI(w,c) = \begin{cases} log \frac{p(w,c)}{p(w)p(c)} = \frac{\frac{f(w,c)}{\sum_{i=1}^{N}\sum_{j=1}^{M} f(w_i,c_j)}}{\frac{f(w)}{|V|}\frac{f(c)}{|V|}} & \text{if} > 0 \\ 0 & \text{if} \leq 0 \end{cases}$$

with: $|V| =$ Vocabulary ($\#$ tokens in corpus), N number of words w_i, M number of context words c_j, $f(w) = \sum_c f(w,c)$, $f(c) = \sum_w f(w,c)$

Distributional Similarities (Vector-Space Semantics)

Transform Words into Sparse Vectors – How...?

Word-Word Co-Occurrence Matrix Representation

- Binary Values $f(w, c)$

words/context ($k = \pm 7$)	**game**	**university**	**smart**	**beautiful**	**sport**
soccer	1	1	0	1	1

- Frequency Values $f(w, c)$

words/context ($k = \pm 7$)	**game**	**university**	**smart**	**beautiful**	**sport**
soccer	3	2	0	2	2

- Probability Values PPMI(w, c)

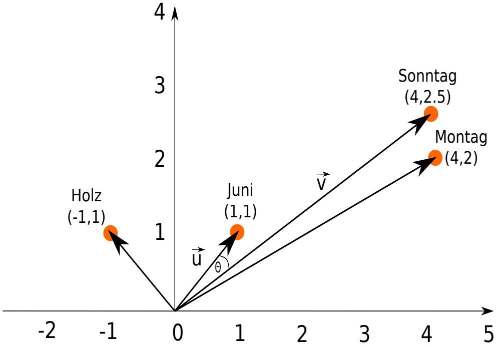
words/context ($k = \pm 7$)	**game**	**university**	**smart**	**beautiful**	**sport**	**p(word w_N)**
soccer	0.33	0.22	0.00	0.22	0.22	1.00
p(context c_i)	0.33	0.22	0.00	0.22	0.22	1.00

- PPMI biased in terms of infrequent events → Smoothing!

Distributional Similarities (Vector-Space Semantics)

Vector Similarity

- "Word Similarity" describes vectors which present related directions and magnitudes

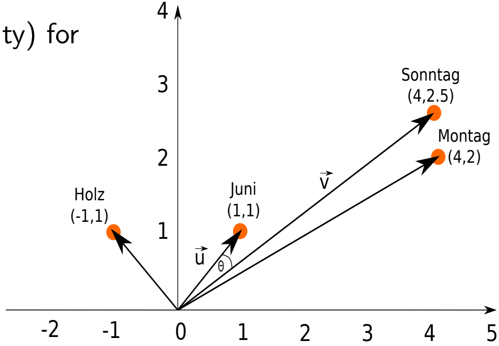


Distributional Similarities (Vector-Space Semantics)

Vector Similarity

- "Word Similarity" describes vectors which present related directions and magnitudes
- Dot-product ($\vec{u} \cdot \vec{v} = \sum_{i=1}^{N} u_i v_i$) as similarity metric

  → Becomes very large (indicator of a high similarity) for long vectors – Better metric?



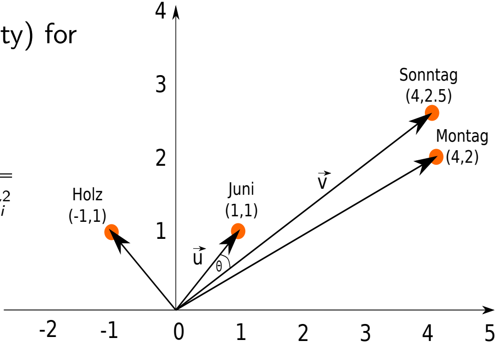
Distributional Similarities (Vector-Space Semantics)

Vector Similarity

- "Word Similarity" describes vectors which present related directions and magnitudes
- Dot-product ($\vec{u} \cdot \vec{v} = \sum_{i=1}^{N} u_i v_i$) as similarity metric

  $\rightarrow$ Becomes very large (indicator of a high similarity) for long vectors – Better metric?
- Cosine Similarity:

$$cos(\Theta) = cos(\vec{u}, \vec{v}) = \frac{\vec{u}\,\vec{v}}{||\vec{u}||\;||\vec{v}||} = \frac{\sum_{i=1}^{N} u_i v_i}{\sqrt{\sum_{i=1}^{N} u_i^2}\sqrt{\sum_{i=1}^{N} v_i^2}}$$



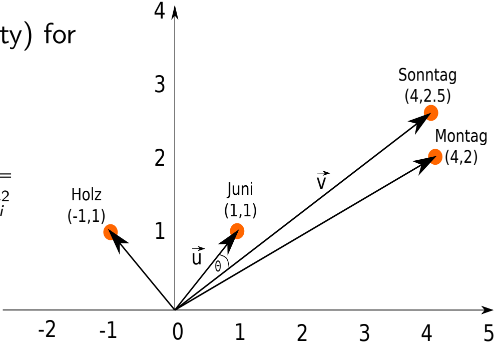
Distributional Similarities (Vector-Space Semantics)

Vector Similarity

- "Word Similarity" describes vectors which present related directions and magnitudes
- Dot-product ($\vec{u} \cdot \vec{v} = \sum_{i=1}^{N} u_i v_i$) as similarity metric
  $\rightarrow$ Becomes very large (indicator of a high similarity) for long vectors – Better metric?
- Cosine Similarity:

$$cos(\Theta) = cos(\vec{u}, \vec{v}) = \frac{\vec{u}\,\vec{v}}{||\vec{u}||\,||\vec{v}||} = \frac{\sum_{i=1}^{N} u_i v_i}{\sqrt{\sum_{i=1}^{N} u_i^2}\sqrt{\sum_{i=1}^{N} v_i^2}}$$

- $cos(\Theta) = cos(\vec{u}, \vec{v}) = 1 \rightarrow$ same direction!
- $cos(\Theta) = cos(\vec{u}, \vec{v}) = 0 \rightarrow$ orthogonal!
- $cos(\Theta) = cos(\vec{u}, \vec{v}) = -1 \rightarrow$ opposite direction!
  $\rightarrow$ Only directional information without magnitude!



Distributional Similarities (Vector-Space Semantics)

Transform Words into Dense Vectors, How...?

Dense Vector – (Static) Word Embeddings

- Recap: Vectors of the term-document matrix and/or word-word co-occurrence matrix (binary, frequency-based, PPMI) are generally extremely long ($|V|$) and sparse (lot of zero elements)

Distributional Similarities (Vector-Space Semantics)

Transform Words into Dense Vectors, How...?

Dense Vector – (Static) Word Embeddings

- Recap: Vectors of the term-document matrix and/or word-word co-occurrence matrix (binary, frequency-based, PPMI) are generally extremely long ($|V|$) and sparse (lot of zero elements)
- Dense vector representations possess a lower dimensionality than the actual vocabulary size $|V|$ ($< 1{,}000$ elements) with mostly non-zero elements

Distributional Similarities (Vector-Space Semantics)

Transform Words into Dense Vectors, How...?

Dense Vector – (Static) Word Embeddings

- Recap: Vectors of the term-document matrix and/or word-word co-occurrence matrix (binary, frequency-based, PPMI) are generally extremely long ($|V|$) and sparse (lot of zero elements)
- Dense vector representations possess a lower dimensionality than the actual vocabulary size $|V|$ ($<$ 1,000 elements) with mostly non-zero elements
- Dense vectors allow a better model training procedure → Less parameters to optimize and better generalization ("learn, rather than count!")

Distributional Similarities (Vector-Space Semantics)

Transform Words into Dense Vectors, How...?

Dense Vector – (Static) Word Embeddings

- Recap: Vectors of the term-document matrix and/or word-word co-occurrence matrix (binary, frequency-based, PPMI) are generally extremely long ($|V|$) and sparse (lot of zero elements)
- Dense vector representations possess a lower dimensionality than the actual vocabulary size $|V|$ ($<$ 1,000 elements) with mostly non-zero elements
- Dense vectors allow a better model training procedure → Less parameters to optimize and better generalization ("learn, rather than count!")
- Mapping function ignores that a particular word may possess multiple meanings/senses (single vector in the semantic space → Static word embedding for each word!)
  → Polysemy and Homonymy are not covered properly!

Distributional Similarities (Vector-Space Semantics)

Transform Words into Dense Vectors, How…?

Dense Vector – (Static) Word Embeddings

- Recap: Vectors of the term-document matrix and/or word-word co-occurrence matrix (binary, frequency-based, PPMI) are generally extremely long ($|V|$) and sparse (lot of zero elements)
- Dense vector representations possess a lower dimensionality than the actual vocabulary size $|V|$ ($<$ 1,000 elements) with mostly non-zero elements
- Dense vectors allow a better model training procedure $\rightarrow$ Less parameters to optimize and better generalization ("learn, rather than count!")
- Mapping function ignores that a particular word may possess multiple meanings/senses (single vector in the semantic space $\rightarrow$ Static word embedding for each word!) $\rightarrow$ Polysemy and Homonymy are not covered properly!
- Vocabulary size fixed ($<$UNK$>$ token needed!)

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec (Two-Layer Neural Net)

- Mikolov et al., *"Efficient estimation of word representations in vector space"*, 2013
  - Popular, fast to train, available code, and pre-trained embeddings, trained on huge corpora
  - Word2Vec: https://code.google.com/archive/p/word2vec/

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec (Two-Layer Neural Net)

- Mikolov et al., *"Efficient estimation of word representations in vector space"*, 2013
  - Popular, fast to train, available code, and pre-trained embeddings, trained on huge corpora
  - Word2Vec: https://code.google.com/archive/p/word2vec/
- Two Word2Vec variants: Skip-Gram versus Continuous Bag-of-Words (CBOW)
  - Skip-Gram: a given central word is used to predict the neighboring context (fixed $\pm k$)
    → "... fascinating soccer game last Saturday ..."
  - CBOW: a given neighboring context is used and combined (summation of context word vectors → BoW) to predict a central target word (fixed $\pm k$)
    → "... fascinating soccer game last Saturday ..."

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec (Two-Layer Neural Net)

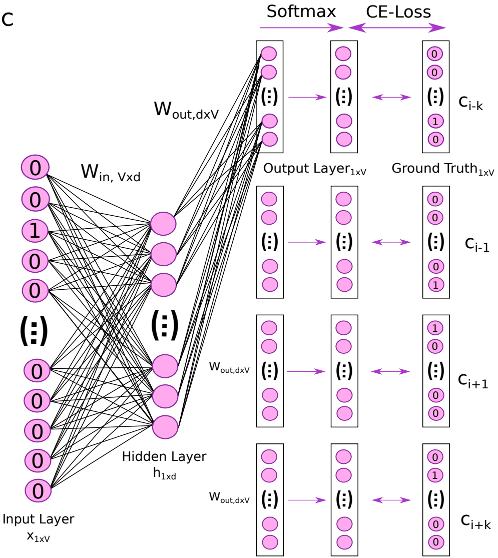
- Mikolov et al., *"Efficient estimation of word representations in vector space"*, 2013
  - Popular, fast to train, available code, and pre-trained embeddings, trained on huge corpora
  - Word2Vec: https://code.google.com/archive/p/word2vec/
- Two Word2Vec variants: Skip-Gram versus Continuous Bag-of-Words (CBOW)
  - Skip-Gram: a given central word is used to predict the neighboring context (fixed $\pm k$) → "… fascinating soccer game last Saturday …"
  - CBOW: a given neighboring context is used and combined (summation of context word vectors → BoW) to predict a central target word (fixed $\pm k$) → "… fascinating soccer game last Saturday …"
  - Two optimization objectives: Negative sampling (NS) vs. Hierarchical Softmax
- Focus: Skip-Gram, together with negative sampling, as the most popular approach (Mikolov et al., *"Distributed Representations of Words and Phrases and their Compositionality", 2013*)

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Main Idea

- Prediction of neighboring word vectors c_i with $\pm k$ as the surrounding window (context), near a given target/center word t

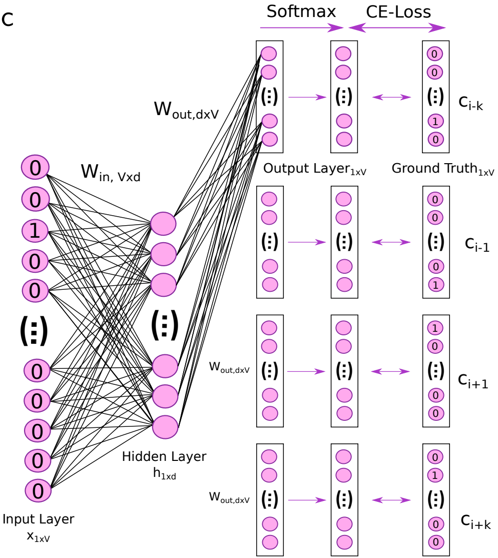


Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Main Idea

- Prediction of neighboring word vectors c_i with $\pm k$ as the surrounding window (context), near a given target/center word t
- Two-layer network:
  - Network input: one-hot vector of the center/target word $t_{V \times 1}$
  - Network output: context predictions $c_{i,V \times 1}$ in $\pm k$
- Two-layer-specific weight matrices:
  - Input matrix: $W_{in,d \times V}$ ($=$ word embedding matrix)
  - Output matrix (shared weights for k): $W_{out,V \times d}$

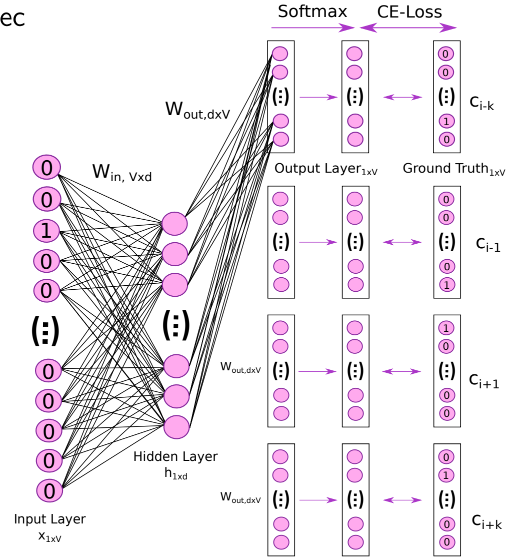


Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Main Idea

- Objective: $\frac{1}{T}\sum_{t=1}^{T}\sum_{-k\leq j\leq k, j\neq 0} log\ p(w_{t+j}|w_t)$
  - k as the context size (left/right)
  - T batch size $\rightarrow$ $T = |V|$ batch gradient descent, $T = 1$ stochastic gradient descent

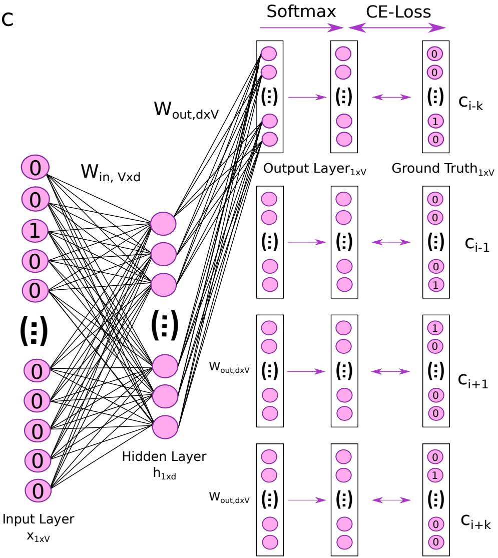


Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Main Idea

- Objective: $\frac{1}{T}\sum_{t=1}^{T}\sum_{-k\leq j\leq k, j\neq 0} log\ p(w_{t+j}|w_t)$
  - k as the context size (left/right)
  - T batch size $\rightarrow$ $T = |V|$ batch gradient descent, $T = 1$ stochastic gradient descent
- Loss: $J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-k\leq j\leq k, j\neq 0} log\ p(w_{t+j}|w_t;\theta)$

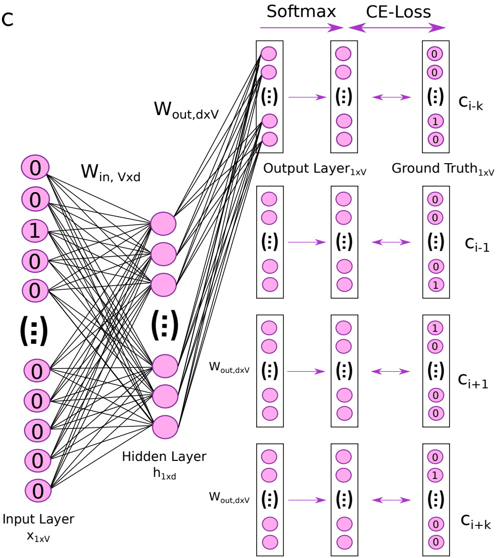


Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Main Idea

- Objective: $\frac{1}{T}\sum_{t=1}^{T}\sum_{-k\leq j\leq k, j\neq 0} log\ p(w_{t+j}|w_t)$
    - k as the context size (left/right)
    - T batch size $\rightarrow$ $T = |V|$ batch gradient descent, $T = 1$ stochastic gradient descent
- Loss: $J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-k\leq j\leq k, j\neq 0} log\ p(w_{t+j}|w_t;\theta)$
    - $log\ p(w_{t+j}|w_t;\theta) = \frac{e^{h^T \cdot W_{out,c_{i-k}}}}{\sum_{i=1}^{|V|} e^{h^T \cdot W_{out,w_i}}} = Softmax$

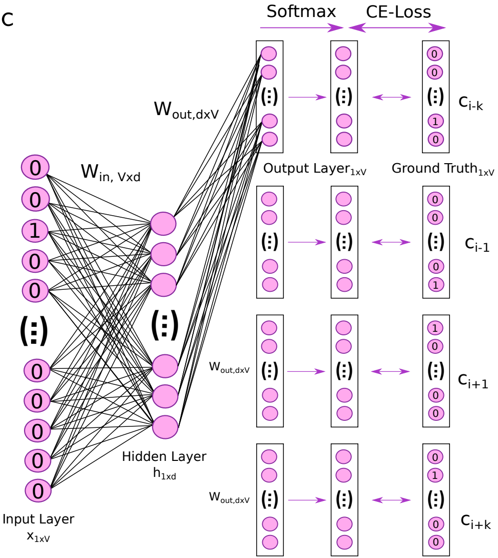


Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Main Idea

- Objective: $\frac{1}{T}\sum_{t=1}^{T}\sum_{-k\leq j\leq k, j\neq 0} log\ p(w_{t+j}|w_t)$
  - k as the context size (left/right)
  - T batch size $\rightarrow$ $T = |V|$ batch gradient descent, $T = 1$ stochastic gradient descent
- Loss: $J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-k\leq j\leq k, j\neq 0} log\ p(w_{t+j}|w_t;\theta)$
  - $log\ p(w_{t+j}|w_t;\theta) = \frac{e^{h^T \cdot W_{out,c_{i-k}}}}{\sum_{i=1}^{|V|} e^{h^T \cdot W_{out,w_i}}} = Softmax$
- Params: $\theta = [W_{in,d\times V}, W_{out,V\times d}] \rightarrow 2\cdot|V|\cdot d$

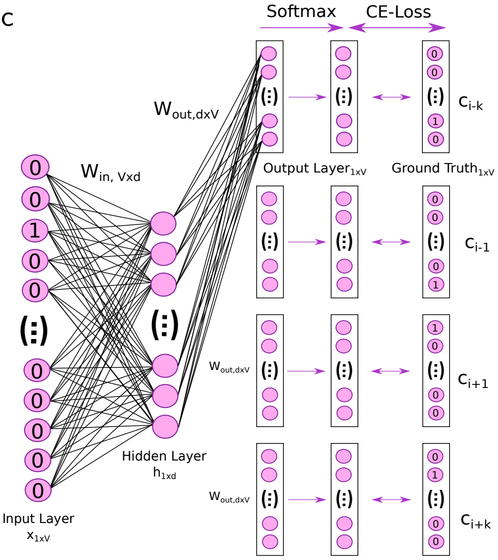


Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

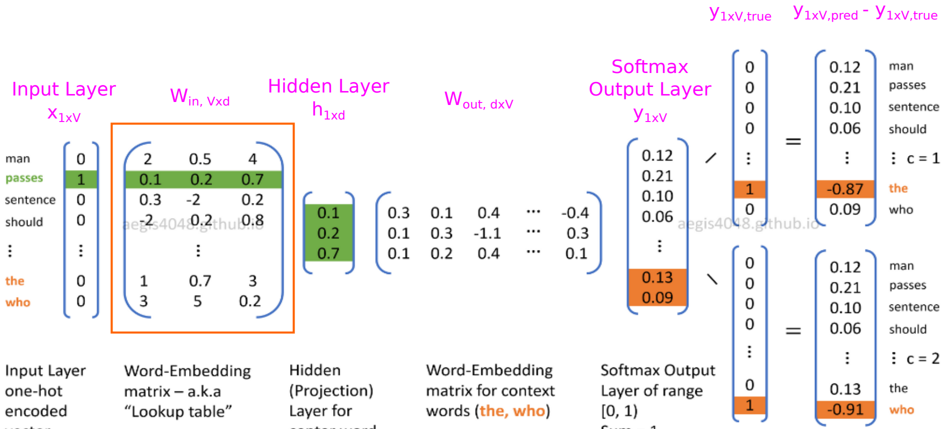
Word2Vec – Skip-Gram – Main Idea

- Objective: $\frac{1}{T}\sum_{t=1}^{T}\sum_{-k\leq j\leq k, j\neq 0} log\ p(w_{t+j}|w_t)$
  - k as the context size (left/right)
  - T batch size $\rightarrow$ $T = |V|$ batch gradient descent, $T = 1$ stochastic gradient descent
- Loss: $J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-k\leq j\leq k, j\neq 0} log\ p(w_{t+j}|w_t;\theta)$
  - $log\ p(w_{t+j}|w_t;\theta) = \frac{e^{h^T \cdot W_{out,c_{i-k}}}}{\sum_{i=1}^{|V|} e^{h^T \cdot W_{out,w_i}}} = Softmax$
- Params: $\theta = [W_{in,d\times V}, W_{out,V\times d}] \rightarrow 2\cdot|V|\cdot d$
- $J(\theta)$ is the sum of all context-related CE-losses



Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec



$$\begin{array}{l|c|}
\text{man} & 0\\
\text{passes} & 1\\
\text{sentence} & 0\\
\text{should} & 0\\
\vdots & \vdots\\
\text{the} & 0\\
\text{who} & 0
\end{array}
\quad
\begin{pmatrix}
2 & 0.5 & 4\\
0.1 & 0.2 & 0.7\\
0.3 & -2 & 0.2\\
-2 & 0.2 & 0.8\\
 & \vdots & \\
1 & 0.7 & 3\\
3 & 5 & 0.2
\end{pmatrix}
\quad
\begin{bmatrix}0.1\\0.2\\0.7\end{bmatrix}
\quad
\begin{bmatrix}
0.3 & 0.1 & 0.4 & \cdots & -0.4\\
0.1 & 0.3 & -1.1 & \cdots & 0.3\\
0.1 & 0.2 & 0.4 & \cdots & 0.1
\end{bmatrix}
\quad
\begin{bmatrix}0.12\\0.21\\0.10\\0.06\\\vdots\\0.13\\0.09\end{bmatrix}$$

$$/ \begin{bmatrix}0\\0\\0\\0\\\vdots\\1\\0\end{bmatrix} = \begin{bmatrix}0.12\\0.21\\0.10\\0.06\\\vdots\\-0.87\\0.09\end{bmatrix}\begin{array}{l}\text{man}\\\text{passes}\\\text{sentence}\\\text{should}\\\vdots\ c=1\\\text{the}\\\text{who}\end{array}$$

$$\backslash \begin{bmatrix}0\\0\\0\\0\\\vdots\\0\\1\end{bmatrix} = \begin{bmatrix}0.12\\0.21\\0.10\\0.06\\\vdots\\0.13\\-0.91\end{bmatrix}\begin{array}{l}\text{man}\\\text{passes}\\\text{sentence}\\\text{should}\\\vdots\ c=2\\\text{the}\\\text{who}\end{array}$$

Input Layer one-hot encoded vector

Word-Embedding matrix – a.k.a "Lookup table"

Hidden (Projection) Layer for center word (**passes**)

Word-Embedding matrix for context words (**the, who**)

Softmax Output Layer of range [0, 1) Sum = 1

Prediction Error – a.k.a function loss

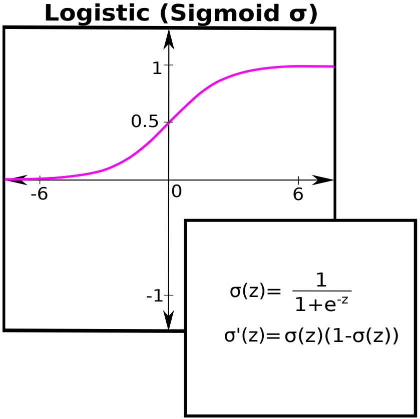
Source: https://aegis4048.github.io/demystifying_neural_network_in_skip_gram_language_modeling#negsample

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Negative Sampling

- Softmax is computationally very expensive → $\approx O(V)$



$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$\sigma'(z) = \sigma(z)(1-\sigma(z))$$

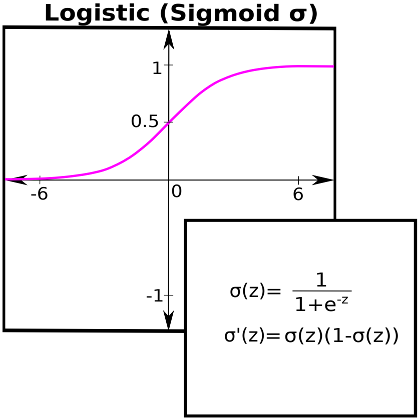
Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Negative Sampling

- Softmax is computationally very expensive $\rightarrow \approx O(V)$
- Change approach from "predicting the neighboring context words c_i of t" to "are two words neighbors"?

  $\rightarrow$ Change: Multi-class classification scenario (via Softmax) to binary classification, using σ-function



$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$\sigma'(z) = \sigma(z)(1-\sigma(z))$$

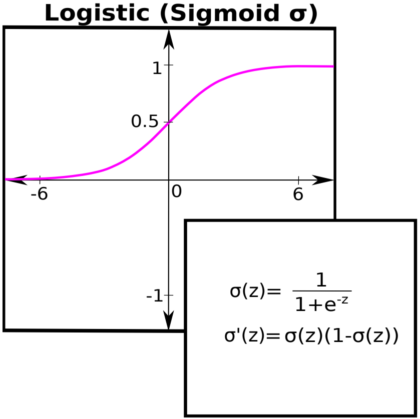
Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Negative Sampling

- Softmax is computationally very expensive $\rightarrow \approx O(V)$
- Change approach from "predicting the neighboring context words c_i of t" to "are two words neighbors"?

  $\rightarrow$ Change: Multi-class classification scenario (via Softmax) to binary classification, using σ-function
- Positive training samples $P(X = 1|t, c_{i,pos})$ in the neighborhood of $\pm k$ (e.g. $t =$"soccer", $c_{1,pos} =$"game", $c_{2,pos} =$"university", etc.)



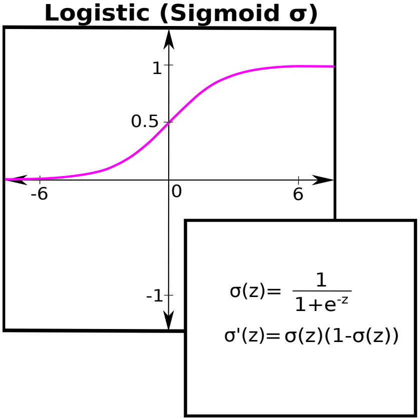
Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Negative Sampling

- Softmax is computationally very expensive → $\approx O(V)$
- Change approach from “predicting the neighboring context words c_i of t” to “are two words neighbors”?

  → Change: Multi-class classification scenario (via Softmax) to binary classification, using σ-function
- Positive training samples $P(X = 1 | t, c_{i,pos})$ in the neighborhood of $\pm k$ (e.g. $t =$“soccer”, $c_{1,pos} =$“game”, $c_{2,pos} =$“university”, etc.)
- Negative training samples $P(X = 0 | t, c_{i,neg})$, sampled from a noise distribution $P_n(w) = \frac{U(w)^{3/4}}{Z}$
  - Unigram distribution $U(w)$, Z normalization factor



$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$\sigma'(z) = \sigma(z)(1-\sigma(z))$$

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Negative Sampling

- Instead of computing $|V|$ binary classifier outputs, a handful of training samples (one positive and M negatives) is used during parameter update/optimization $\rightarrow O(M+1)$

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Negative Sampling

- Instead of computing $|V|$ binary classifier outputs, a handful of training samples (one positive and M negatives) is used during parameter update/optimization $\rightarrow O(M+1)$
- Recommend $M = 2 - 20$ (according to Mikolov et. al)

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Negative Sampling

- Instead of computing $|V|$ binary classifier outputs, a handful of training samples (one positive and M negatives) is used during parameter update/optimization → $O(M+1)$
- Recommend $M = 2 - 20$ (according to Mikolov et. al)
- Sigmoid $\sigma(z) = 1 - \sigma(-z)$

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Negative Sampling

- Instead of computing $|V|$ binary classifier outputs, a handful of training samples (one positive and M negatives) is used during parameter update/optimization $\rightarrow O(M+1)$
- Recommend $M = 2 - 20$ (according to Mikolov et. al)
- Sigmoid $\sigma(z) = 1 - \sigma(-z)$
- Loss: $J(\theta) = -log\, \sigma(t \cdot c_{i,pos}) - \sum_{i=1}^{M} \mathbb{E}_{c_{i,neg} \sim P_n(c)} \left[log\, \sigma(-t \cdot c_{i,neg}) \right]$
  with $t = h^T$, together with $c_{i,pos} = w_{c_{i,pos}} \in W_{out}$, and $c_{i,neg} = w_{c_{i,neg}} \in W_{out}$

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Negative Sampling

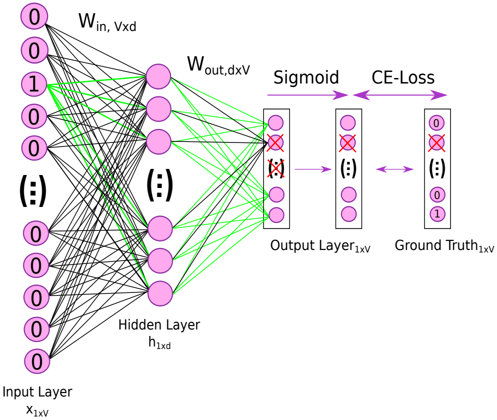
- Instead of computing $|V|$ binary classifier outputs, a handful of training samples (one positive and M negatives) is used during parameter update/optimization → $O(M+1)$
- Recommend $M = 2 - 20$ (according to Mikolov et. al)
- Sigmoid $\sigma(z) = 1 - \sigma(-z)$
- Loss: $J(\theta) = -log\,\sigma(t \cdot c_{i,pos}) - \sum_{i=1}^{M} \mathbb{E}_{c_{i,neg} \sim P_n(c)} \left[log\,\sigma(-t \cdot c_{i,neg}) \right]$ with $t = h^T$, together with $c_{i,pos} = w_{c_{i,pos}} \in W_{out}$, and $c_{i,neg} = w_{c_{i,neg}} \in W_{out}$
- Probability of discarding frequent words during sub-sampling: $P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$ with $f(w_i)$ as the relative word frequency and threshold t (typically 10^{-5})

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Negative Sampling

- Assumption: $|V| = 3,000,000, d = 100 \rightarrow \theta = 2 \cdot |V| \cdot d = 600,000,000$ million parameter for optimization!!!

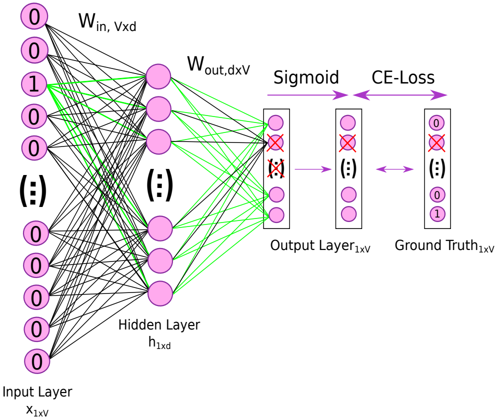


Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Negative Sampling

- Assumption: $|V| = 3,000,000, d = 100 \rightarrow \theta = 2 \cdot |V| \cdot d = 600,000,000$ million parameter for optimization!!!
- Irrespective of negative sampling, one-hot input $x_{1 \times V}$ leads to a parametric optimization of $d = 100$ weights in $W_{in, V \times d}$ for a target word t

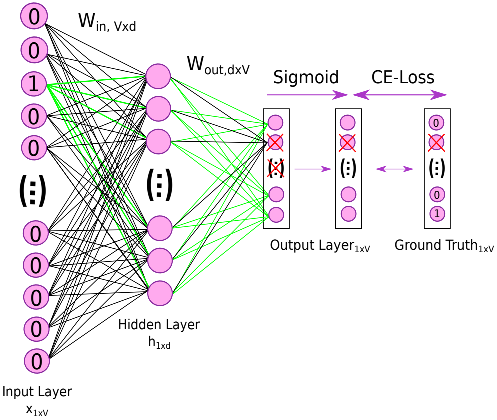


Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Negative Sampling

- Assumption: $|V| = 3,000,000, d = 100 \rightarrow \theta = 2 \cdot |V| \cdot d = 600,000,000$ million parameter for optimization!!!
- Irrespective of negative sampling, one-hot input $x_{1 \times V}$ leads to a parametric optimization of $d = 100$ weights in $W_{in, V \times d}$ for a target word t
- Negative sampling ($M = 2$): one positive, two negative samples $\rightarrow$ additional parametric reduction, only another 300 weights are active (similar to dropout)

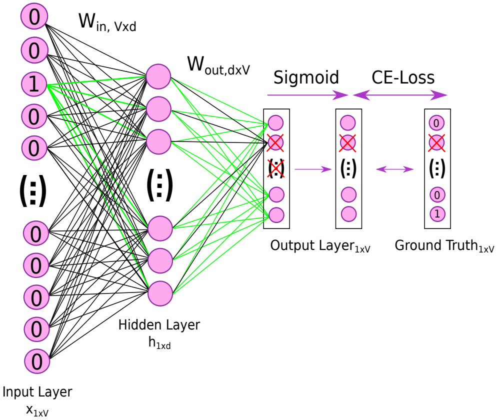


Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec

Word2Vec – Skip-Gram – Negative Sampling

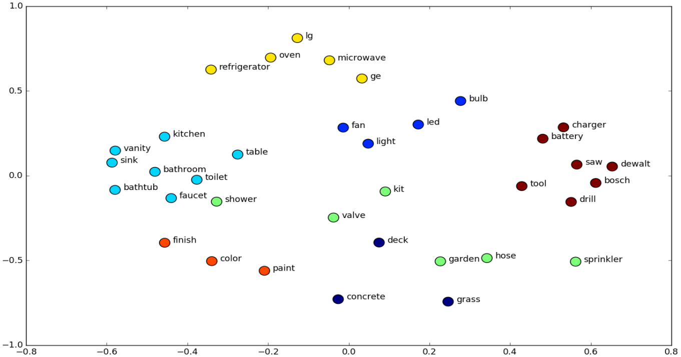
- Assumption: $|V| = 3,000,000, d = 100 \rightarrow \theta = 2 \cdot |V| \cdot d = 600,000,000$ million parameter for optimization!!!
- Irrespective of negative sampling, one-hot input $x_{1 \times V}$ leads to a parametric optimization of $d = 100$ weights in $W_{in, V \times d}$ for a target word t
- Negative sampling ($M = 2$): one positive, two negative samples $\rightarrow$ additional parametric reduction, only another 300 weights are active (similar to dropout)
- From originally $\theta = 600,000,000$ parameters per backpropagation to a total of $\theta = 400$



Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – Word2Vec: 2D-Embedding

- How to visualize a 1 × *d*-large vector in a 2D/3D-space?
- Key-Word: Dimensionality Reduction (e.g. PCA, t-SNE, ... part of "Machine Learning")



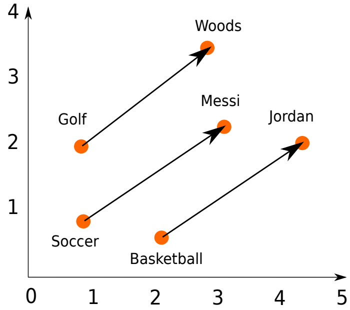
Source: https://neptune.ai/blog/word-embeddings-guide

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings

Embedding Properties

- “Word similarity” is dependent on the window size $\pm k$
  - Small window length (e.g. $k = 2$): nearest words illustrated syntactical similarities of the same taxonomy (e.g. Golf, Soccer, Basketball)
  - Large window length (e.g. $k = 10$): nearest words describe words with related semantic characteristics (e.g. Golf, Soccer, Basketball)

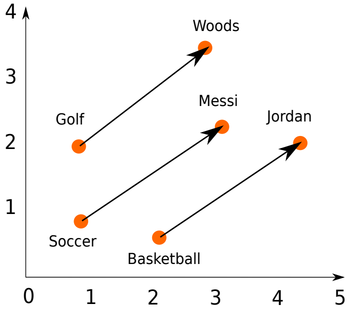


Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings

Embedding Properties

- “Word similarity” is dependent on the window size $\pm k$
  - Small window length (e.g. $k = 2$): nearest words illustrated syntactical similarities of the same taxonomy (e.g. Golf, Soccer, Basketball)
  - Large window length (e.g. $k = 10$): nearest words describe words with related semantic characteristics (e.g. Golf, Soccer, Basketball)
- Word embeddings capture relational meanings (analogy)
  - “Golf” to “Woods” is “Basketball” to “Jordan”
  - $\vec{w_1}(Basketball) - \vec{w_2}(Golf) + \vec{w_3}(Woods) = \vec{w_4}(Jordan)$



Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe)

- Paper: Pennington et al., *"GloVe: Global vectors for word representation"*, 2014
  - GloVe: http://nlp.stanford.edu/projects/glove/

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe)

- Paper: Pennington et al., "*GloVe: Global vectors for word representation*", 2014
  - GloVe: http://nlp.stanford.edu/projects/glove/
- Combines global count-based information, in addition to a prediction-/learning-based (*Word2Vec*) approach, which covers only local properties in the window $\pm k$

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe)

- Paper: Pennington et al., “*GloVe: Global vectors for word representation*”, 2014
  - GloVe: http://nlp.stanford.edu/projects/glove/
- Combines global count-based information, in addition to a prediction-/learning-based (*Word2Vec*) approach, which covers only local properties in the window $\pm k$
- Global information represented in the Word-Word Co-Occurrence Matrix

words	x_1 = **soccer**	**...**	**...**	**...**	x_j =**sport**
x_1 = **soccer**	0	3	...	4	2
...	...	...	...	...	...
x_i = **sport**	0	...	...	...	$X_{ij} = f(x_i, x_j) = 3$

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe)

- Paper: Pennington et al., *"GloVe: Global vectors for word representation"*, 2014
  - GloVe: http://nlp.stanford.edu/projects/glove/
- Combines global count-based information, in addition to a prediction-/learning-based (*Word2Vec*) approach, which covers only local properties in the window $\pm k$
- Global information represented in the Word-Word Co-Occurrence Matrix

words	$x_1 =$ **soccer**	**...**	**...**	**...**	$x_j =$**sport**
$x_1 =$ **soccer**	0	3	...	4	2
...	...	...	...	...	...
$x_i =$ **sport**	0	...	...	...	$X_{ij} = f(x_i, x_j) = 3$

- $P(x_j|x_i) = \frac{X_{ij}}{X_i = \sum_k X_{ik}}$: Probability of x_j being in the $\pm k$-large windowed context of x_i

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe)

- Paper: Pennington et al., *"GloVe: Global vectors for word representation"*, 2014
  - GloVe: http://nlp.stanford.edu/projects/glove/
- Combines global count-based information, in addition to a prediction-/learning-based (*Word2Vec*) approach, which covers only local properties in the window $\pm k$
- Global information represented in the Word-Word Co-Occurrence Matrix

words	$x_1 =$ **soccer**	**…**	**…**	**…**	$x_j =$**sport**
$x_1 =$ **soccer**	0	3	…	4	2
…	…	…	…	…	…
$x_i =$ **sport**	0	…	…	…	$X_{ij} = f(x_i, x_j) = 3$

- $P(x_j|x_i) = \frac{X_{ij}}{X_i = \sum_k X_{ik}}$: Probability of x_j being in the $\pm k$-large windowed context of x_i
- $i, j = |V| \rightarrow W_{V \times V}$ matrix $\rightarrow$ Symmetry (Hint: X_{ij} vs. X_{ji})? $W = W^T$?

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe)

- Idea of "word ratios": better than "raw probabilities" especially in detecting relevant vs. irrelevant words

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe)

- Idea of "word ratios": better than "raw probabilities" especially in detecting relevant vs. irrelevant words

center/context words	k=**solid**	k=**gas**	k=**water**	**...**	k=**fashion**
$w_i =$ **ice**, $P(k\|ice)$	high	small	high	(...)	low
$w_j =$ **steam**, $P(k\|steam)$	small	high	high	(...)	low
$\frac{P(k\|ice)}{P(k\|steam)}$	high	small	≈ 1	(...)	≈ 1

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe)

- Idea of “word ratios”: better than “raw probabilities” especially in detecting relevant vs. irrelevant words

center/context words	k=**solid**	k=**gas**	k=**water**	**...**	k=**fashion**
$w_i =$ **ice**, $P(k\|ice)$	high	small	high	(...)	low
$w_j =$ **steam**, $P(k\|steam)$	small	high	high	(...)	low
$\frac{P(k\|ice)}{P(k\|steam)}$	high	small	≈ 1	(...)	≈ 1

- Problem? → Dimensionality!

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe)

- Idea of "word ratios": better than "raw probabilities" especially in detecting relevant vs. irrelevant words

center/context words	k=**solid**	k=**gas**	k=**water**	**...**	k=**fashion**
$w_i =$ **ice**, $P(k\|ice)$	high	small	high	(...)	low
$w_j =$ **steam**, $P(k\|steam)$	small	high	high	(...)	low
$\frac{P(k\|ice)}{P(k\|steam)}$	high	small	≈ 1	(...)	≈ 1

- Problem? → Dimensionality!
- Question: "How to use the power of word-word co-occurrence without a direct computation?"

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe)

- Idea of “word ratios”: better than “raw probabilities” especially in detecting relevant vs. irrelevant words

center/context words	k=**solid**	k=**gas**	k=**water**	**...**	k=**fashion**
$w_i =$ **ice**, $P(k\|ice)$	high	small	high	(...)	low
$w_j =$ **steam**, $P(k\|steam)$	small	high	high	(...)	low
$\frac{P(k\|ice)}{P(k\|steam)}$	high	small	≈ 1	(...)	≈ 1

- Problem? → Dimensionality!
- Question: “How to use the power of word-word co-occurrence without a direct computation?”
- $F(w_i, w_j, w_k) = \frac{P(w_k|w_i)}{P(w_k|w_j)}$ → Challenges: Vector vs. Scalar, number of arguments, F?

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe)

- Vector vs. Scalar: $\rightarrow F((w_i - w_j)^T \cdot w_k)$, computing the vector difference between $w_i - w_j$ (Vector Analogy – see before), followed by the dot-product, together with the context word w_k, leading to a scalar value

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe)

- Vector vs. Scalar: $\rightarrow F((w_i - w_j)^T \cdot w_k)$, computing the vector difference between $w_i - w_j$ (Vector Analogy – see before), followed by the dot-product, together with the context word w_k, leading to a scalar value
- Function F (“Homomorphism“): $F(w_i^T \cdot w_k - w_j^T \cdot w_k) = \frac{F(w_i^T \cdot w_k)}{F(w_j^T \cdot w_k)} = \frac{P(w_k|w_i)}{P(w_k|w_j)}$

  $\rightarrow F(w_i^T \cdot w_k) = c \cdot P(w_k|w_i)$ (ignore constant c) $\rightarrow F(x) = e^x$

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe)

- Vector vs. Scalar: $\rightarrow F((w_i - w_j)^T \cdot w_k)$, computing the vector difference between $w_i - w_j$ (Vector Analogy – see before), followed by the dot-product, together with the context word w_k, leading to a scalar value
- Function F (“Homomorphism“): $F(w_i^T \cdot w_k - w_j^T \cdot w_k) = \frac{F(w_i^T \cdot w_k)}{F(w_j^T \cdot w_k)} = \frac{P(w_k|w_i)}{P(w_k|w_j)}$

  $\rightarrow F(w_i^T \cdot w_k) = c \cdot P(w_k|w_i)$ (ignore constant c) $\rightarrow F(x) = e^x$
- $P(w_k|w_i) = e^{w_i^T \cdot w_k} \rightarrow w_i^T \cdot w_k = ln\, P(w_k|w_i) = ln\,(X_{ik}) - ln\,(X_i)$

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe)

- Vector vs. Scalar: $\rightarrow F((w_i - w_j)^T \cdot w_k)$, computing the vector difference between $w_i - w_j$ (Vector Analogy – see before), followed by the dot-product, together with the context word w_k, leading to a scalar value
- Function F (“Homomorphism“): $F(w_i^T \cdot w_k - w_j^T \cdot w_k) = \frac{F(w_i^T \cdot w_k)}{F(w_j^T \cdot w_k)} = \frac{P(w_k|w_i)}{P(w_k|w_j)}$

  $\rightarrow F(w_i^T \cdot w_k) = c \cdot P(w_k|w_i)$ (ignore constant c) $\rightarrow F(x) = e^x$
- $P(w_k|w_i) = e^{w_i^T \cdot w_k} \rightarrow w_i^T \cdot w_k = ln\, P(w_k|w_i) = ln\,(X_{ik}) - ln\,(X_i)$
- $P(w_k|w_i) = e^{w_i^T \cdot w_k} \rightarrow w_i^T \cdot w_k = ln\, P(w_k|w_i) = ln\,(X_{ik}) - ln\,(X_i)$

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe)

- Vector vs. Scalar: $\rightarrow F((w_i - w_j)^T \cdot w_k)$, computing the vector difference between $w_i - w_j$ (Vector Analogy – see before), followed by the dot-product, together with the context word w_k, leading to a scalar value
- Function F (“Homomorphism“): $F(w_i^T \cdot w_k - w_j^T \cdot w_k) = \frac{F(w_i^T \cdot w_k)}{F(w_j^T \cdot w_k)} = \frac{P(w_k|w_i)}{P(w_k|w_j)}$

  $\rightarrow F(w_i^T \cdot w_k) = c \cdot P(w_k|w_i)$ (ignore constant c) $\rightarrow F(x) = e^x$
- $P(w_k|w_i) = e^{w_i^T \cdot w_k} \rightarrow w_i^T \cdot w_k = ln\, P(w_k|w_i) = ln\,(X_{ik}) - ln\,(X_i)$
- $P(w_k|w_i) = e^{w_i^T \cdot w_k} \rightarrow w_i^T \cdot w_k = ln\, P(w_k|w_i) = ln\,(X_{ik}) - ln\,(X_i)$
- Reduce number arguments $(w_i, w_j, w_k$ vs. $w_i, w_k)$: $w_i^T \cdot w_k + \underbrace{b_i + b_k}_{ln(X_i)} = ln\,(X_{ik})$

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe)

- Vector vs. Scalar: $\rightarrow F((w_i - w_j)^T \cdot w_k)$, computing the vector difference between $w_i - w_j$ (Vector Analogy – see before), followed by the dot-product, together with the context word w_k, leading to a scalar value
- Function F (“Homomorphism“): $F(w_i^T \cdot w_k - w_j^T \cdot w_k) = \frac{F(w_i^T \cdot w_k)}{F(w_j^T \cdot w_k)} = \frac{P(w_k|w_i)}{P(w_k|w_j)}$

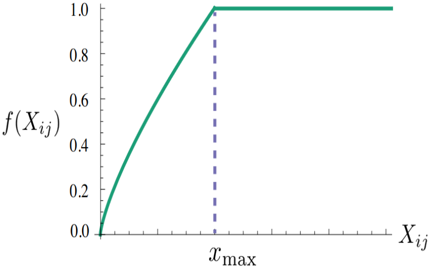
  $\rightarrow F(w_i^T \cdot w_k) = c \cdot P(w_k|w_i)$ (ignore constant c) $\rightarrow F(x) = e^x$
- $P(w_k|w_i) = e^{w_i^T \cdot w_k} \rightarrow w_i^T \cdot w_k = ln\,P(w_k|w_i) = ln\,(X_{ik}) - ln\,(X_i)$
- $P(w_k|w_i) = e^{w_i^T \cdot w_k} \rightarrow w_i^T \cdot w_k = ln\,P(w_k|w_i) = ln\,(X_{ik}) - ln\,(X_i)$
- Reduce number arguments (w_i, w_j, w_k vs. w_i, w_k): $w_i^T \cdot w_k + \underbrace{b_i + b_k}_{ln(X_i)} = ln\,(X_{ik})$
- Loss Function: $J(\theta) = \sum_{i,j=1}^{|V|} f(X_{ij}) \left[w_i^T \cdot w_j + b_i + b_j - ln\,(X_{ij}) \right]^2$, with context $j = k$

Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe) – Cost Function $J(\theta)$

- $J(\theta) = \sum_{i,j=1}^{|V|} f(X_{ij}) \underbrace{\left[w_i^T \cdot w_j + b_i + b_j - ln\left(X_{ij}\right)\right]^2}_{(y-\hat{y})^2 = Do\ you\ know\ the\ loss?}$
- Weighting function $f(X_{ij})$, using the word-word co-occurrence counts X_{ij}
- $f(x) = \begin{cases} \left(\frac{x}{x_{max}(=100)}\right)^{\alpha=\frac{3}{4}}, & \text{if } x \geq x_{max} \\ 1, & \text{otherwise} \end{cases}$

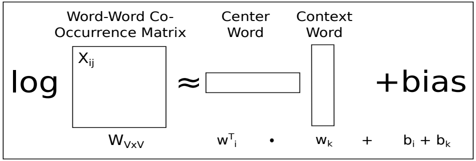


Source: Image taken from Pennington et al., *“GloVe: Global vectors for word representation”*, 2014, Figure 1

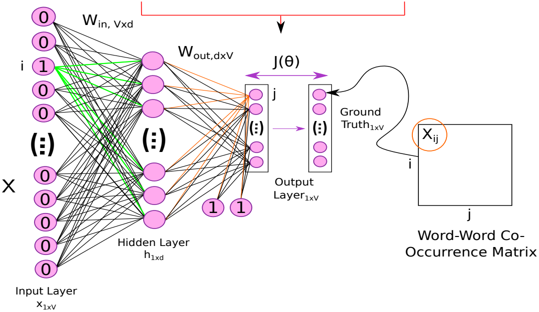
Distributional Similarities (Vector-Space Semantics)

Dense Vector – (Static) Word Embeddings – GloVe

Global Vectors (GloVe) – Model Architecture



- $$J(\theta) = \sum_{i,j=1}^{|V|} f(X_{ij}) \underbrace{\left[w_i^T \cdot w_j + b_i + b_j - ln\left(X_{ij}\right)\right]^2}_{(y-\hat{y})^2 = L2-Loss}$$
- Learning word embeddings $W_{in,V \times d}$ by approximating the word-word co-occurrence matrix (global information)



Distributional Similarities (Vector-Space Semantics)

Dense Vector – Contextualized Word Embeddings

Outlook: Contextualized Word Embeddings (usually derived by Transformer-based Models)

- McCann et al., *Learned in translation: Contextualized word vectors*, 2017

Distributional Similarities (Vector-Space Semantics)

Dense Vector – Contextualized Word Embeddings

Outlook: Contextualized Word Embeddings (usually derived by Transformer-based Models)

- McCann et al., *Learned in translation: Contextualized word vectors*, 2017
- Devlin et al., *"BERT: Pre-training of deep bidirectional transformers for language understanding"*, 2018

Distributional Similarities (Vector-Space Semantics)

Dense Vector – Contextualized Word Embeddings

Outlook: Contextualized Word Embeddings (usually derived by Transformer-based Models)

- McCann et al., *Learned in translation: Contextualized word vectors*, 2017
- Devlin et al., "*BERT: Pre-training of deep bidirectional transformers for language understanding*", 2018
- Peters et al., "*Deep contextualized word representations*", 2018

Distributional Similarities (Vector-Space Semantics)

Dense Vector – Contextualized Word Embeddings

Outlook: Contextualized Word Embeddings (usually derived by Transformer-based Models)

- McCann et al., *Learned in translation: Contextualized word vectors*, 2017
- Devlin et al., *“BERT: Pre-training of deep bidirectional transformers for language understanding”*, 2018
- Peters et al., *“Deep contextualized word representations”*, 2018
- Recap: static word embeddings (Word2Vec, GloVe) ignore that a particular word may comprise multiple meanings/senses → Just single vector in the semantic space

Distributional Similarities (Vector-Space Semantics)

Dense Vector – Contextualized Word Embeddings

Outlook: Contextualized Word Embeddings (usually derived by Transformer-based Models)

- McCann et al., *Learned in translation: Contextualized word vectors*, 2017
- Devlin et al., *"BERT: Pre-training of deep bidirectional transformers for language understanding"*, 2018
- Peters et al., *"Deep contextualized word representations"*, 2018
- Recap: static word embeddings (Word2Vec, GloVe) ignore that a particular word may comprise multiple meanings/senses → Just single vector in the semantic space
  → Polysemy and Homonymy are not covered properly!

Distributional Similarities (Vector-Space Semantics)

Dense Vector – Contextualized Word Embeddings

Outlook: Contextualized Word Embeddings (usually derived by Transformer-based Models)

- McCann et al., *Learned in translation: Contextualized word vectors*, 2017
- Devlin et al., *"BERT: Pre-training of deep bidirectional transformers for language understanding"*, 2018
- Peters et al., *"Deep contextualized word representations"*, 2018
- Recap: static word embeddings (Word2Vec, GloVe) ignore that a particular word may comprise multiple meanings/senses → Just single vector in the semantic space

  → Polysemy and Homonymy are not covered properly!
- Contextualized Word Embeddings (not static!) – Key idea:
  - Build and compute a vector for each word conditioned on its context: $f : (w_1, w_2, \dots, w_n) \rightarrow x_1, \dots, x_N \in \mathbb{R}^N$ → Sequence-level semantics!

Hands on Vector Semantics and Word Embeddings in Python...

Source: https://www.activestate.com/blog/top-10-coding-mistakes-in-python-how-to-avoid-them/

Further Questions?

https://www.oth-aw.de/hochschule/ueber-uns/personen/bergler-christian/
https://www.oth-aw.de/hochschule/ueber-uns/personen/levi-patrick/

References

Used Literature...

- Daniel Jurafsky, James H. Martin, Speech and Language Processing, Copyright © 2023. All rights reserved. Draft of January 7, 2023
- Mikolov et al., Efficient Estimation of Word Representations in Vector Space, 2013
- Mikolov et al., Distributed Representations of Words and Phrases and their Compositionality, 2013
- https://web.stanford.edu/~jurafsky/slp3/ed3book_jan72023.pdf
- https://web.stanford.edu/~jurafsky/slp3/6.pdf
- https://web.stanford.edu/~jurafsky/slp3/slides/6_Vector_Apr18_2021.pdf
- https://courses.grainger.illinois.edu/cs447/fa2020/Slides/Lecture06.pdf