Distributional and Distributed Semantics

Chapter 14

Felix Dietrich 2020/07/10

Motivation

- Mapping words to meaning
  - One word multiple meanings e.g. *bank*
  - Multiple surface forms with same meaning: **synonymy**

- Previous two chapters:
  - Hand-crafted mappings from words to semantic predicates
  - Labeled data required (does not scale well)

- **Problem:** How to deal with unseen words?
- **Approach:** Try to learn representations of word meanings by analyzing unlabeled data

➡ **Distributional hypothesis**

The Distributional Hypothesis

- *"You shall know a word by the company it keeps"*
  —Firth (1957)
- Learn the meaning of a word from context
  - On large amounts of unlabeled data to learn also about rare words

- What does ***tezgüino*** mean?
  - *A bottle of* ***tezgüino*** *is on the table.*
  - *Everybody likes* ***tezgüino****.*
  - *Don't have* ***tezgüino*** *before you drive.*
  - *We make* ***tezgüino*** *out of corn.*

- We can infer a lot about the meaning of ***tezgüino*** out of the context it appears in

The Distributional Hypothesis (cont.)

(14.1) A bottle of _____ is on the table.

(14.2) Everybody likes _____.

(14.3) Don't have _____ before you drive.

(14.4) We make _____ out of corn.

- What words fits into these contexts?

	(14.1)	(14.2)	(14.3)	(14.4)	...
tezgüino	1	1	1	1	
loud	0	0	0	0	
motor oil	1	0	0	1	
tortillas	0	1	0	1	
choices	0	1	0	0	
wine	1	1	1	0	

completely different

very similar

These vectors are called **word representations**

Distributional Properties of Words

- Distributional statistics capture lexical semantic relationships such as analogies:

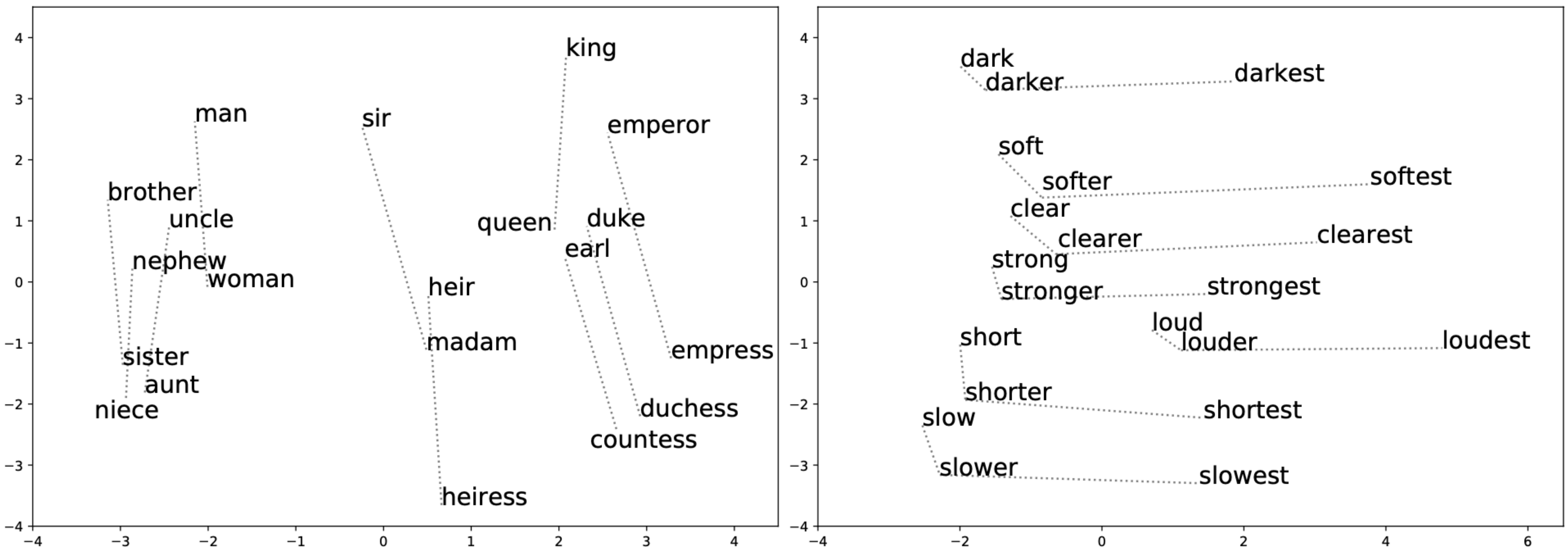


Figure 14.1: Lexical semantic relationships have regular linear structures in two dimensional projections of distributional statistics (Pennington et al., 2014).

Design Decisions for Word Representations

- Three main dimensions of decisions to consider
- **Representation:**
  - Nature of the representation
- **Context:**
  - Source of the contextual information
- **Estimation:**
  - Estimation procedure

Representation

- Today, mostly: **word embeddings**
  - k-dimensional real valued vectors
  - Continuous representation
  - Well suited for neural networks, linear classifiers, and structured prediction models
- Popular alternative: **Brown clusters**
  - Words are represented by variable-length bit strings
  - Discrete representation
  - Good for perceptron and conditional random fields

- **Question:** One embedding per surface form or multiple?
  - **Intuitively:** multiple meanings should have multiple embeddings ⇒ Use unsupervised clustering
  - **Arguably:** not necessary (surface form embedding is a linear combination of the underlying senses)

Context

- Context of *tezgüino* example: the entire sentence
  - Not practical (too many sentences exists)

- Possible alternative smaller contexts:

*The moment one **learns** English, complications set in* (Alfau, 1999)	
Brown Clusters	$\{one\}$
WORD2VEC, $h = 2$	$\{moment, one, English, complications\}$
Structured WORD2VEC, $h = 2$	$\{(moment, -2), (one, -1), (English, +1), (complications, +2)\}$
Dependency contexts,	$\{(one, \text{NSUBJ}), (English, \text{DOBJ}), (moment, \text{ACL}^{-1})\}$

- Much larger contexts possible:
  - In **latent semantic analysis**: a whole document
  - In **explicit semantic analysis**: a Wikipedia page

Context (cont.)

- Applying latent semantic analysis with context size h for the word *dog* (nearest-neighbors):
  - *(h=2): cat, horse, fox, pet, rabbit, pig, animal, mongrel, sheep, pigeon*
  - *(h=30): kennel, puppy, pet, bitch, terrier, rottweiler, canine, cat, to bark, Alsatian*

- Which one is better?
  - *(h=2):* More sensitive to syntax
  - *(h=30):* More sensitive to topic

- Choice of context has a profound effect on the resulting representations

Estimation

- Estimate word embeddings by optimizing some objective
- **Maximum likelihood estimation**
  - Objective: $log\, p(\boldsymbol{w}; \boldsymbol{U})$
  - $\boldsymbol{U} \in \mathbb{R}^{K \times V}$ matrix of embeddings
  - $\boldsymbol{w} = \{w_m\}_{m=1}^{M}$ the corpus with M tokens
  - RNNs work directly
    - Backpropagate to the input embeddings
    - But difficult to scale to large data
  - Usually simplified likelihoods or heuristics are used

Estimation (cont.)

- **Matrix factorization**
  - $\boldsymbol{C} = \{\text{count}(i, j)\}$ co-occurence counts of word i in context j

  - Minimize: $\min_{\boldsymbol{u},\boldsymbol{v}} ||\mathbf{C} - \tilde{\mathbf{C}}(\boldsymbol{u}, \boldsymbol{v})||_F$

    - $\tilde{\mathbf{C}}(\boldsymbol{u}, \boldsymbol{v})$: approximate reconstruction from embeddings $\boldsymbol{u}$ and $\boldsymbol{v}$
    - $||\mathbf{X}||_F$: Forbenius norm $\sum_{i,j} x_{i,j}^2$

    - Counts are often transformed by information-theoretic metrics s.a. **pointwise mutual information** (PMI)

Latent Semantic Analysis

- Get vector representation using **truncated singular value decomposition** (SVD):

$$\min_{\mathbf{U}\in\mathbb{R}^{V\times K},\mathbf{S}\in\mathbb{R}^{K\times K},\mathbf{V}\in\mathbb{R}^{|\mathcal{C}|\times K}} \quad ||\mathbf{C}-\mathbf{U}\mathbf{S}\mathbf{V}^\top||_F$$

(approximation error)

$$\textbf{s.t.} \quad \mathbf{U}^\top\mathbf{U}=\mathbb{I}$$

$$\mathbf{V}^\top\mathbf{V}=\mathbb{I}$$

(uncorrelated dimensions)

$$\forall i \neq j, \mathbf{S}_{i,j}=0,$$

(diagonal matrix)

  - V : size of Vocabulary
  - $|\mathcal{C}|$: Number of contexts
  - K : resulting embedding size
  - Element $c_{i,j}$ is reconstructed as a **bilinear product**:

$$c_{i,j} \approx \sum_{k=1}^{K} u_{i,k} s_k v_{j,k}$$

Latent Semantic Analysis (cont.)

- It is most effective if the count matrix is transformed before applying SVD
- Example: **pointwise mutual information** (PMI)
  - Degree of association between word i and context j

$$\mathrm{PMI}(i,j) = \log \frac{\mathrm{p}(i,j)}{\mathrm{p}(i)\mathrm{p}(j)} = \log \frac{\mathrm{p}(i \mid j)\mathrm{p}(j)}{\mathrm{p}(i)\mathrm{p}(j)} = \log \frac{\mathrm{p}(i \mid j)}{\mathrm{p}(i)}$$

$$= \log \mathrm{count}(i,j) - \log \sum_{i'=1}^{V} \mathrm{count}(i',j)$$

$$- \log \sum_{j' \in \mathcal{C}} \mathrm{count}(i,j') + \log \sum_{i'=1}^{V} \sum_{j' \in \mathcal{C}} \mathrm{count}(i',j')$$

Latent Semantic Analysis (cont.)

- If word i is statistically associated with context j then $\mathrm{PMI}(i,j) > 0$

➡Focus on reconstructing strong word-context associations instead of large counts

- PMI is negative when word and context occur together less often than if they were independent
  - This is unreliable (counts of rare events have high variance)
- PMI is undefined for $\mathrm{count}(i,j) = 0$

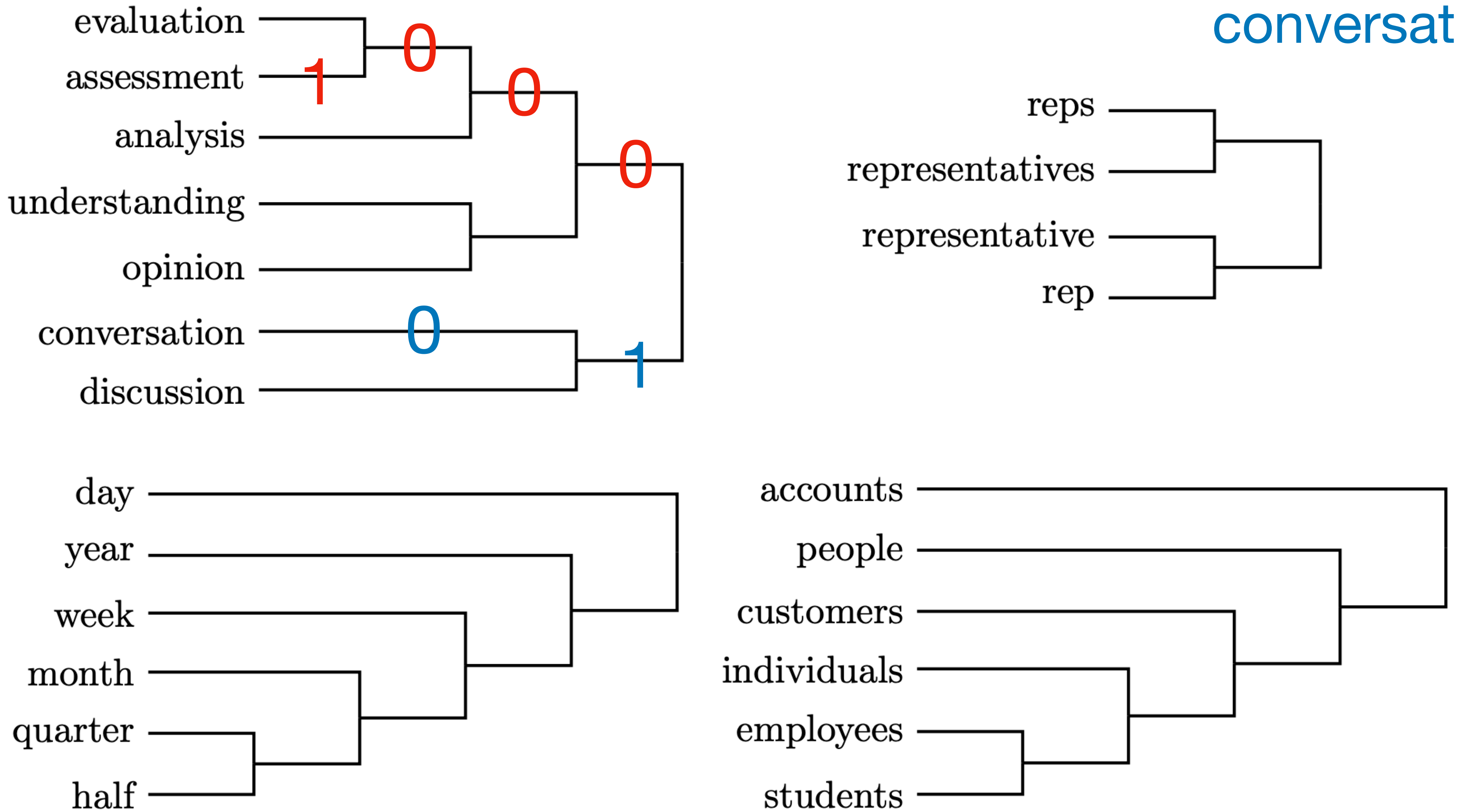
- Possible solution: **Positive PMI** (PPMI) works better

$$\mathrm{PPMI}(i,j) = \begin{cases} \mathrm{PMI}(i,j), & \mathrm{p}(i \mid j) > \mathrm{p}(i) \\ 0, & \text{otherwise.} \end{cases}$$

Brown Clusters

- Discrete feature vectors
  - Useful for perceptron and conditional random fields
- Cluster by similar distributional statistics
- **Hierarchical clustering:**

assessment 0001
conversation 10



Brown Clusters (cont.)

bitstring	ten most frequent words
01111010**0111**	*excited thankful grateful stoked pumped anxious hyped psyched exited geeked*
01111010**100**	*talking talkin complaining talkn bitching tlkn tlkin bragging raving +k*
01111010**1010**	*thinking thinkin dreaming worrying thinkn speakin reminiscing dreamin daydreaming fantasizing*
01111010**1011**	*saying sayin suggesting stating sayn jokin talmbout implying insisting 5′2*
01111010**1100**	*wonder dunno wondered duno donno dno dono wonda wounder dunnoe*
01111010**1101**	*wondering wonders debating deciding pondering unsure wonderin debatin woundering wondern*
01111010**1110**	*sure suree suuure suure sure- surre sures shuree*

This prefix groups by: communicating and knowing, especially in the present participle (Brown clustering on Twitter data)

Brown Clusters (cont.)

- Hierarchical trees can be induced from a likelihood-based objective, $k_i \in \{1, 2, \ldots, K\}$ to represent cluster of word i:

$$\log \mathrm{p}(\boldsymbol{w}; \boldsymbol{k}) \approx \sum_{m=1}^{M} \log \mathrm{p}(w_m \mid w_{m-1}; \boldsymbol{k})$$

$$\triangleq \sum_{m=1}^{M} \log \mathrm{p}(w_m \mid k_{w_m}) + \log \mathrm{p}(k_{w_m} \mid k_{w_{m-1}})$$

  - Different from hidden Markov model with

$\forall k \neq k_{w_m}, \mathrm{p}(w_m \mid k) = 0$ (a word can only be emitted by a single cluster)

Brown Clusters (cont.)

- Construct tree bottom up
    - Start with each word in its own cluster
    - Merge clusters incrementally until one remains such that the objective remains maximized at each step

- Optimal merges at each step maximize the **average mutual information:**

$$I(\boldsymbol{k}) = \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \mathrm{p}(k_1, k_2) \times \mathrm{PMI}(k_1, k_2)$$

$$\mathrm{p}(k_1, k_2) = \frac{\mathrm{count}(k_1, k_2)}{\sum_{k_{1'}=1}^{K} \sum_{k_{2'}=1}^{K} \mathrm{count}(k_{1'}, k_{2'})},$$

- $\mathrm{p}(k_1, k_2)$ joint probability of a bigram of word in cluster k_1 followed by word in cluster k_2

Neural Word Embeddings

- Continuous vector representation
- Likelihood-based objective
- Inner product of K-dimensional embeddings: $\boldsymbol{u}_i \cdot \boldsymbol{v}_j$
    - Represents compatibility between word i and context j
- Incorporate inner product into an approximation of the log-likelihood of a corpus
    - Backpropagate to the embeddings

- Two variants of Word2Vec:
    - **Continuous bag-of-words (CBOW)**
    - **Skipgrams**

Continuous Bag-of-Words (CBOW)

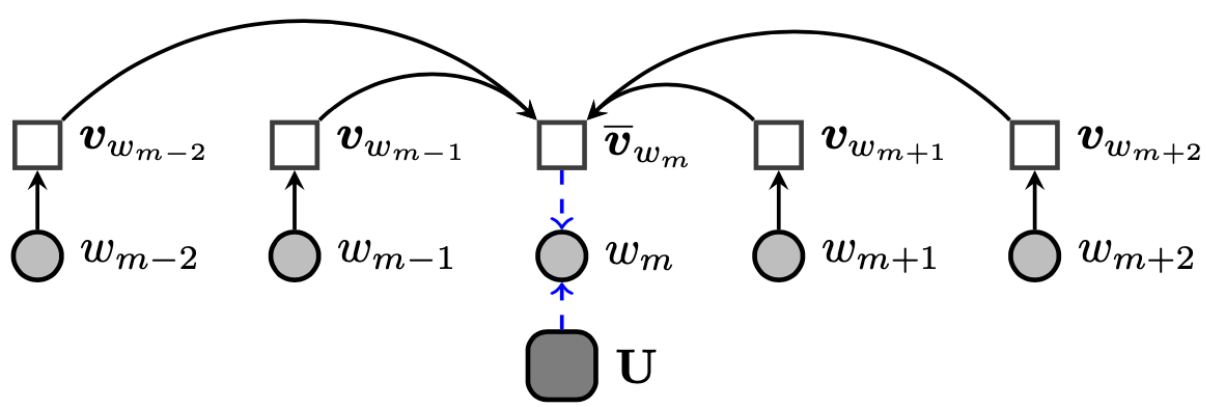
- Words are predicted from the their context

- Local context computed as an average of embeddings for words in the immediate neighborhood:

$$m-h, m-h+1, \ldots, m+h-1, m+h,$$

$$\overline{\boldsymbol{v}}_m = \frac{1}{2h}\sum_{n=1}^{h} \boldsymbol{v}_{w_{m+n}} + \boldsymbol{v}_{w_{m-n}}$$

  - Order of the context does not matter
  - h is the neighborhood size



Continuous Bag-of-Words (CBOW) (cont.)

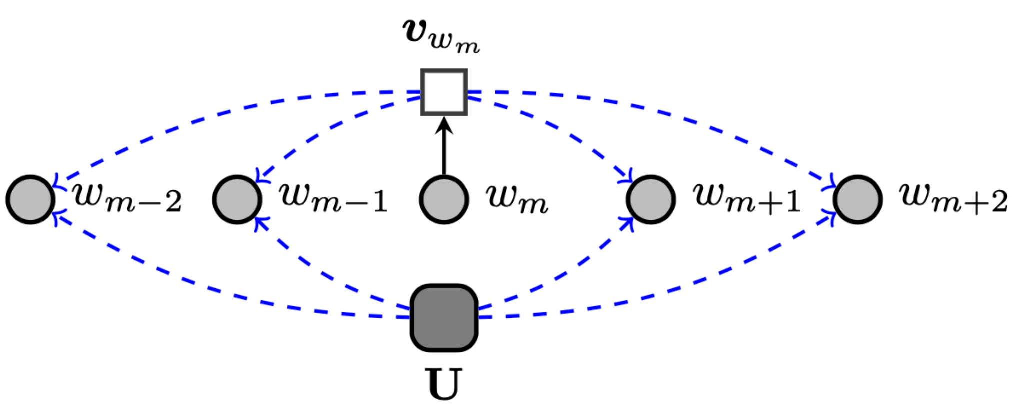
- CBOW optimizes:

$$
\begin{aligned}
\log \mathrm{p}(\boldsymbol{w}) &\approx \sum_{m=1}^{M} \log \mathrm{p}(w_m \mid w_{m-h}, w_{m-h+1}, \ldots, w_{m+h-1}, w_{m+h}) \\
&= \sum_{m=1}^{M} \log \frac{\exp\left(\boldsymbol{u}_{w_m} \cdot \overline{\boldsymbol{v}}_m\right)}{\sum_{j=1}^{V} \exp\left(\boldsymbol{u}_j \cdot \overline{\boldsymbol{v}}_m\right)} \\
&= \sum_{m=1}^{M} \boldsymbol{u}_{w_m} \cdot \overline{\boldsymbol{v}}_m - \log \sum_{j=1}^{V} \exp\left(\boldsymbol{u}_j \cdot \overline{\boldsymbol{v}}_m\right).
\end{aligned}
$$

  - M is the size of the corpus

Skipgrams

- Context is predicted from the word (opposite to CBOW)



- Objective:

$$\begin{aligned}\log \mathrm{p}(\boldsymbol{w}) &\approx \sum_{m=1}^{M}\sum_{n=1}^{h_m} \log \mathrm{p}(w_{m-n} \mid w_m) + \log \mathrm{p}(w_{m+n} \mid w_m)\\ &= \sum_{m=1}^{M}\sum_{n=1}^{h_m} \log \frac{\exp(\boldsymbol{u}_{w_{m-n}} \cdot \boldsymbol{v}_{w_m})}{\sum_{j=1}^{V}\exp(\boldsymbol{u}_j \cdot \boldsymbol{v}_{w_m})} + \log \frac{\exp(\boldsymbol{u}_{w_{m+n}} \cdot \boldsymbol{v}_{w_m})}{\sum_{j=1}^{V}\exp(\boldsymbol{u}_j \cdot \boldsymbol{v}_{w_m})}\\ &= \sum_{m=1}^{M}\sum_{n=1}^{h_m} \boldsymbol{u}_{w_{m-n}} \cdot \boldsymbol{v}_{w_m} + \boldsymbol{u}_{w_{m+n}} \cdot \boldsymbol{v}_{w_m} - 2\log\sum_{j=1}^{V}\exp\left(\boldsymbol{u}_j \cdot \boldsymbol{v}_{w_m}\right)\end{aligned}$$

  - Local neighborhood size h_m is uniformly sampled over the range $\{1, 2, \ldots, h_{\max}\}$

➡Nearer neighbors are weighted more

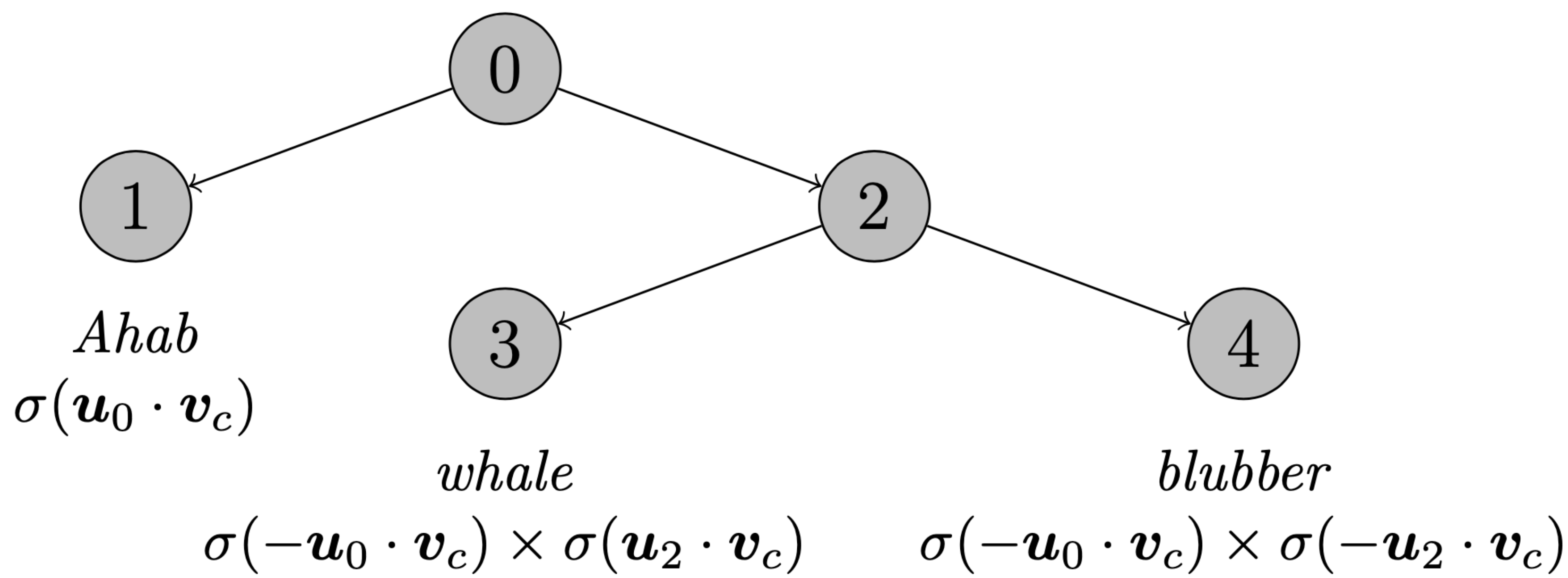
Computational Complexity

- Word2Vec as efficient alternative to RNN language models
  - Recurrent state update: quadratic in recurrent state vector size
  - CBOW and skipgram: linear complexity in word and context representations

- Normalization of probability required
  - **Naive implementation:**
    - Sum over the entire vocabulary
    - Complexity $\mathcal{O}(V \times K)$
  - **Hierarchical softmax:**
    - Tree-based computation
    - Logarithmic in the size of the vocabulary
  - **Negative sampling:**
    - Approximation removing the dependency on the size of the vocabulary

Hierarchical Softmax

- Normalized probability is reparameterized as sum over all paths in a binary tree (Brown clustering):



$$\sigma(x) = \frac{1}{1+\exp(-x)}$$

$$\begin{aligned}\Pr(\text{left at } n \mid c) =& \sigma(\boldsymbol{u}_n \cdot \boldsymbol{v}_c) \\ \Pr(\text{right at } n \mid c) =& 1 - \sigma(\boldsymbol{u}_n \cdot \boldsymbol{v}_c) = \sigma(-\boldsymbol{u}_n \cdot \boldsymbol{v}_c)\end{aligned}$$

- $\boldsymbol{u}_n$ output embedding for node n
- In balanced binary tree $\mathcal{O}(\log V)$

Negative Sampling

- Use alternative objective, maximize: $\sum_{m=1}^{M} \psi(w_m, c_m)$

$$\psi(i, j) = \log \sigma(\boldsymbol{u}_i \cdot \boldsymbol{v}_j) + \sum_{i' \in \mathcal{W}_{\text{neg}}} \log(1 - \sigma(\boldsymbol{u}_{i'} \cdot \boldsymbol{v}_j))$$

  - $\psi(i, j)$: score for word i in context j
  - $\mathcal{W}_{\text{neg}}$: set of negative samples (by sampling from a unigram language model)
    - Mikolov et. Al. (2013) use $\hat{p}(i) \propto (\textbf{count}(i))^{\frac{3}{4}}$
    - Redistributes probability mass from common to rare words
    - Mikolov: 5-20 samples for small training sets, 2-5 for larger corpora

Word Embeddings as Matrix Factorization

- For a matrix with all word-context counts non-zero, negative sampling is equivalent to factorization of matrix:

$$M_{ij} = \mathrm{PMI}(i, j) - \log k$$

  - k number of negative samples
  - Is $-\infty$ for not observed data

- **Shifted positive point wise mutual information:**

$$M_{ij} = \max(0, \mathrm{PMI}(i, j) - \log k)$$

  - Obtain word embeddings from this matrix with truncated SVD

GloVe ("global vectors")

- Factor matrix $M_{ij} = \log \mathrm{count}(i, j)$
- Estimate word embeddings with:

$$\min_{\boldsymbol{u},\boldsymbol{v},b,\tilde{b}} \quad \sum_{j=1}^{V} \sum_{j \in \mathcal{C}} f(M_{ij}) \left(\widehat{\log M_{ij}} - \log M_{ij} \right)^2$$

$$\textbf{s.t.} \quad \widehat{\log M_{ij}} = \boldsymbol{u}_i \cdot \boldsymbol{v}_j + b_i + \tilde{b}_j,$$

  - b_i and $\tilde{b}_j$ are offsets for word i and context j
  - Embeddings $\boldsymbol{u}$ and $\boldsymbol{v}$
  - Weighting function $f(M_{ij})$
    - Zero at $M_{ij} = 0$ (to avoid log of zero counts)
    - Saturates at $M_{ij} = m_{\max}$ (to avoid over-counting)
- Complexity scales with number of non-zero word-context counts
  - For English roughly $\mathcal{O}(N^{0.8})$ (Word2Vec is linear)

Evaluating Word Embeddings

- Two main ways

- **Intrinsic** evaluation:
  - How good are the embeddings in general?

- **Extrinsic** evaluation:
  - How good are the embeddings for a specific downstream task?

Intrinsic Evaluations

- Is similarity of word i and j reflected in the embeddings $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$?
- **Cosine similarity** (others possible):

$$\cos(\boldsymbol{u}_i, \boldsymbol{u}_j) = \frac{\boldsymbol{u}_i \cdot \boldsymbol{u}_j}{||\boldsymbol{u}_i||_2 \times ||\boldsymbol{u}_j||_2}$$

- Human judgement evaluation:
  - WS-353 dataset

word 1	word 2	similarity
love	*sex*	6.77
stock	*jaguar*	0.92
money	*cash*	9.15
development	*issue*	3.97
lad	*brother*	4.46

- Word analogies evaluation:
  - *king:queen :: man:woman*
  - $i_1 : j_1 :: i_2 :?$ (Most similar embedding to $\boldsymbol{u}_{i_1} - \boldsymbol{u}_{j_1} + \boldsymbol{u}_{i_2}$)
- **Supersense** similarity:
  - Broad lexical semantic categories (annotated in synsets)

Extrinsic Evaluations

- Word representations’ contribution to the downstream task
- Form of **semi-supervised learning**
- **Pre-trained word representations** can be used as features

- Evaluate performance of the downstream task that consumes them
  - GloVe convincingly better then Latent Semantic Analysis for named entity recognition
  - Extrinsic and intrinsic evaluations may conflict

- **Fine-tuning** of pre-trained embeddings possible
  - Or use both in conjunction

- **ELMo (embeddings from language models)**
  - Use deep BiLSTM for a contextualized representation
  - Yields often significant gains

Fairness and Bias

- *king:queen :: man:woman* gender-specific
- Other professions may be biased towards a gender
  - *homemaker, nurse, receptionist* (female bias)
  - *maestro, skipper, protege* (male bias)

- Word embeddings encode stereotypes
  - Gender, ethnic, …
  - Historical drift can be analyzed

- Biases often propagate or get amplified
  - Systems can fail to resolve pronouns

- Active research in “debiasing” machine learning

Distributed Representations Beyond Distributional Statistics

- Distributional word representations
  - Estimated from huge unlabeled data
    - For GloVe over 800 billion tokens of web data
  - **Problems:**
    - New words in the future
    - Unreliable embeddings for very rare words

➡Leverage other sources of information

Word-Internal Structure

- Words can be **composed** from sub-word units and are no longer atomic
- Examples:
  - **millicuries:** (unit of radioactivity)
    - Has **morphological** structure
    - *milli-* indicates amount, *-s* indicates plural
  - **caesium:** (chemical element)
    - Has a single morpheme
    - *-ium* often associated with chemical elements
  - **IAEA:** (International Atomic Energy Agency)
    - Acronym as suggested by the capitalization
    - *I-* often International, *-A* often Agency
  - **Zhezhgan:** (mining facility in Kazakhstan)
    - Title case suggests person or place
    - *zh* indicates transliteration

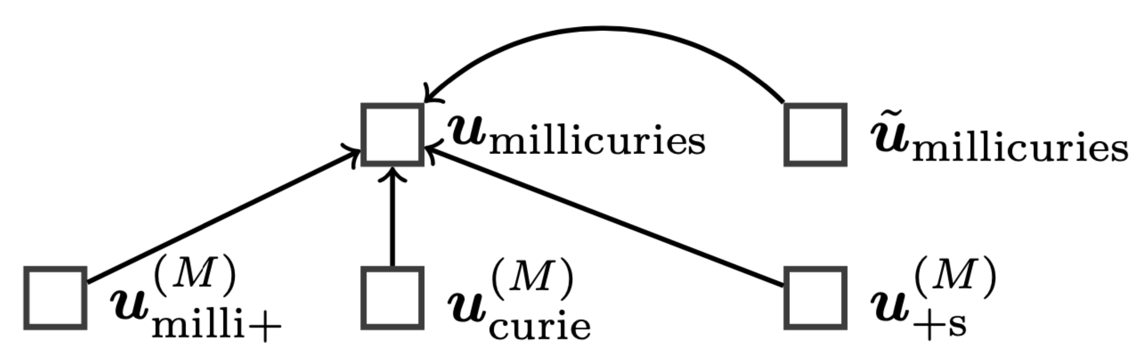
Word-Internal Structure (cont.)

- Split word i into morphological segments $\mathcal{M}_i$

$$\boldsymbol{u}_i = \tilde{\boldsymbol{u}}_i + \sum_{j \in \mathcal{M}_i} \boldsymbol{u}_j^{(M)}$$

  - $\boldsymbol{u}_m^{(M)}$ morpheme embedding
  - $\tilde{\boldsymbol{u}}_i$ non-compositional embedding of the whole word
  - Estimate from **log-bilinear language** model
    - Similar to CBOW
    - Includes only contextual information from preceding words
  - Use unsupervised morphological segmenter

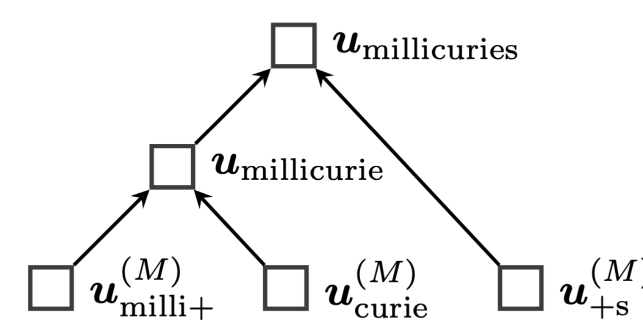
- Construct embedding of unseen words from their morphemes



Word-Internal Structure (cont.)

- **Subword units:**
  - *IAEA* and *Zhezhgan* don't follow morphological composition

  ➡Use characters, character n-grams, or **byte-pair encoding** (compression technique capturing frequent substrings)
- **Composition:**
  - Use a **recursive neural network** to differentiate between subword ordering
  - *((milli+curie)+s), ((in+flam)+able), (in+(vis+ible))*



- **Estimation:**
  - Estimate subword embeddings over pre-trained word embeddings
  - Reduces complexity to only the vocabulary size

Lexical Semantic Resources

- **Retrofit** pre-trained word embeddings across a network of lexical semantic relationships (e.g. WordNet):

$$\min_{\mathbf{U}} \quad \sum_{j=1}^{V} ||\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i||_2 + \sum_{(i,j)\in\mathcal{L}} \beta_{ij}||\boldsymbol{u}_i - \boldsymbol{u}_j||_2$$

  - $\hat{\boldsymbol{u}}_i$ pretrained embedding of word I
  - $\mathcal{L} = \{(i,j)\}$ is a lexicon of word relations
  - β_{ij} controls the importance of adjacent words having similar embeddings
  - Faruqui et al. (2015): $\beta_{ij} = |\{j : (i,j) \in \mathcal{L}\}|^{-1}$

  - Improves range of intrinsic evaluation performances
  - Small improvements on extrinsic document an classification task

Distributed Representations of Multiword Units

- What about the meaning of multiple words?
  - Phrases, sentences, paragraphs, …
- Can distributed representation be used?
  - No, larger spans of words usually don’t occur twice

➡Compute meaning of larger texts compositionally from smaller spans

Purely Distributional Methods

- Non-compositional multiword phrases
  - *San Francisco, kick the bucket, …*
  - Distributional approach can work
- **Collocation extraction:**
  - Problem of identifying multiword units
  - Collocation has high **pointwise mutual information**
    - Example: *Naïve Bayes*

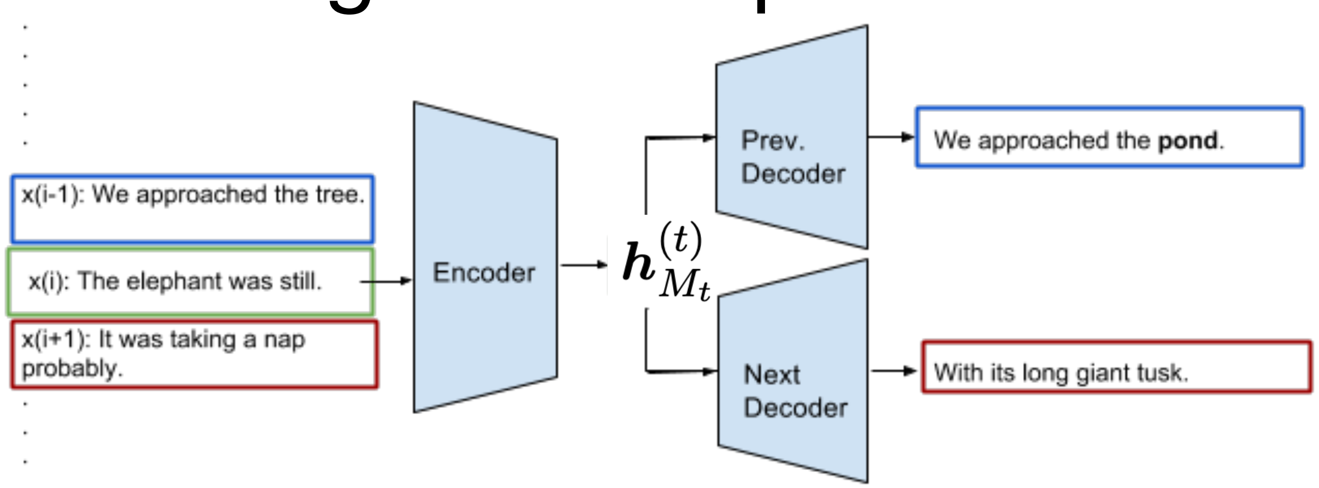
$$\mathrm{p}(w_t = \textit{Bayes} \mid w_{t-1} = \textit{naïve}) > \mathrm{p}(w_t = \textit{Bayes})$$

  - Identify longer sequences with a greedy incremental search
  - Treat a collocation as a single word

Distributional-Compositional Hybrids

- Beyond short multiword phrases, composition is necessary
- Represent meaning of a sentence by the average of its word embeddings (simple but effective)

- "Skip-thought" model:
  - Encode sentence t with RNN, use final hidden state $\boldsymbol{h}_{M_t}^{(t)}$
  - Decoder generates previous and next sentence



  - Hybrid of distributional and compositional approaches

Distributional-Compositional Hybrids (cont.)

- **Autoencoders:**
  - Encoder-decoder model trying to reconstruct the input
- **Denoising autoencoders:**
  - Corrupted version of the sentence tries to reconstruct the uncorrupted original sentence

- It is possible to interpolate between two sentences' distributional representation to combine their aspects:

$$\alpha \boldsymbol{u}_i + (1-\alpha)\boldsymbol{u}_j$$

this was the only way
it was the only way
it was her turn to blink
it was hard to tell
it was time to move on
he had to do it again
they all looked at each other
they all turned to look back
they both turned to face him
they both turned and walked away

Supervised Compositional Methods

- Given is a supervision signal to predict a label
    - Sentiment
    - Meaning of a sentence
    - …

- Simplest model: Average embeddings and input into a feedforward neural network
- Convolutional and RNNs capture multiword phenomena
- Recursive neural networks capture syntactic structures

- **Key question:** Is supervised sentence representation task-specific?
- **Stanford Natural Language Inference corpus**
    - Trained embeddings on this dataset transfer to a wider range of classification tasks

Hybrid Distributed-Symbolic Representations

- Distributed representations serve as summary of meaning
- Can be used to recognize the paraphrase relationship:
  - *a) Donald thanked Vlad profusely.*
  - *b) Donald conveyed to Vlad his profound appreciation.*
  - *c) Vlad was showered with gratitude by Donald.*

- Symbolic representations can reason about what happens between the entities *Vlad* and *Donald*
- Difficult for distributes representations

  ➡Hybrid between both

Hybrid Distributed-Symbolic Representations (cont.)

- “top-down” hybrid approach:
  - Begin with logical semantics
  - Replace the predefined lexicon with distributional representations
- “bottom-up” hybrid approach:
  - Add minimalistic symbolic structure to existing distributional representations
    - e.g. vector representations for each entity
- Improves performance for the problems:
  - **Discourse relations**
  - **Coreference resolution**

Questions?

Thank you for listening