D7041E Applied Artificial Intelligence
Natural Language Processing

Evgeny Osipov; Denis Kleyko; Gulnara Zhabelova; Niklas Karvonen

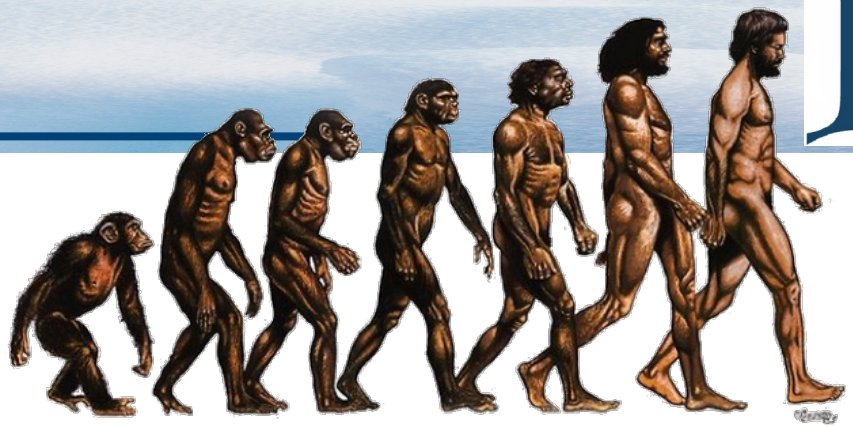
Dependable Communication and Computation Systems
Luleå Tekniska Universitet

Outline

- Introduction to NLP
- *N*-gram based models
- Common NLP tasks
- Word Embedding
- Methods of Word Embedding
  - Latent Semantic Analysis
  - Random Indexing
  - Recurrent Neural Networks
- Evaluation methods for Word Embedding
- Commercial usage of Word Embedding
- Laboratory description

Why NLP?

- Language is considered as the first huge outcome of the cognitive revolution in Homo

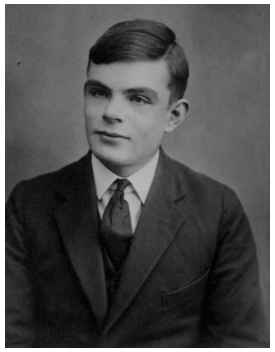


- It allows us to collaborate with other human beings

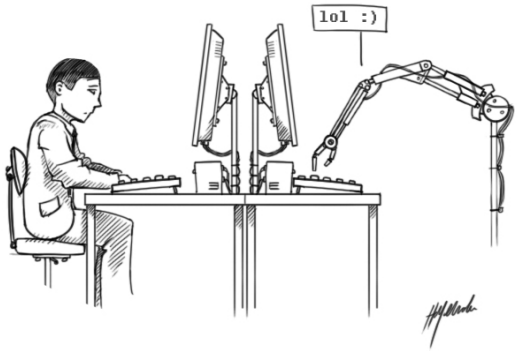
- We can convey information and store it in form of narratives

Why NLP?

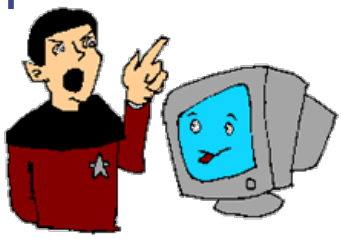
- Rich Language is a distinctive feature of a human being



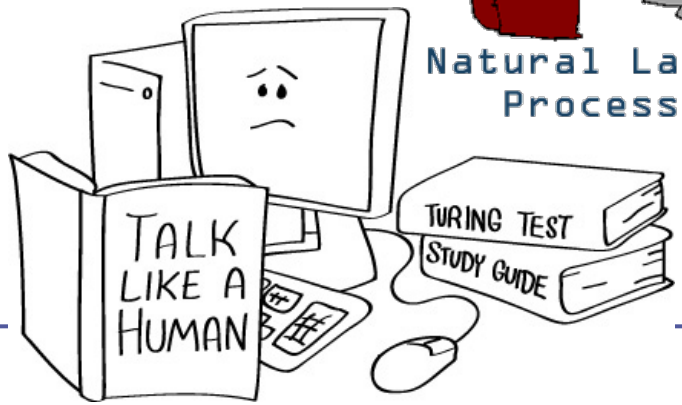
Famous Turing‘s test
is based on a natural language



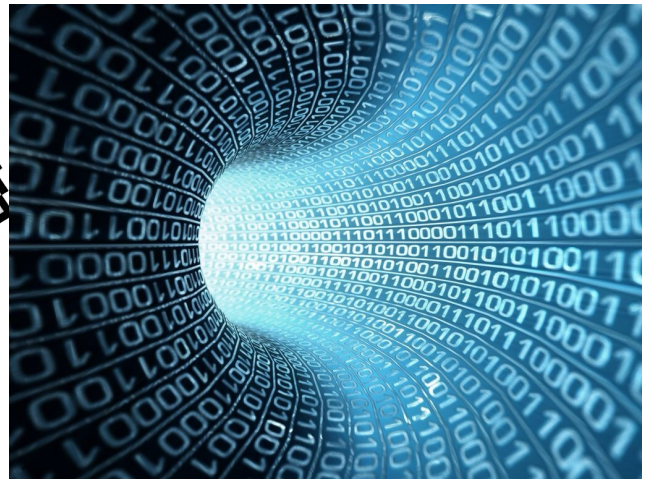
- Why do we want machines to be able to process natural languages?
  - To communicate with humans (not our topic today)



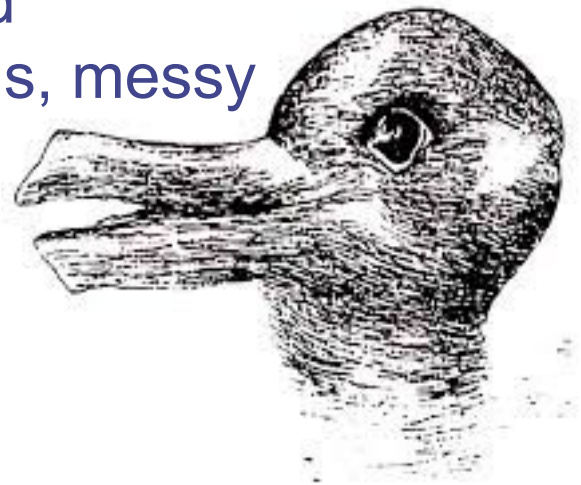
  - To acquire information from written language



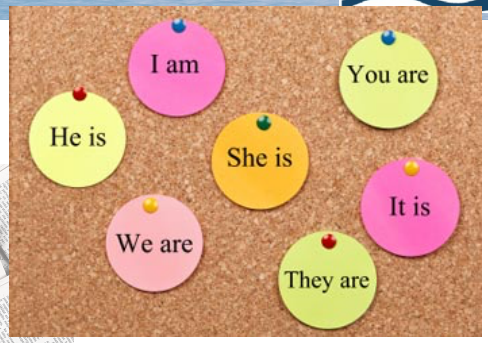
Some Facts



- Over a trillion pages of information on the Web
- Constantly increasing volume of data
- A machine needs to understand (at least partially) the ambiguous, messy languages that humans use.
- In order to do it machines should build language models

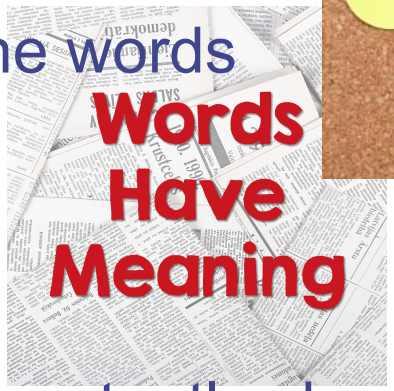


Language model



Models could reflect

- Grammar – set of rules to combine words
- Semantics – meaning of words and their combinations



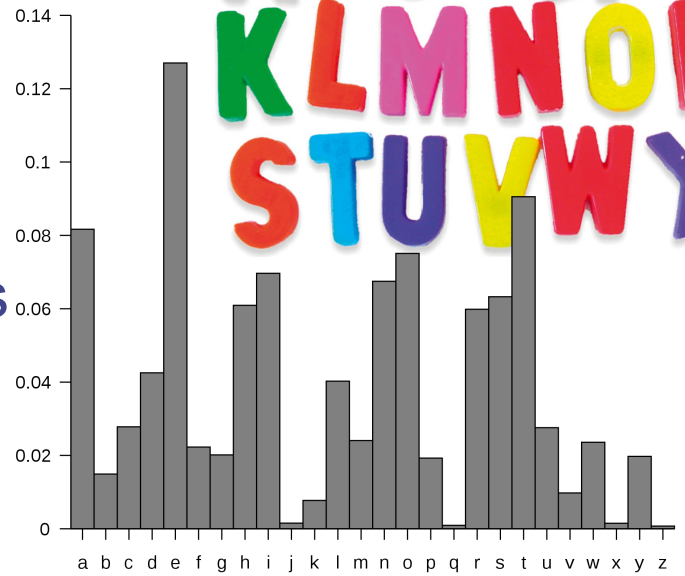
- Languages are ambiguous and constantly changing
  - A sentence could have several meanings

- In General, **a language model** predicts the probability distribution of language expressions
  - Models are, at best, an approximation

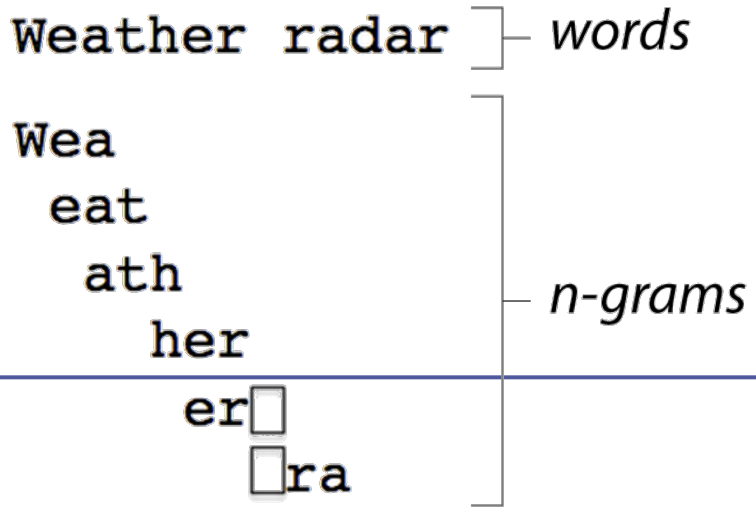
A simple language model: *N*-gram based

- A text is composed of characters

- Simple model: a probability distribution over sequences of characters

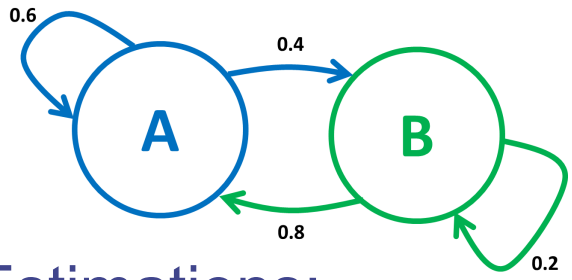


- A sequence of written symbols of length *n* is called an *n*-gram



A simple language model: *N*-gram based

- *N*-gram model: a model of the probability distribution of *n*-letter sequences. In fact it could be words, syllables, or other units
- An *n*-gram model is a Markov chain of order $n - 1$.
- *N*-gram model is approximated using a text corpus
- Smoothing improves the approximations. It is a process of adjusting the probability of low-frequency counts



Estimations:
0.58; 0.42
0.80; 0.20

BBABAAAABBABBABAABABAABBABAA
BABAAAABAABAAABAAAABABABAAAA
BAAABAABBBAAAABABABAAABAABAA

ABAAABABABAAABABABAAABABBBAA
ABABAAAABAAABAAAAAAAAAAABBAA
ABAABAAAAABBAAAABAAABABAAABA

Common NLP tasks

Text classification: given a text, decide which of a predefined set of classes it belongs

- Genre classification (*n*-grams are well suited): decide whether a text is a news story, a legal document, a scientific article, etc.

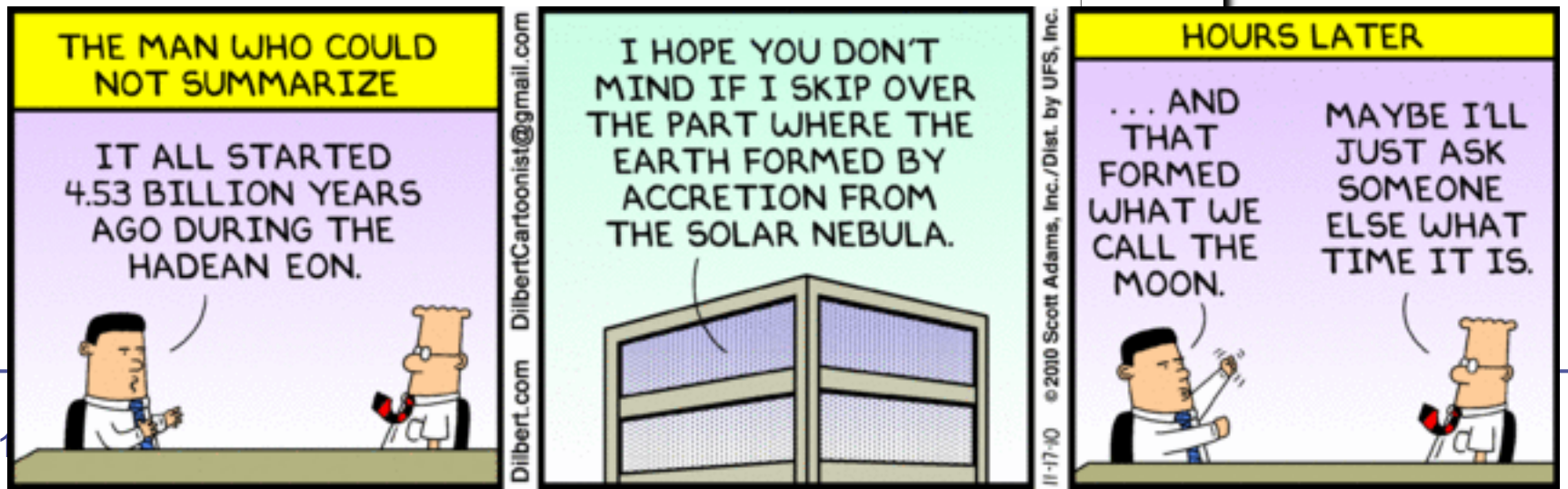
- Language identification (*n*-grams are well suited): given a text, determine what natural language it is written in

- Spam detection: classifying an email message as spam or not-spam

Common NLP tasks

- Spelling correction
  (*n*-grams are well suited)

- Automatic summarization:
  given a large chunk of text,
  produce a readable summary of it



Common NLP tasks

- Information retrieval: task of finding documents that are relevant to a user's need for information
  - Search engines on the Web

  - Scoring functions are important, e.g. PageRank

Common NLP tasks

- Question answering
  - Query really is a question
  - The answer is a short response – a sentence, or a phrase

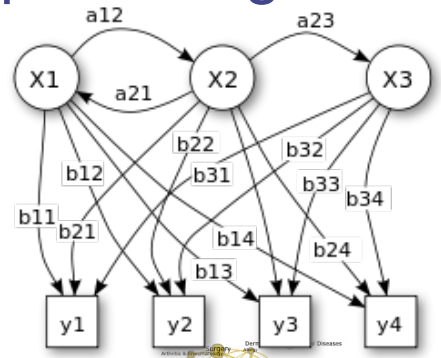
- Machine translation: one human language is automatically translated into another human language

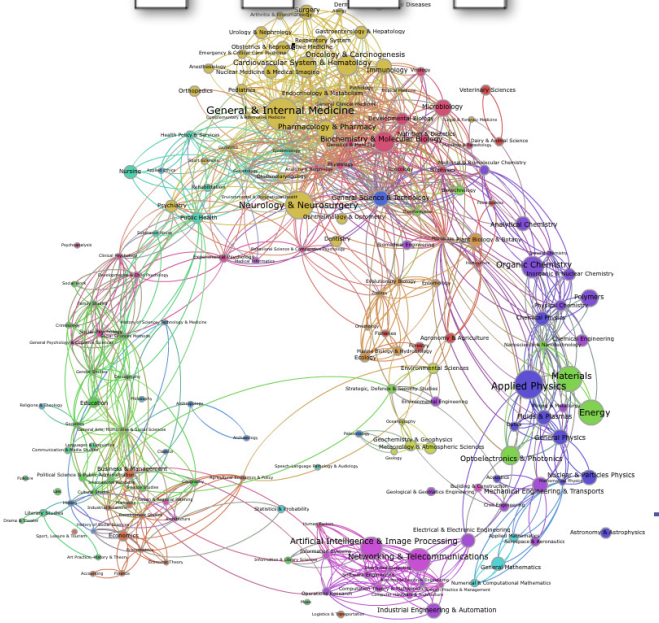
Common NLP tasks

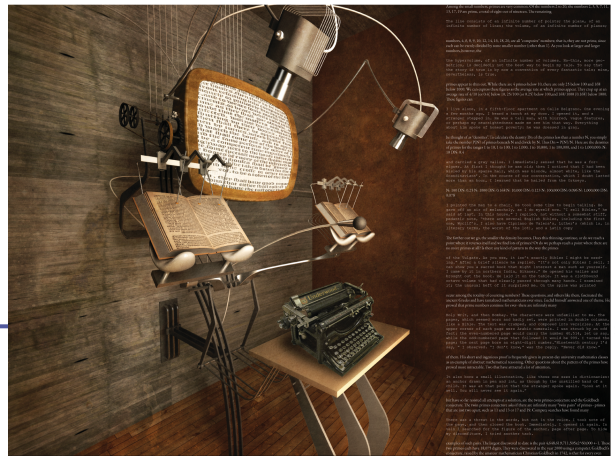
- Information extraction: is the process of acquiring knowledge by skimming a text and looking for occurrences of a particular class of object and for relationships among objects.
  - For example, extract instances of addresses from Web pages
  - Limited models that approximate the full English model
  - Attribute-based extraction – the simplest model

Common NLP tasks

- Information extraction: is the process of acquiring knowledge by skimming a text and looking for occurrences of a particular class of object and for relationships among objects.
  - Probabilistic models (hidden Markov model): extraction from noisy or varied input
  - Ontology extraction: building a large knowledge base of facts from a corpus
  - Machine reading







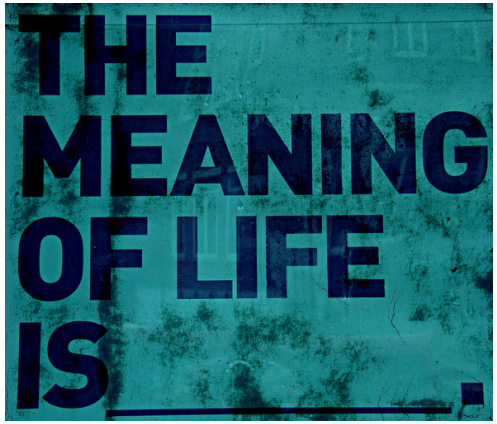
Meaning and Semantics

- Many NLP tasks require some understanding of meaning
  - Finding synonyms in Information retrieval such as “sofa” for “couch”
  - Machine reading
  - Machine translation

- Machines need meaning for communicating with humans

Meaning and Semantics

- Meaning is a fundamental component of nearly all aspects of human cognition



- A semantic memory is necessary for humans to construct meaning from otherwise meaningless words

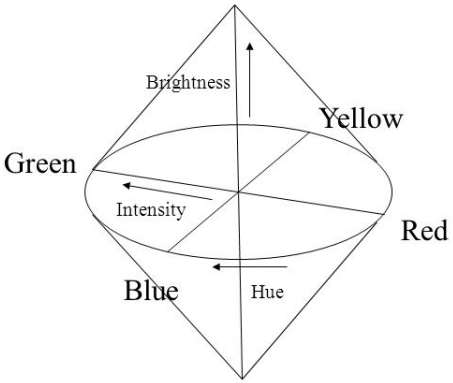
Meaning and Semantics

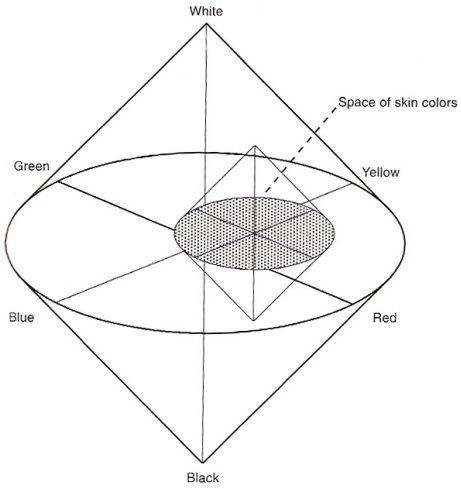
- korkskruv
- corkscrew
- shtopor
- dugohuzo

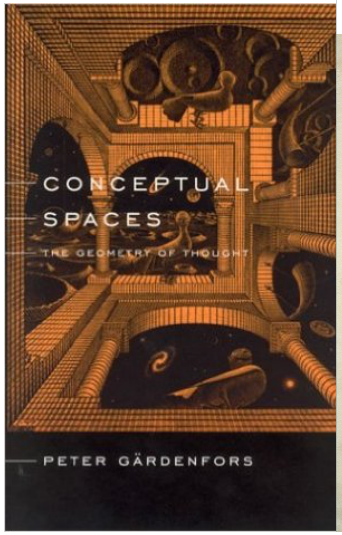
Meaning and Semantics

- Clear need for language models of semantics

The color domain







THE GEOMETRY OF MEANING
SEMANTICS BASED ON CONCEPTUAL SPACES
PETER GÄRDENFORS

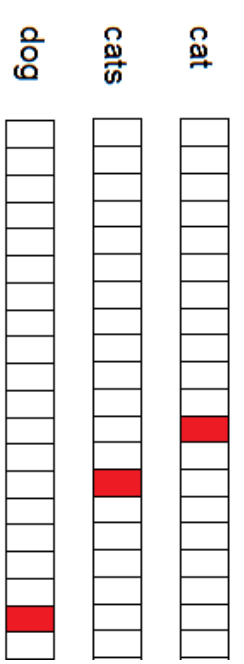
  - Constructing meaning from experience

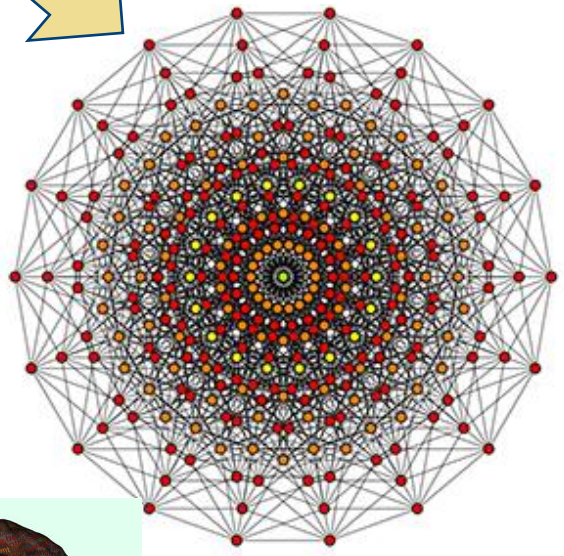
Word Embedding

- Term for a set of language modeling where **words** or phrases from the **vocabulary** are mapped to **vectors** of real numbers
- The idea that contextual information alone constitutes a viable representation of linguistic items

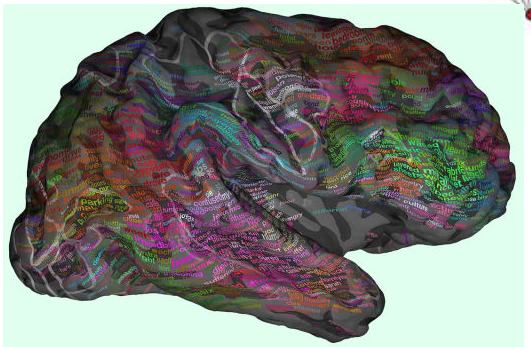
  - Provides vector representations of words such that the relationship between two vectors mirrors the linguistic relationship between the two words

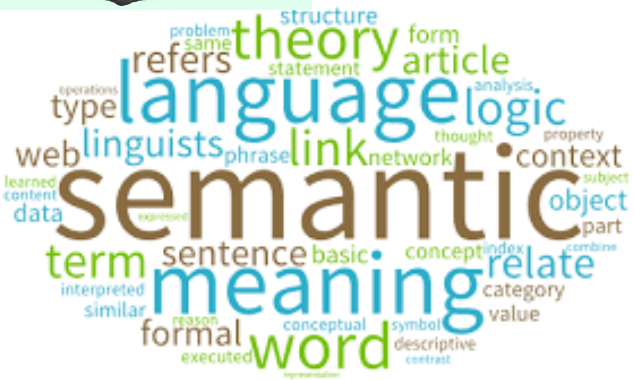
Word Embedding





- Involves a mathematical embedding from a space with one dimension (1 of *N*) per word to a continuous vector space with much lower dimension
- Relatively little is known about how humans compute meaning from experience
  - Not exactly true ☺
  - http://gallantlab.org/huth2016/
- Word Embeddings are computational methods trying to represent semantics of words





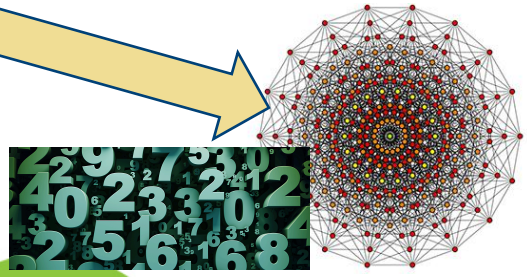
Word Embedding. Problem Formulation

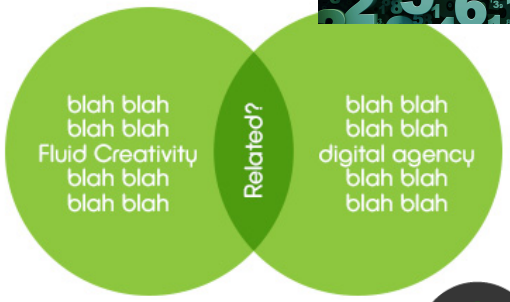
Given

- Vocabulary of words
- Large corpora of texts
- The goal is to learn vectors representing words in the vocabulary

TEXT

Word Embeddings





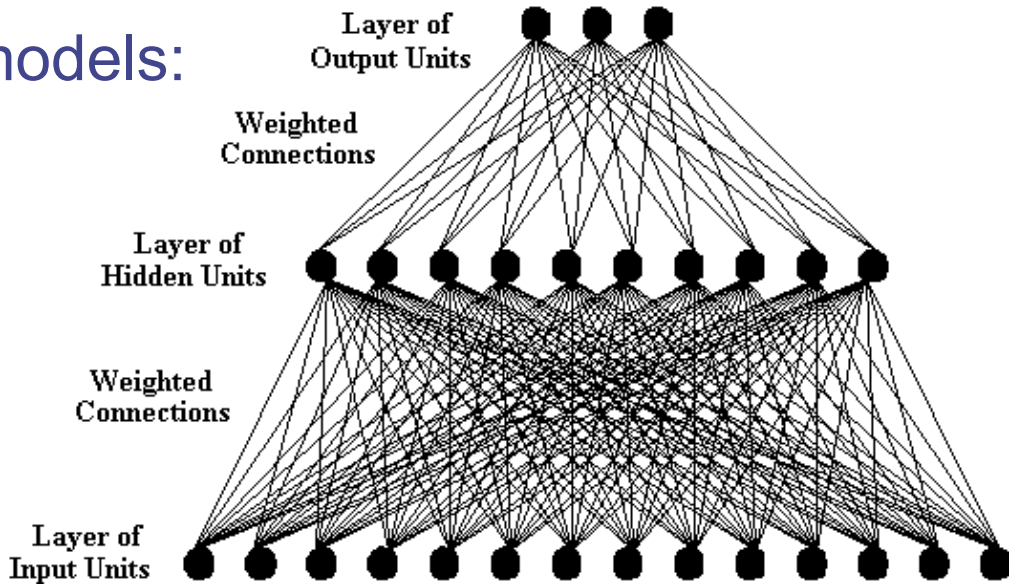
- Co-occurrences in text corpora are used for learning

- Distance between vectors is used for semantic comparison

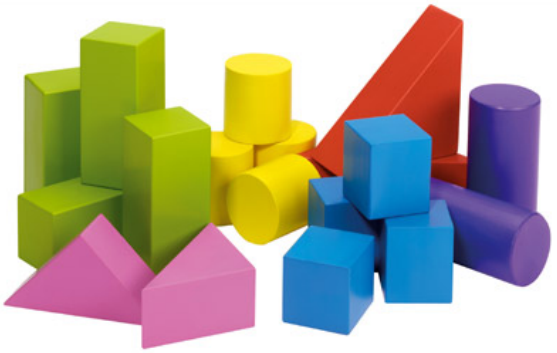
Word Embedding. Taxonomy

- Two major clusters of models:

connectionist models

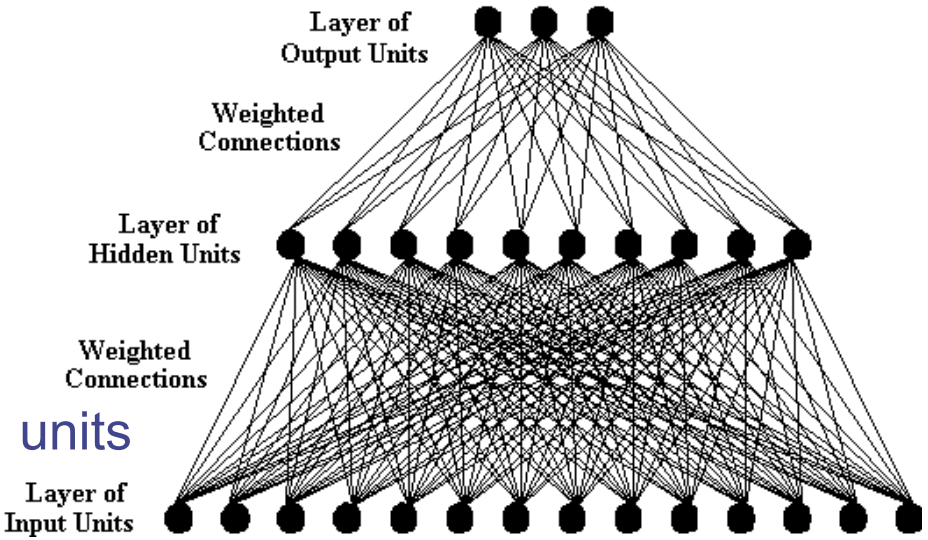


distributional models



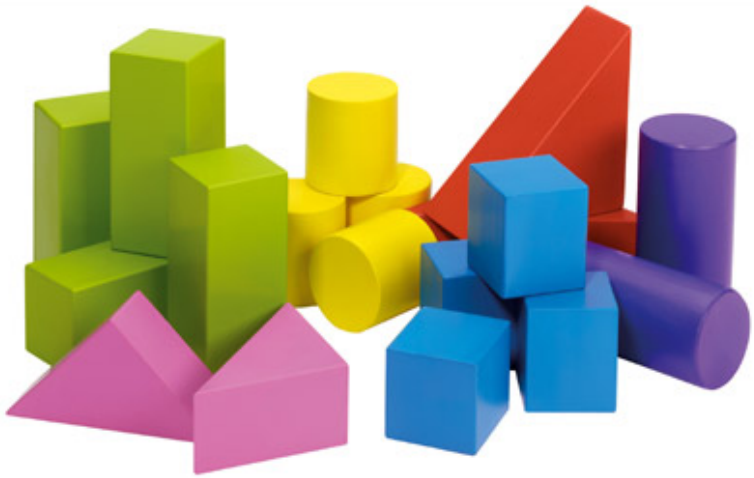
Word Embedding. Taxonomy

- Connectionist models or Artificial Neural Networks
  - Represent knowledge in terms of weighted connections between interconnected units
  - Several different architectures
  - At least one set of input units and one set of target or output units
  - Weights between units are initialized to a random state



Word Embedding. Taxonomy

- Distributional or corpus-based models
  - Hypothesize a mechanism to learn semantics from repeated episodic experience in a text corpus
  - “you shall know a word by the company it keeps”
  - Words are related as they are frequently used in similar contexts
  - The overall goal of formalizing the construction of semantic representations from statistical redundancies in language



Word Embedding Evaluation

- It is all great. But how do we evaluate it???

- A very common procedure evaluates judgments of semantic similarity

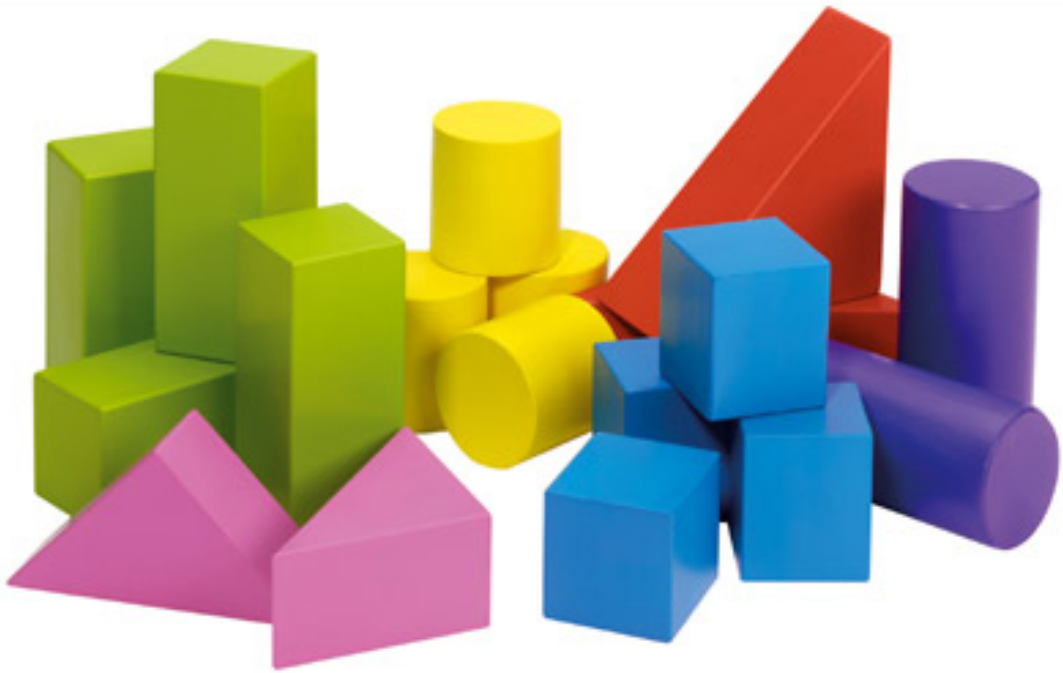
- Test of English as a Foreign Language (TOEFL) has a part related to the choice of a synonym for the given word

Word Embedding Evaluation

- For example, for word “Wildly” choose a synonym between

- A. “Distinctively”
- B. “Mysteriously”
- C. “Abruptly”
- **D**. “Furiously”

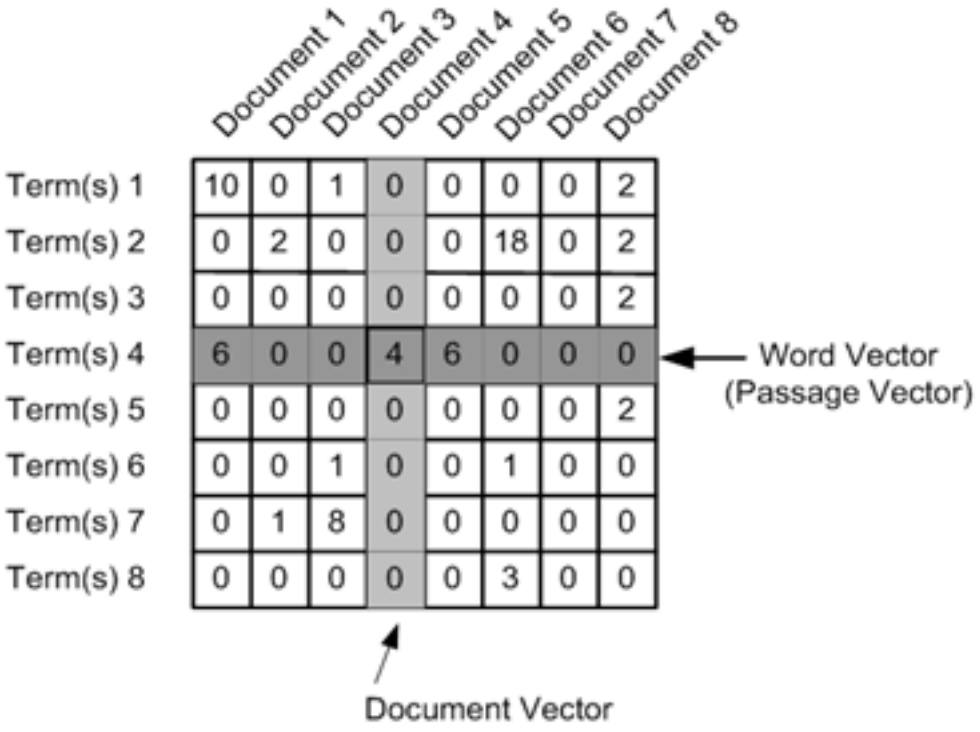
- The evaluation score is based on the number of the correct answers among 80 synonym tasks



Latent Semantic Analysis

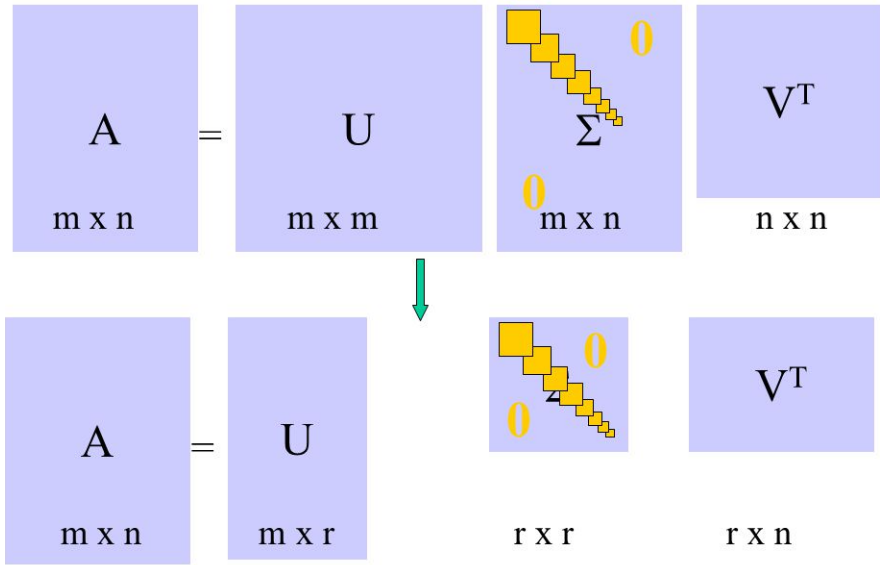
- The best-known distributional model. Landauer&Dumais, 1997

- Begins with a term-by-document frequency matrix of a text corpus
  - Each row vector is a word's frequency distribution over documents
  - A document is simply a "bag-of-words"



Latent Semantic Analysis

- The matrix is factorized using singular-value decomposition
  - SVD reduces dimensionality of row vectors usually to 300 elements
  - Brings out higher-order statistical relationships among words more sophisticated than mere direct co-occurrence
  - Even though two words (e.g., boat and ship) might have had zero similarity in the original matrix they may be proximal in the reduced space reflecting their deeper semantic similarity



- LSA achieved a score on TOEFL about 51% that would allow it entrance into most U.S. colleges.

Random Indexing

- LSA is computationally demanding due to SVD
- Begins with random vectors representing documents in corpus
  - Dimensionality is set a priori
  - Ternary elements {-1,0,1}
  - Sparse
  - Random

D1	0	1	0	0	1	-1	0	0	0	-1
D2	0	1	-1	0	-1	0	0	0	0	1
D3	1	-1	0	-1	0	0	1	0	0	0
D4	0	0	0	-1	1	-1	0	0	1	0
D5	0	1	-1	0	1	0	0	-1	0	0
D6	1	0	0	0	1	0	0	0	-1	-1
D7	-1	0	0	0	0	0	1	1	-1	0
D8	0	0	1	1	-1	0	0	0	0	-1
D9	1	0	1	-1	-1	0	0	0	0	0
D10	0	-1	0	0	-1	0	0	1	0	1

Random Indexing

- Representations for words are initially empty
- Every time a word is experienced in a document, word's representation is incremented by document's random vector
- Multiple runs of RI on the same corpus produce different vectors but overall similarity of the matrix is remarkably similar
- RI achieved a score on TOEFL about 52%

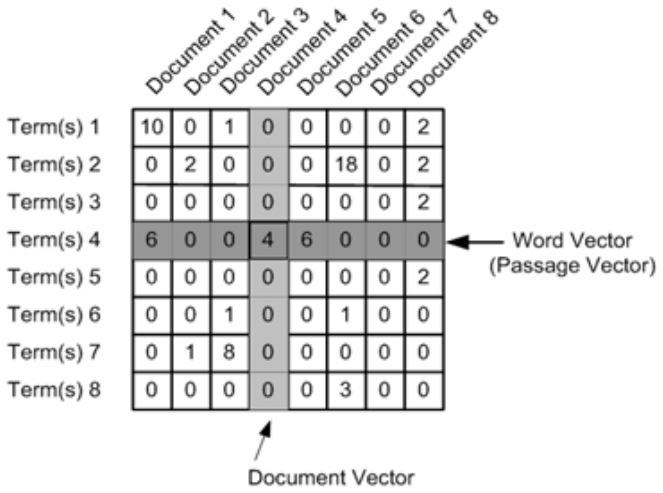
W1	0	0	0	0	0	0	0	0	0	0
W2	0	0	0	0	0	0	0	0	0	0
W3	0	0	0	0	0	0	0	0	0	0
W4	0	0	0	0	0	0	0	0	0	0
W5	0	0	0	0	0	0	0	0	0	0

W1	0	1	0	0	1	-1	0	0	0	-1
W2	0	0	0	0	0	0	0	0	0	0
W3	0	3	0	0	3	-3	0	0	0	-3
W4	0	0	0	0	0	0	0	0	0	0
W5	0	0	0	0	0	0	0	0	0	0

D1	-1	0	1	0	0	0	0	1	0	-1
D2	0	0	1	-1	0	0	0	-1	1	0
D3	0	0	0	0	0	1	-1	-1	1	0
D4	0	-1	0	0	1	0	0	1	0	-1
D5	0	0	0	0	1	0	-1	-1	1	0

Window as a context

- The original LSA and RI use documents as a context



- Instead, one can use surrounding words as a context

to be or ***not to*** be ***that is*** the question

4-word window

- Number of the considered words is called *N*-word window

Window as a context

- Moving-window models operationalize a word's context in terms of the other words that it is commonly seen with in temporal contexts
- Gradually develops semantic structure from simple co-occurrence counting
- With some adjustments allows to use word-order information
- Latent similarity naturally emerges if words tend to occur with similar context

Random Indexing with Permutations

- RI is modified to use surrounding words as a context and account for word-order information
- Begins with random representations for **words in vocabulary**
  - The same vectors as for RI
  - Fixed since the initialization
- Semantic representations for words are initially empty

not	0	1	-1	0	1	0	0	-1	0	0
to	1	0	0	0	1	0	0	0	-1	-1
be	-1	0	0	0	0	0	1	1	-1	0
that	0	0	1	1	-1	0	0	0	0	-1
is	1	0	1	-1	-1	0	0	0	0	0

not	0	0	0	0	0	0	0	0	0	0
to	0	0	0	0	0	0	0	0	0	0
be	0	0	0	0	0	0	0	0	0	0
that	0	0	0	0	0	0	0	0	0	0
is	0	0	0	0	0	0	0	0	0	0

Random Indexing with Permutations

not	0	1	-1	0	1	0	0	-1	0	0
to	1	0	0	0	1	0	0	0	-1	-1
be	-1	0	0	0	0	0	1	1	-1	0
that	0	0	1	1	-1	0	0	0	0	-1
is	1	0	1	-1	-1	0	0	0	0	0

- A semantic representation of a target word is updated with the sum of random representations of words within *N*-word window

to be or ***not to*** be ***that is*** the question

not	0	0	0	0	0	0	0	0	0	0
to	0	0	0	0	0	0	0	0	0	0
be	2	1	1	0	0	0	0	-1	-1	-2
that	0	0	0	0	0	0	0	0	0	0
is	0	0	0	0	0	0	0	0	0	0

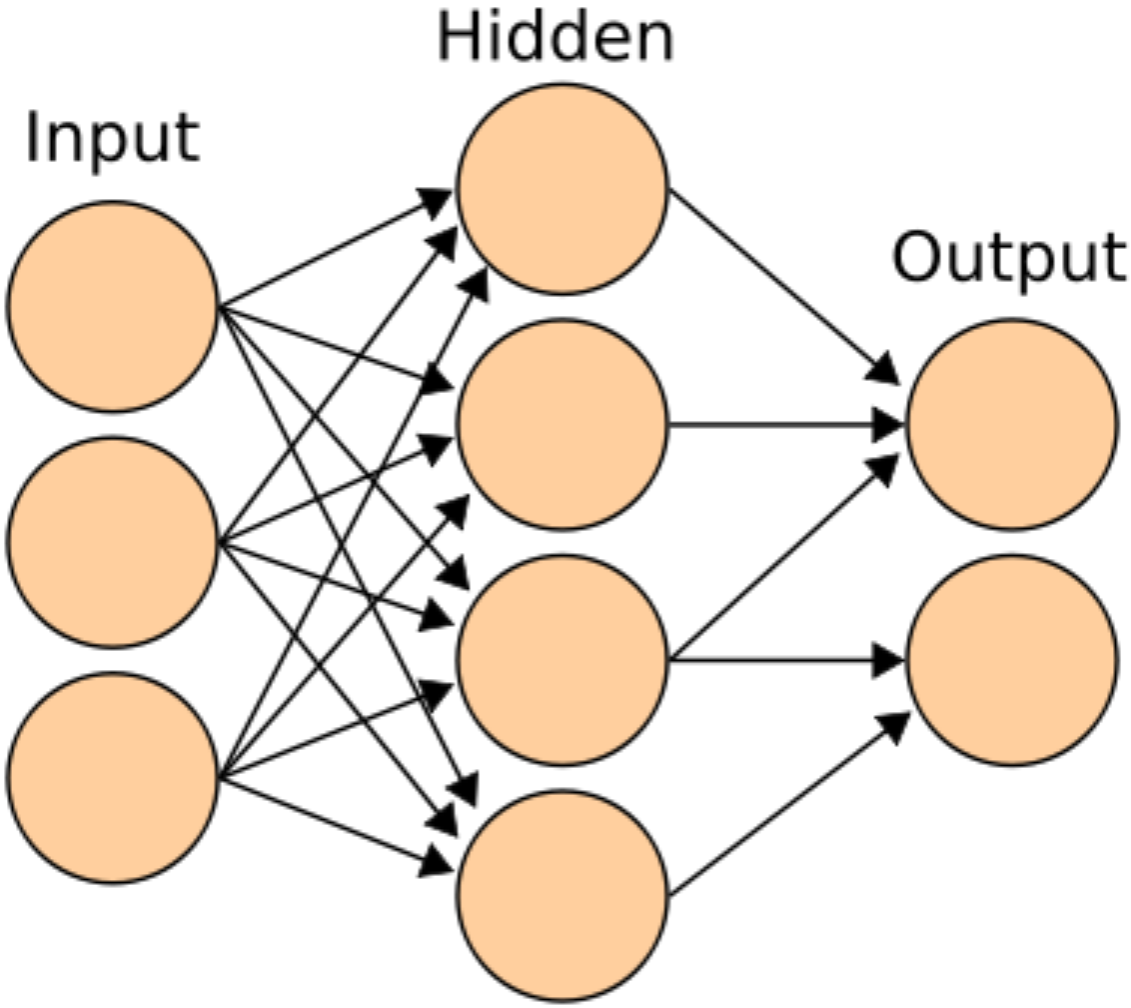
- Permutations of random representations are used to encode word-order

1 2 3 4

to be or ***not to*** be ***that is*** the

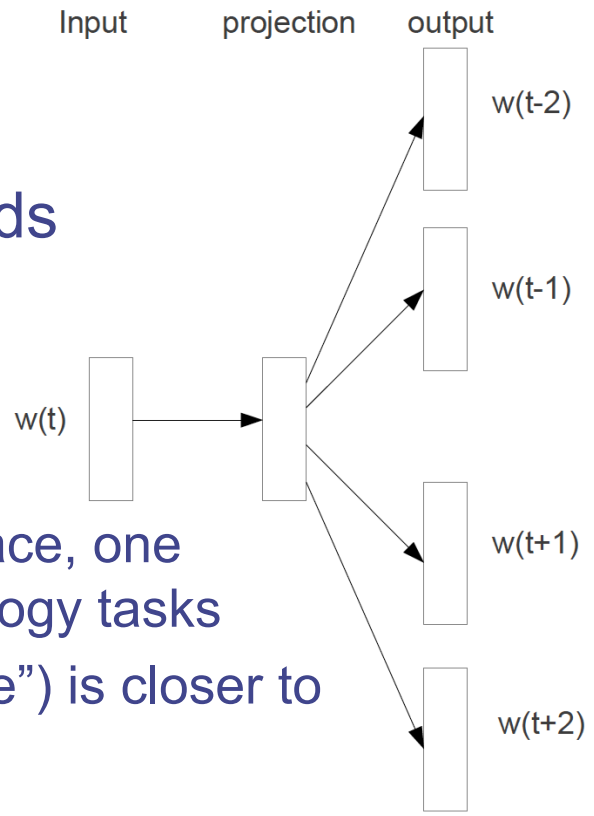
not	-1	0	1	0	0	-1	0	0	0	1
to	0	0	0	1	0	0	0	-1	-1	1
that	-1	0	0	1	1	-1	0	0	0	0
is	0	0	1	0	1	-1	-1	0	0	0

- RP achieved a score on TOEFL about 78%


Hidden
Input
Output

Neural Networks

- Skip-gram model
- The training objective is to learn vector representations predicting the nearby words
- Learned vectors explicitly encode many linguistic regularities and patterns
  - Can be represented as linear translations
  - Simple vector arithmetic in the embedding space, one can solve various syntactic and semantic analogy tasks
  - e.g. vec(“Madrid”) - vec(“Spain”) + vec(“France”) is closer to vec(“Paris”) than to any other word vector

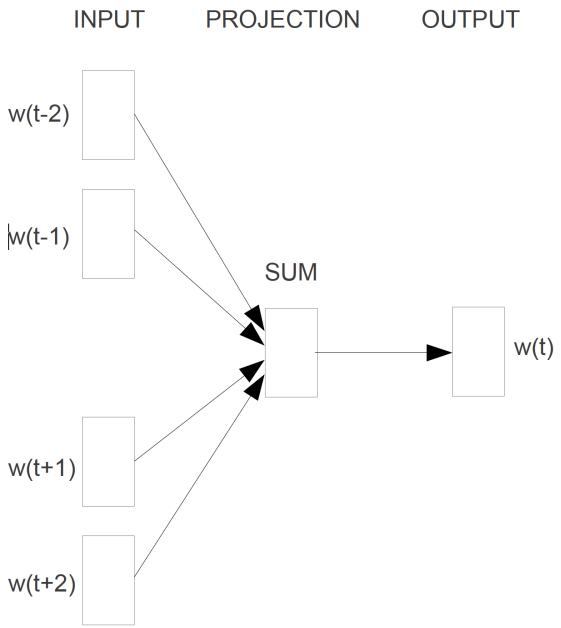


Neural Networks

- The training objective is to find word representations that are useful for predicting the surrounding words in a document
  - maximize the average log probability $\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$
- The basic Skip-gram formulation defines $p(w_{t+j}|w_t)$ using the softmax function $p(w_O|w_I)=\frac{\exp\left({v'_{w_O}}^{\top}v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left({v'_w}^{\top}v_{w_I}\right)}$
  - The basic formulation is impractical
- A comp. efficient approximation is the hierarchical softmax
  - a binary tree representation of the output layer with the *W* words
  - a binary Huffman tree assigning short codes to the frequent words

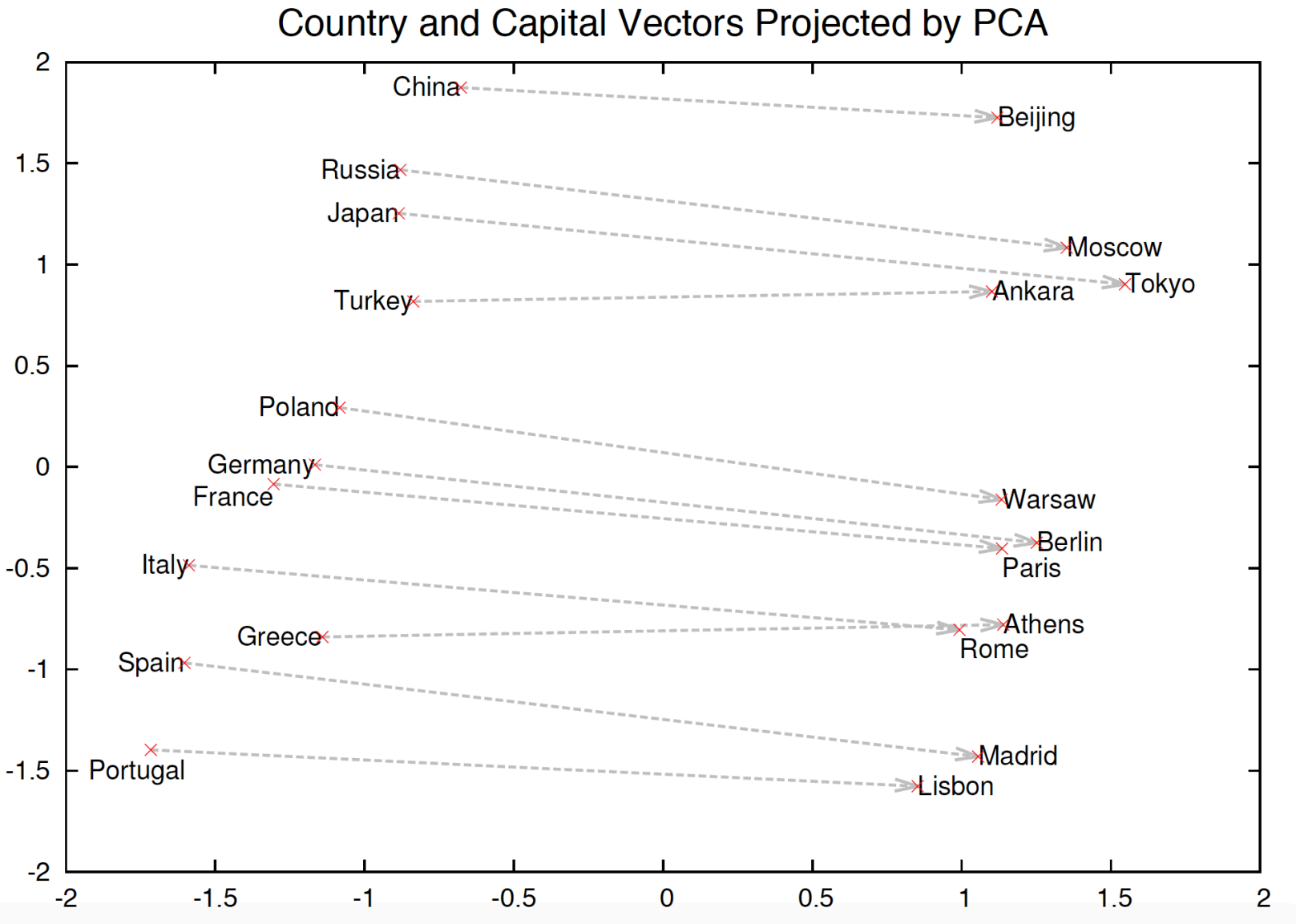
Neural Networks

- CBOW model of word2vec
- Given a number of context words around a target word w, these models formulate the embedding task as that of finding a representation that is good at predicting w from the context representations



- non-obvious degree of language understanding can be obtained by using basic mathematical operations on the word vector representations

Neural Networks



Evaluation methods for Word Embeddings

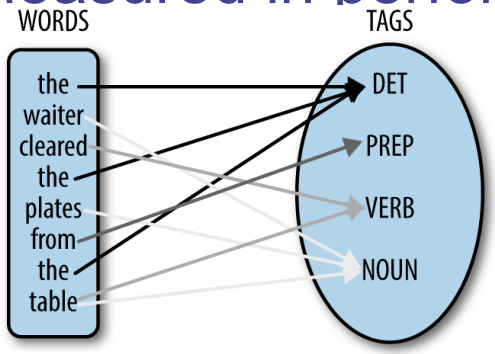
- How should word embedding models be compared?

- Should measure the quality of Word Embedding methods

- Two major categories: extrinsic and intrinsic evaluation

Evaluation methods for Word Embeddings

- Extrinsic: word embeddings are used as input features to a downstream task and changes are measured in performance metrics specific to that task
  - part-of-speech tagging
  - named-entity recognition



- Intrinsic: directly test for syntactic or semantic relationships between words
  - involve a pre-selected set of query terms and semantically related target words
  - evaluated by compiling an aggregate score for each method such as a correlation coefficient

Intrinsic evaluation

- Absolute intrinsic evaluation
  - Methods are evaluated individually and only final scores are compared

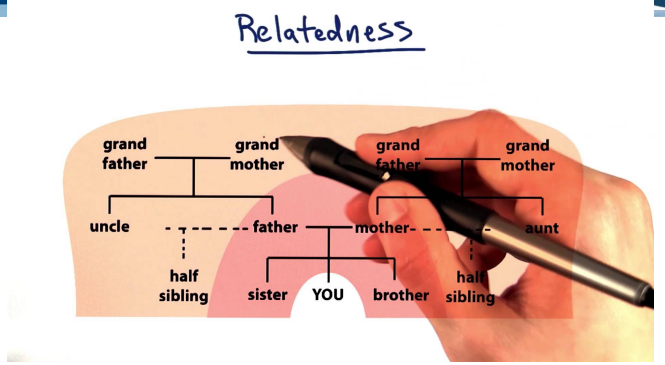
- Comparative intrinsic evaluation
  - People are asked directly for their preferences among the methods

- Comparative and absolute evaluations show similar results

Intrinsic evaluation

Four broad categories:

- Relatedness
- Analogy
- Categorization
- Selectional preference



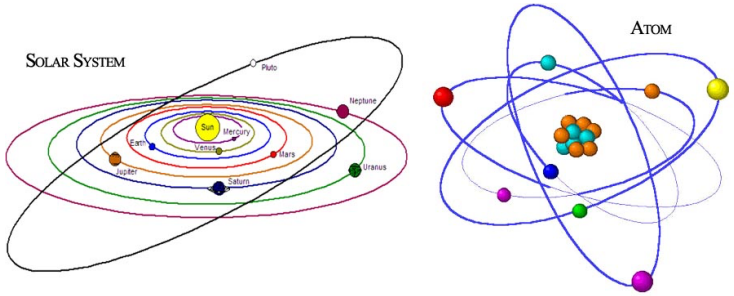
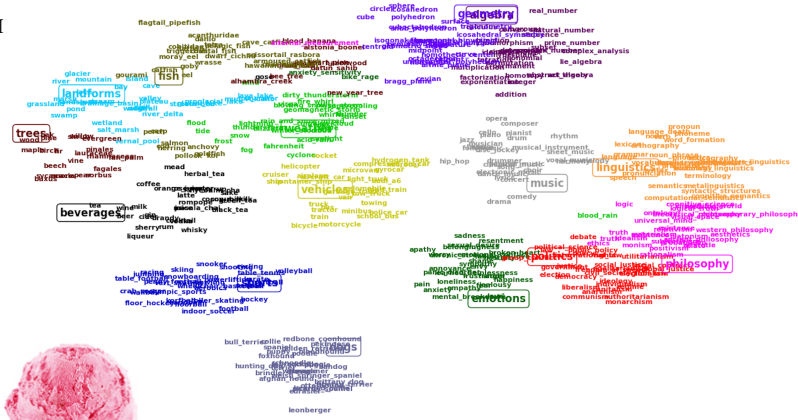
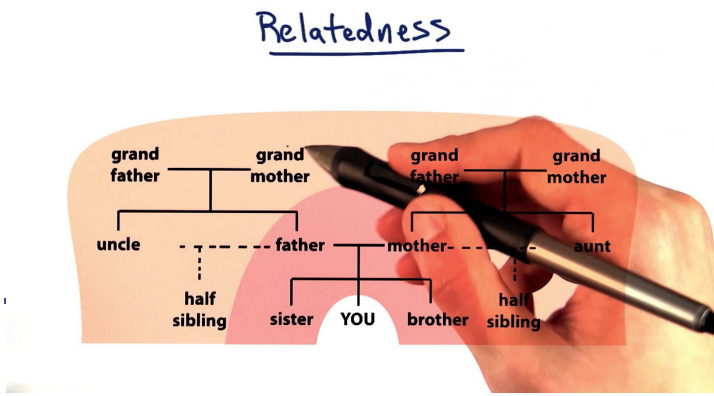


Diagram of the Solar System as compared to that of an Atom



Intrinsic evaluation

- Relatedness
  - Datasets contain relatedness scores for pairs of words; the cosine similarity of the embeddings for two words should have high correlation with human relatedness scores.
  - TOEFL synonyms are an example



- Analogy
  - The goal is to find a term *x* for a given term *y* so that *x* : *y* best resembles a sample relationship *a* : *b*.

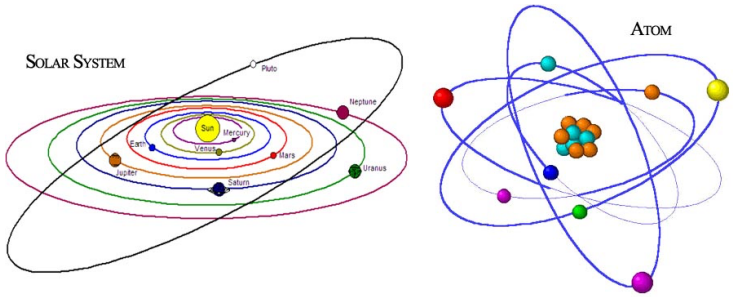
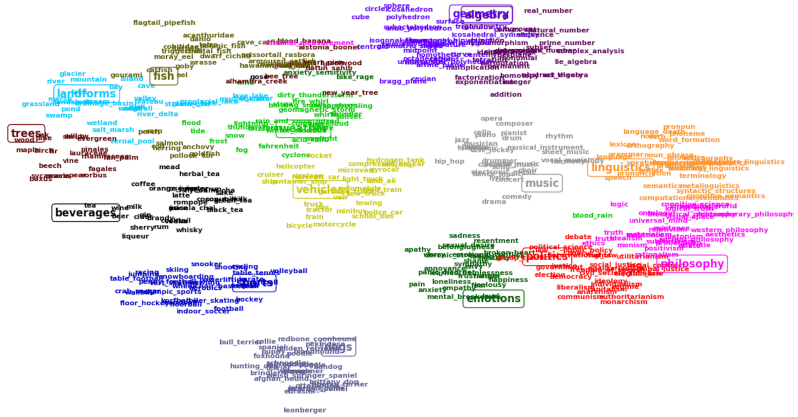


Diagram of the Solar System as compared to that of an Atom

Intrinsic evaluation

- Categorization
  - Goal is to recover a clustering of words into different categories
  - The corresponding word vectors of all words in a dataset are clustered
  - Purity of the clusters is computed with respect to the labeled dataset



- Selectional preference
  - Determine how typical a noun is for a verb either as a subject or as an object (e.g., people eat, but we rarely eat people)

Intrinsic evaluation

- Important design questions come up when designing reusable datasets for relatedness evaluation and other scenarios
- Query inventory
  - How the word pairs are picked affects the results of evaluation
  - The frequency of the words in the language
  - The parts of speech of the words
  - Abstractness vs. concreteness of the terms
- Metric aggregation
  - Shortcoming of correlation-based metrics is that they aggregate scores of different pairs
  - Scores can vary greatly in the embedding space
  - Comparative evaluation does not require metric

Intrinsic evaluation: Coherence

- Assess whether groups of words in a small neighborhood in the embedding space are mutually related

- Good embeddings should have coherent neighborhoods for each word

(a)	finally	(b)	eventually
(c)	immediately	(d)	put

(a) - query
(b) - the nearest neighbor
(c) - 2nd nearest neighbor
(d) - intruder

Extrinsic evaluation

- Measures the contribution of a word embedding model to a specific task
- Different tasks favor different embeddings
- Noun phrase chunking
- Sentiment classification
  - Binary classification
  - Based on movie reviews
  - Review is a linear combination of word embeddings

- Comparing performance across tasks may provide insight into the information encoded by an embedding
- Should not expect any specific task to act as a proxy for abstract quality

Commercial usage of Word Embedding

There are several companies across the world applying methods of word embedding in commercial applications.

- USA. MultiModel Research: http://multimodelresearch.com/index.html
- Products: http://multimodelresearch.com/multi-model-services.html
  - Multi-label classification: http://multimodelresearch.com/multi-model-label-classification-demo.html
  - Add Classification to articles or emails
  - Suggest medical codes automatically from patient data
  - Find Sentiment
  - etc.

Commercial usage of Word Embedding

- Austria. Cortical.IO: http://www.cortical.io
- Demonstrations
  - http://www.cortical.io/demos.html
  - Keyword Extraction
  - Language detection
- Products
  - Classification
  - Semantic Search
  - Streaming Text Filter
- https://www.youtube.com/watch?v=g3ZxJokDpds

Commercial usage of Word Embedding

- Sweden. Gavagai: https://gavagai.se
- Gavagai Living Lexicon
  - http://lexicon.gavagai.se
- Products
  - Text Analytics
  - Online Monitoring

- They have a nice blog: https://gavagai.se/blog/
  - A brief history of word embeddings https://gavagai.se/blog/2015/09/30/a-brief-history-of-word-embeddings/

Commercial usage of Word Embedding

Take a chance and listen to ***Gavagai founders*** in March at LTU

- Topic: Applications of unsupervised semantic models of large-scale streaming text data
- Date&time: March 16 at 10:00
- Place: A3101c

Magnus Sahlgren

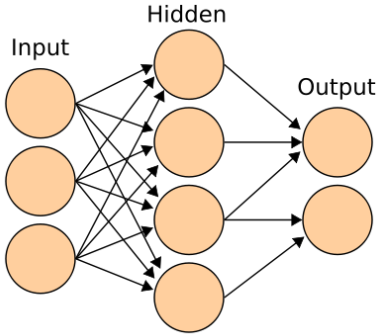
Jussi Karlgren

Assessment: RP and RNN

You get:

- Large text corpora
- 80 synonym tasks from TOEFL
- Word2Vec library
- The stub implementing Random Indexing

Vs



Goals are:

- Solve synonym tasks by both methods
- Vary the parameters and observe how accuracy changes
- Compare the (best) accuracy of methods
- Qualitatively (faster/slower) compare performance of methods

Literature

- S. Russell, P. Norvig. Artificial Intelligence: A modern Approach. Chapter 22.
- D. Widdows. Geometry and Meaning
- Everything You Need to Know about Natural Language Processing. http://www.kdnuggets.com/2015/12/natural-language-processing-101.html
- Wikipedia. Word embedding. https://en.wikipedia.org/wiki/Word_embedding
- M. N. Jones, J. Willits, S. Dennis. Models of Semantic Memory
- T. Landauer, S. Dumais. A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge
- M. Sahlgren, A. Holst, P. Kanerva. Permutations as a means to encode order in word space
- T. Schnabel, I. Labutov, D. Mimno, T. Joachims. Evaluation methods for unsupervised word embeddings
- T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean. Distributed Representations of Words and Phrases and their Compositionality
- T. Mikolov, K. Chen, G. Corrado, J. Dean. Efficient Estimation of Word Representations in Vector Space