# Intelligent Agents

## Sequential Structures, Word Semantics, and Embeddings

Prof. Dr. Ralf Möller
Universität zu Lübeck
Institut für Informationssysteme

# Motivation: Part Of Speech Tagging

- Annotate each word in a sentence with a part-of-speech (POS) tags.
- Lowest level of syntactic analysis.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| John | saw | the | saw | and | decided | to | take | it | to | the | table. |
| NNP | VBD | DT | NN | CC | VBD | TO | VB | PRP | IN | DT | NN |

- Useful for subsequent syntactic parsing and word sense disambiguation

- Topic modeling as discussed before could be extended to better consider POS tags



Abbreviations: https://sites.google.com/site/partofspeechhelp/home

# Information Extraction

- Identify phrases in language that refer to specific types of entities and relations in text.
- Named entity recognition is the task of identifying names of people, places, organizations, etc. in text.

  people organizations places

  - Michael Dell is the CEO of Dell Computer Corporation and lives in Austin Texas.
- Extract pieces of information relevant to a specific application, e.g. used car ads:

  make model year mileage price

  - For sale, 2002 Toyota Prius, 20,000 mi, $15K or best offer. Available starting July 30, 2006.

# Semantic Role Labeling

- For each clause, determine the semantic role played by each noun phrase that is an argument to the verb.
  agent patient source destination instrument
  - John drove Mary from Austin to Dallas in his Toyota Prius.
  - The hammer broke the window.
- Also referred to as "case role analysis," "thematic analysis," and "shallow semantic parsing"

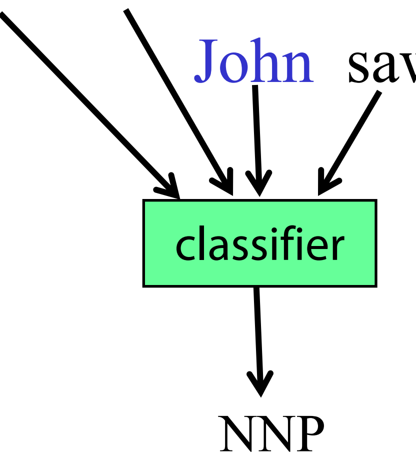
# Sequence Labeling as Classification

## Using Outputs as Inputs

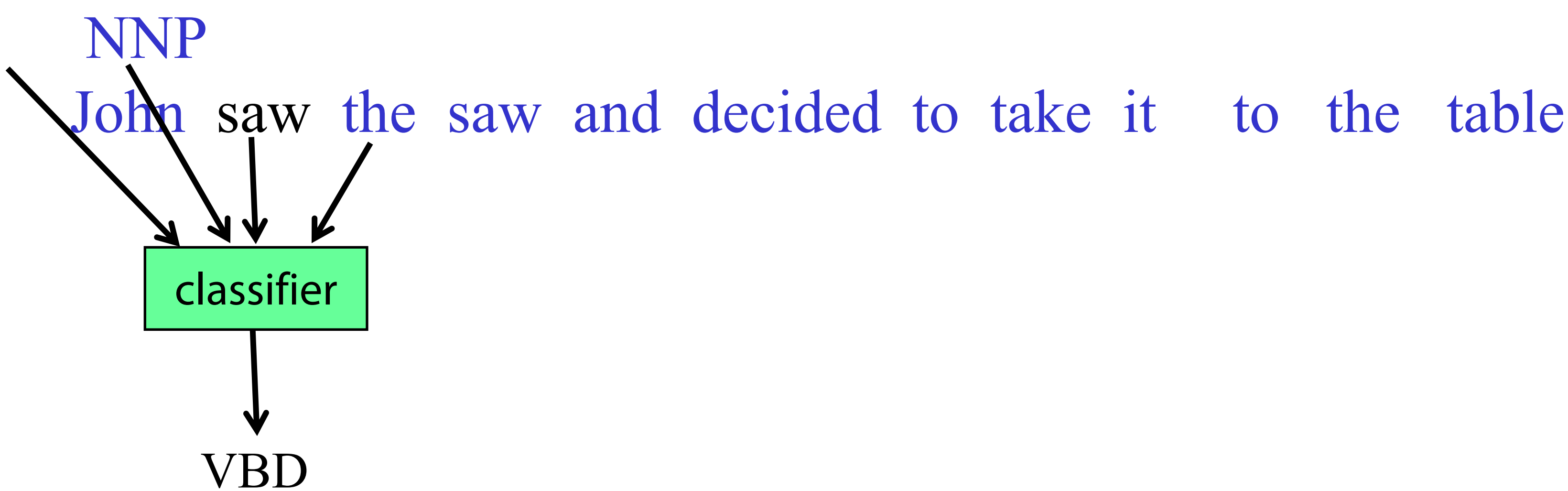
- Better input features are usually the categories of the surrounding tokens, but these are not available yet.
- Can use category of either the preceding or succeeding tokens by going forward or back and using previous output.

# Forward Classification

John saw the saw and decided to take it to the table.



# Forward Classification



# Forward Classification

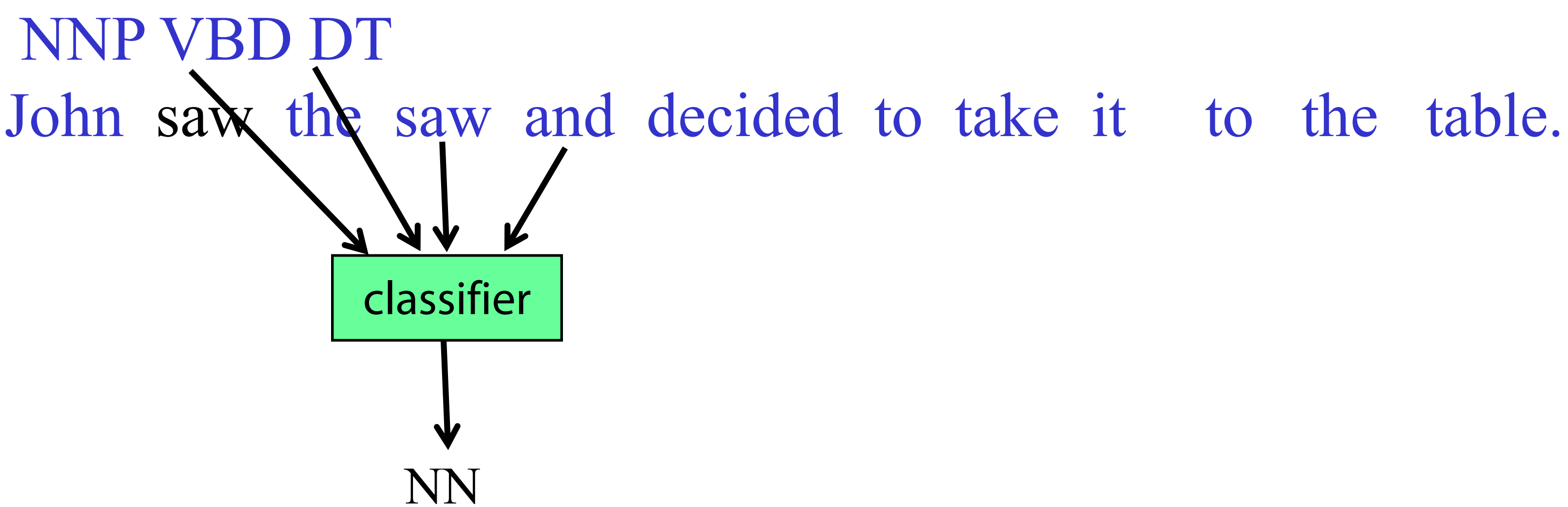
NNP VBD

John saw the saw and decided to take it to the table.

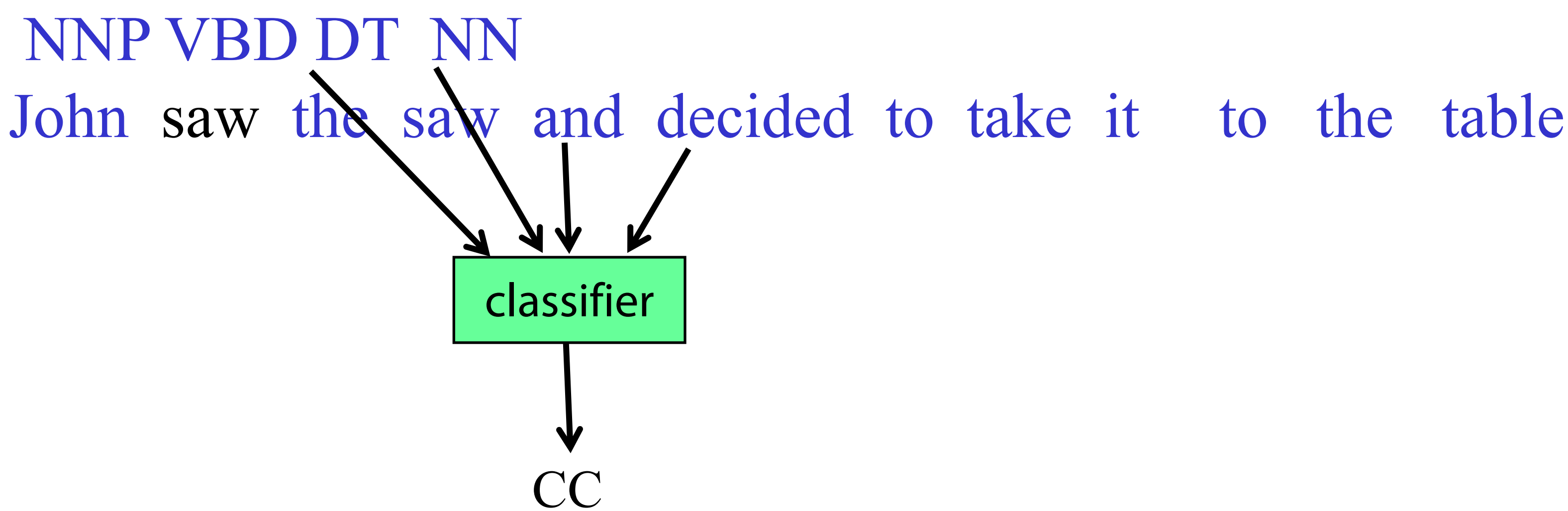
classifier

DT

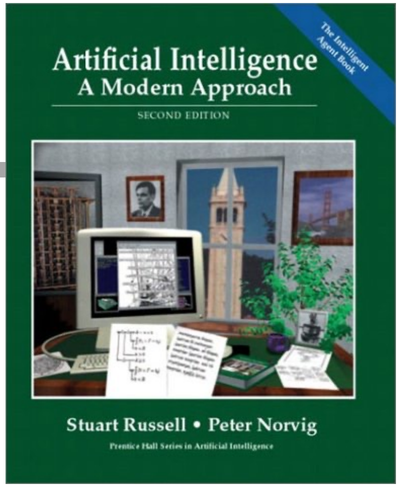
# Forward Classification

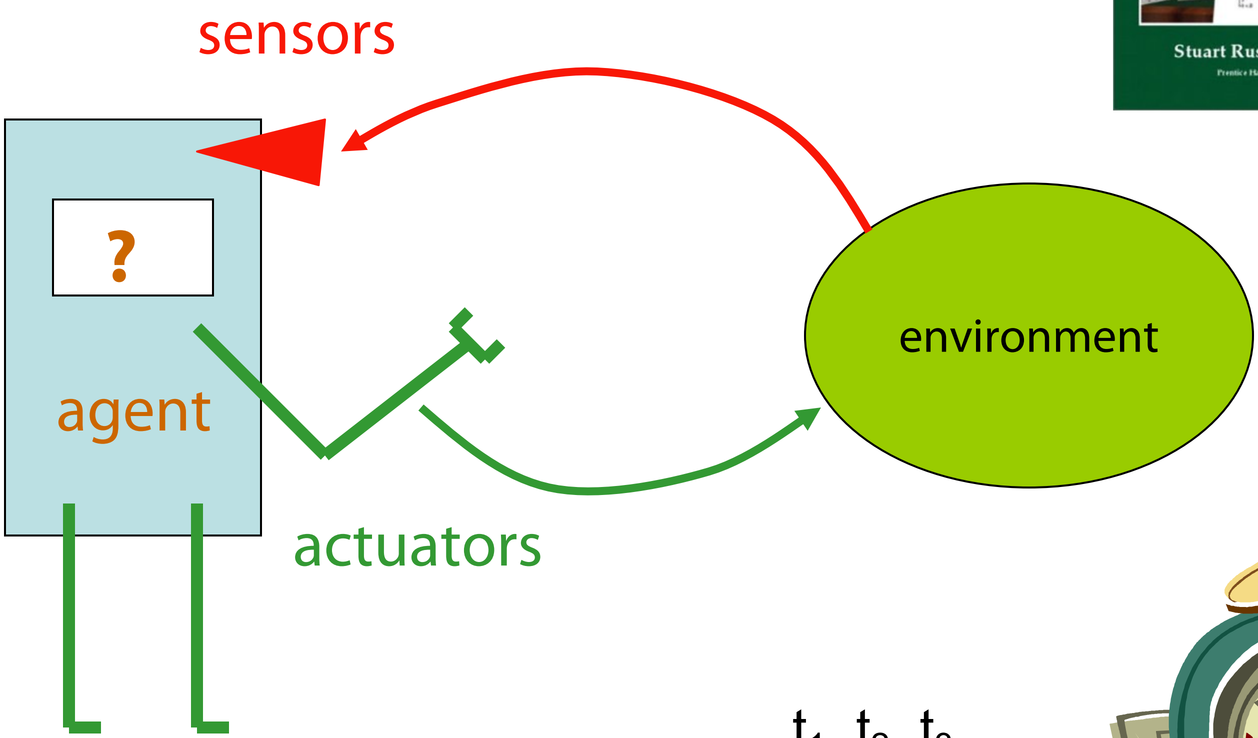


# Forward Classification



# More general perspective…




$t_1$, $t_2$, $t_3$, …

# Dynamic Topic Models

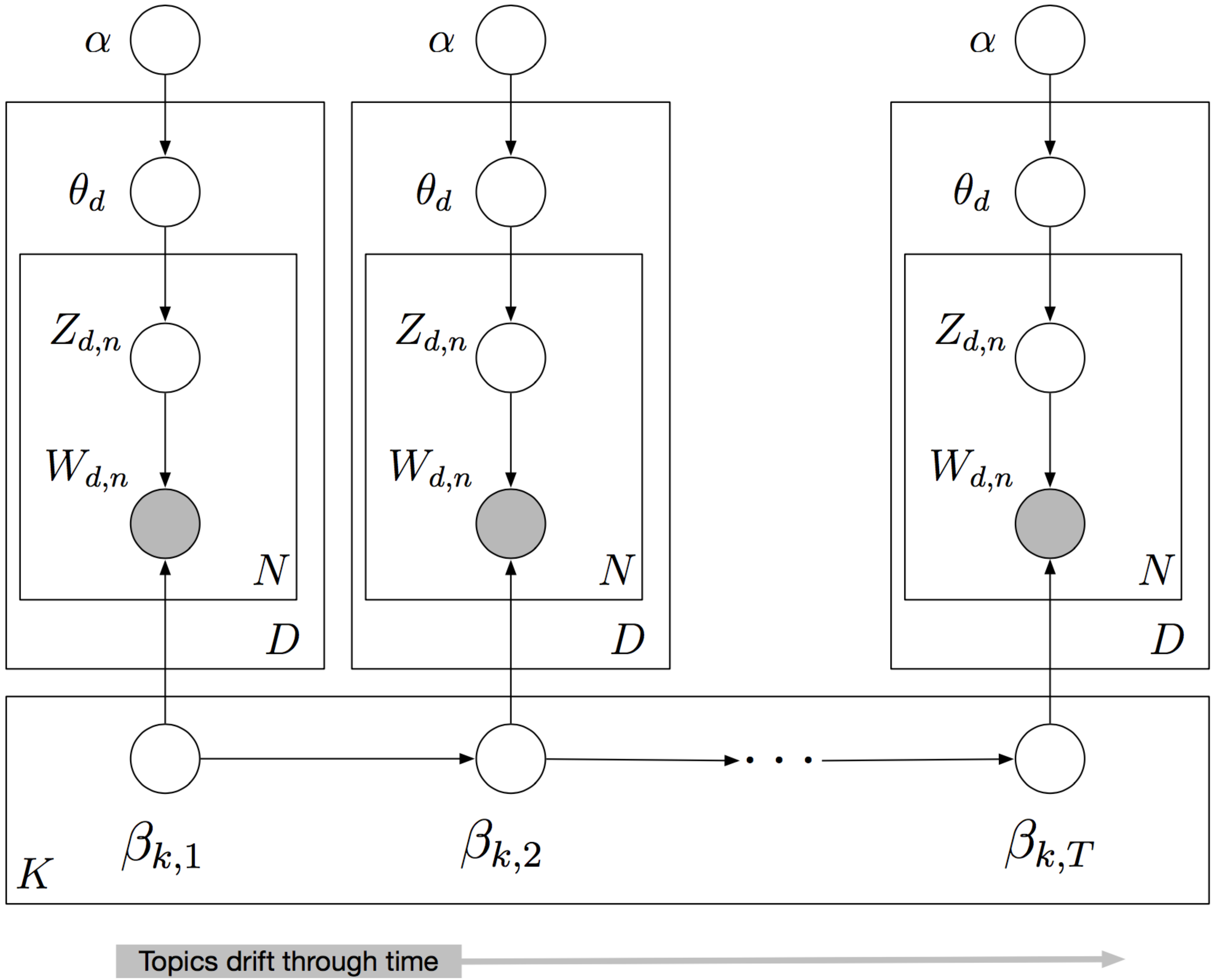
- In LDA the order of documents does not matter
- Not appropriate for sequential corpora (e.g., that span hundreds of years)
- Further, we may want to track how language changes over time
- Let the topics *drift* in a sequence.


My fellow citizens: I stand here today humbled by the task before us, grateful for the trust you have bestowed, mindful of the sacrifices borne by our ancestors...

AMONG the vicissitudes incident to life no event could have filled me with greater anxieties than that of which the notification was transmitted by your order...

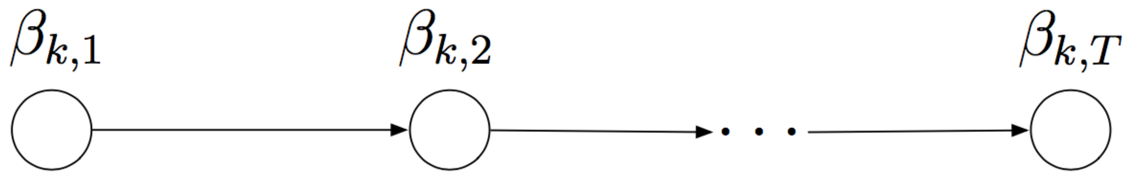


David M. Blei and John D. Lafferty. Dynamic topic models.
In Proc. ICML '06. pp. 113-120. **2006**.



α
θ_d
Z_{d,n}
W_{d,n}
N
D
K
β_{k,1}
β_{k,2}
β_{k,T}
Topics drift through time

# Dynamic Topic Models




- Use a logit normal distribution to model topics evolving over time
- Embed it in a state-space model on the log of the topic distribution

$$\begin{aligned} \beta_{t,k} \,|\, \beta_{t-1,k} &\sim \mathcal{N}(\beta_{t-1,k}, I\sigma^2) \\ p(w \,|\, \beta_{t,k}) &\propto \exp\{\beta_{t,k}\} \end{aligned}$$

- Let us make inferences about sequences of documents

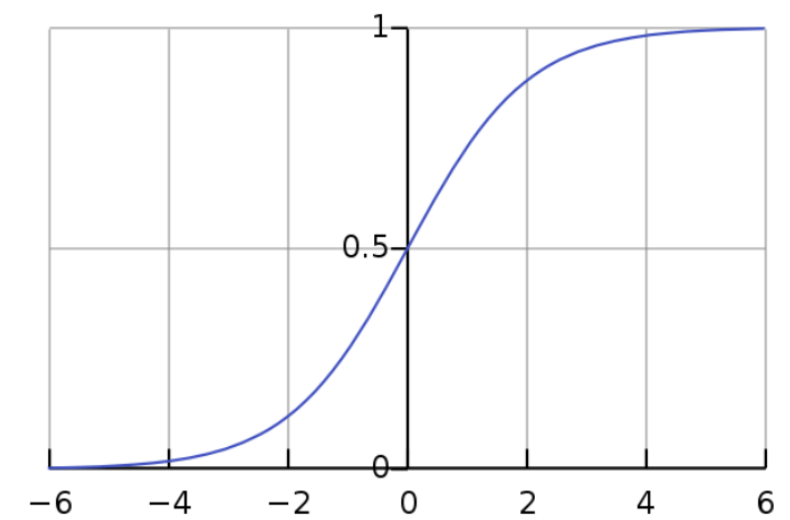
# Recap: Logistic function

A **logistic function** or **logistic curve** is a common "S" shape (sigmoid curve), with equation:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

where

- $e$ = the natural logarithm base (also known as Euler's number),
- $x_0$ = the $x$-value of the sigmoid's midpoint,
- $L$ = the curve's maximum value, and
- $k$ = the steepness of the curve.[1]




Standard logistic sigmoid function i.e. $L = 1, k = 1, x_0 = 0$

# Logit Normal Distribution

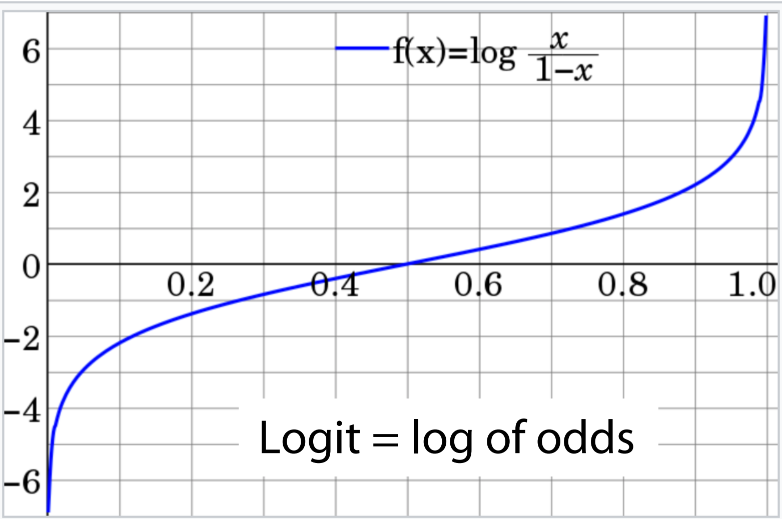
Normal Distribution

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

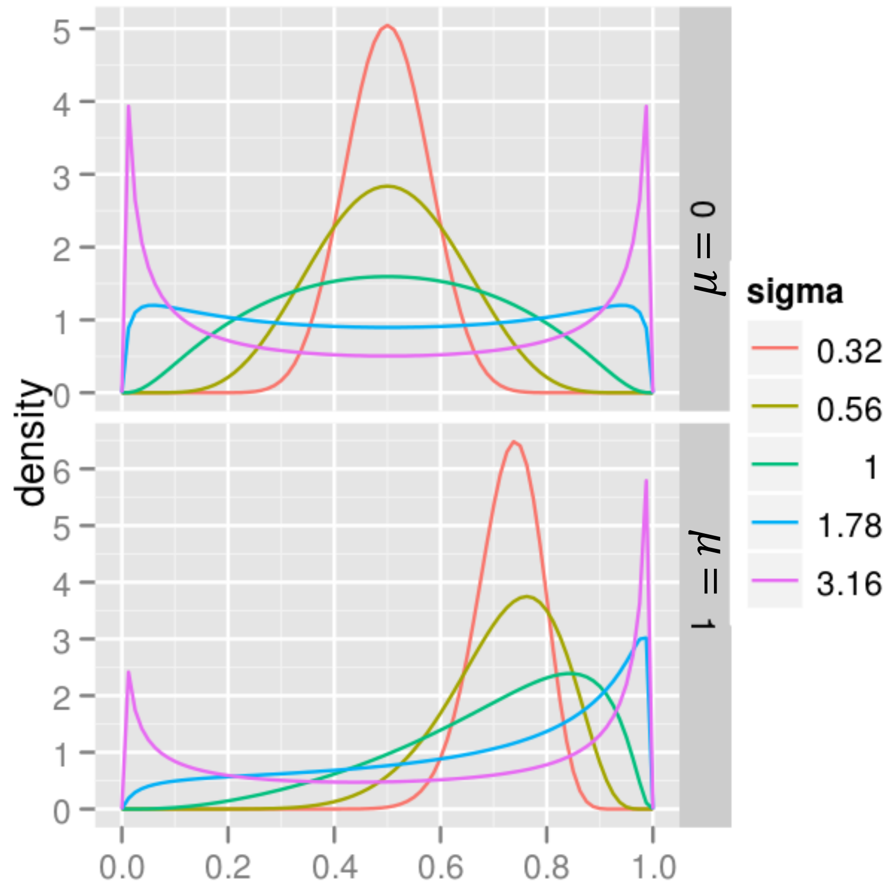
The probability density function (PDF) of a logit-normal distribution, for $0 \le x \le 1$, is:

$$f_X(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \frac{1}{x(1-x)} e^{-\frac{(\mathrm{logit}(x)-\mu)^2}{2\sigma^2}}$$

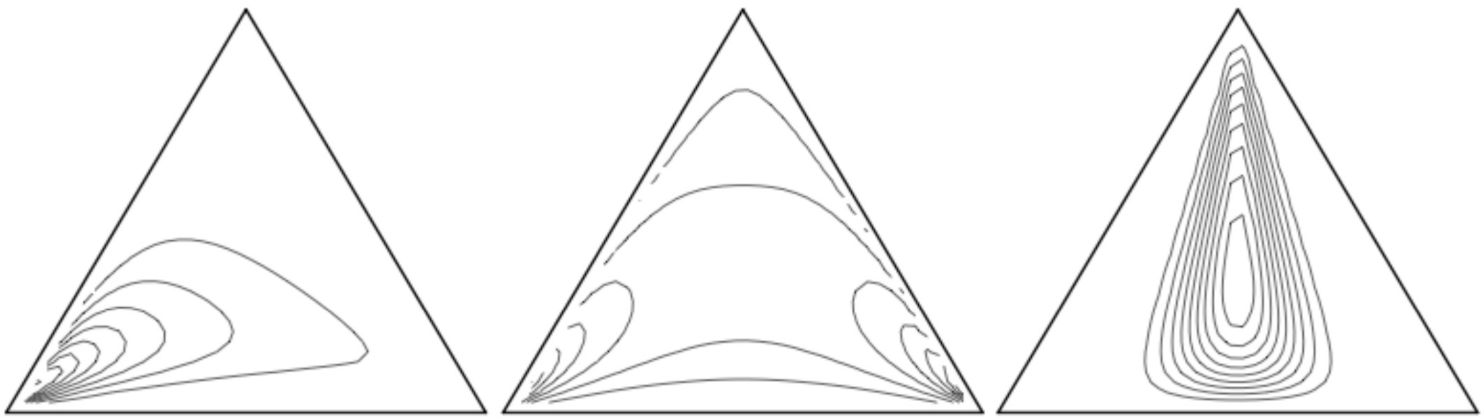
where $\mu$ and $\sigma$ are the mean and standard deviation of the variable's logit (by definition, the variable's logit is normally distributed).




Plot of logit($p$) in the domain of 0 to 1, where the base of logarithm is $e$



# $\beta_{t,k}$ is a multinomial: Simplex again

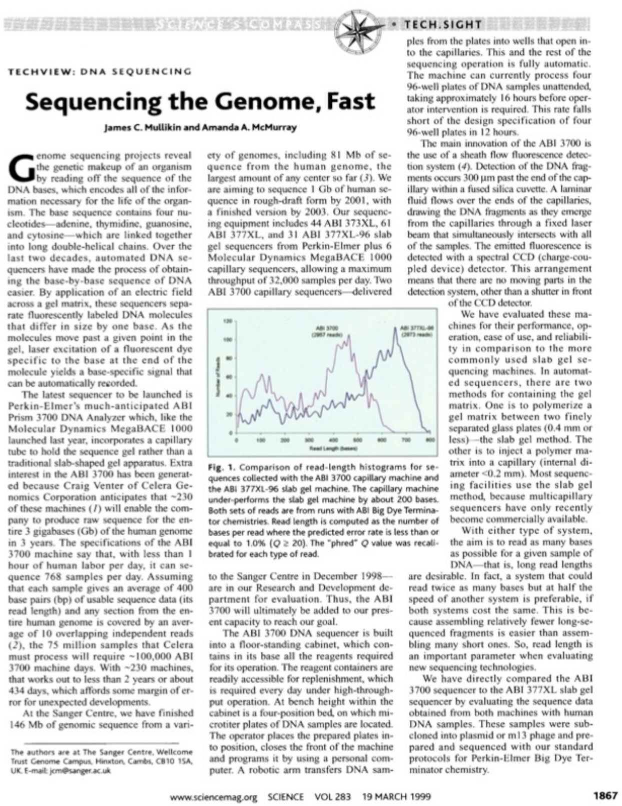


- The **logistic normal** is a distribution on the simplex that can model dependence between components (Aitchison, 1980).
- The log of the parameters of the multinomial are drawn from a multivariate Gaussian distribution,

$$
\begin{aligned}
X &\sim \mathcal{N}_K(\mu, \Sigma) \\
\theta_i &\propto \exp\{x_i\}.
\end{aligned}
$$

# Dynamic Topic Models

**Original article**



SCIENCE'S COMPASS • TECH.SIGHT

TECHVIEW: DNA SEQUENCING

## Sequencing the Genome, Fast

James C. Mullikin and Amanda A. McMurray

Genome sequencing projects reveal the genetic makeup of an organism by reading off the sequence of the DNA bases, which encodes all of the information necessary for the life of the organism. The base sequence contains four nucleotides—adenine, thymidine, guanosine, and cytosine—which are linked together into long double-helical chains. Over the last two decades, automated DNA sequencers have made the process of obtaining the base-by-base sequence of DNA easier. By application of an electric field across a gel matrix, these sequencers separate fluorescently labeled DNA molecules that differ in size by one base. As the molecules move past a given point in the gel, laser excitation of a fluorescent dye specific to the base at the end of the molecule yields a base-specific signal that can be automatically recorded.

The latest sequencer to be launched is Perkin-Elmer's much-anticipated ABI Prism 3700 DNA Analyzer which, like the Molecular Dynamics MegaBACE 1000 launched last year, incorporates a capillary tube to hold the sequence gel rather than a traditional slab-shaped gel apparatus. Extra interest in the ABI 3700 has been generated because Craig Venter of Celera Genomics Corporation anticipates that ~230 of these machines (*1*) will enable the company to produce raw sequence for the entire 3 gigabases (Gb) of the human genome in 3 years. The specifications of the ABI 3700 machine say that, with less than 1 hour of human labor per day, it can sequence 768 samples per day. Assuming that each sample gives an average of 400 base pairs (bp) of usable sequence data (its read length) and any section from the entire human genome is covered by an average of 10 overlapping independent reads (*2*), the 75 million samples that Celera must process will require ~100,000 ABI 3700 machine days. With ~230 machines, that works out to less than 2 years or about 434 days, which affords some margin of error for unexpected developments.

At the Sanger Centre, we have finished 146 Mb of genomic sequence from a variety of genomes, including 81 Mb of sequence from the human genome, the largest amount of any center so far (*3*). We are aiming to sequence 1 Gb of human sequence in rough-draft form by 2001, with a finished version by 2003. Our sequencing equipment includes 44 ABI 373XL, 61 ABI 377XL, and 31 ABI 377XL-96 slab gel sequencers from Perkin-Elmer plus 6 Molecular Dynamics MegaBACE 1000 capillary sequencers, allowing a maximum throughput of 32,000 samples per day. Two ABI 3700 capillary sequencers—delivered to the Sanger Centre in December 1998—are in our Research and Development department for evaluation. Thus, the ABI 3700 will ultimately be added to our present capacity to reach our goal.

The ABI 3700 DNA sequencer is built into a floor-standing cabinet, which contains in its base all the reagents required for its operation. The reagent containers are readily accessible for replenishment, which is required every day under high-throughput operation. At bench height within the cabinet is a four-position bed, on which microtiter plates of DNA samples are located. The operator places the prepared plates into position, closes the front of the machine and programs it by using a personal computer. A robotic arm transfers DNA samples from the plates into wells that open into the capillaries. This and the rest of the sequencing operation is fully automatic. The machine can currently process four 96-well plates of DNA samples unattended, taking approximately 16 hours before operator intervention is required. This rate falls short of the design specification of four 96-well plates in 12 hours.

The main innovation of the ABI 3700 is the use of a sheath flow fluorescence detection system (*4*). Detection of the DNA fragments occurs 300 μm past the end of the capillary within a fused silica cuvette. A laminar fluid flows over the ends of the capillaries, drawing the DNA fragments as they emerge from the capillaries through a fixed laser beam that simultaneously intersects with all of the samples. The emitted fluorescence is detected with a spectral CCD (charge-coupled device) detector. This arrangement means that there are no moving parts in the detection system, other than a shutter in front of the CCD detector.

We have evaluated these machines for their performance, operation, ease of use, and reliability in comparison to the more commonly used slab gel sequencing machines. In automated sequencers, there are two methods for containing the gel matrix. One is to polymerize a gel matrix between two finely separated glass plates (0.4 mm or less)—the slab gel method. The other is to inject a polymer matrix into a capillary (internal diameter <0.2 mm). Most sequencing facilities use the slab gel method, because multicapillary sequencers have only recently become commercially available.

With either type of system, the aim is to read as many bases as possible for a given sample of DNA—that is, long read lengths are desirable. In fact, a system that could read twice as many bases but at half the speed of another system is preferable, if both systems cost the same. This is because assembling relatively fewer long-sequenced fragments is easier than assembling many short ones. So, read length is an important parameter when evaluating new sequencing technologies.

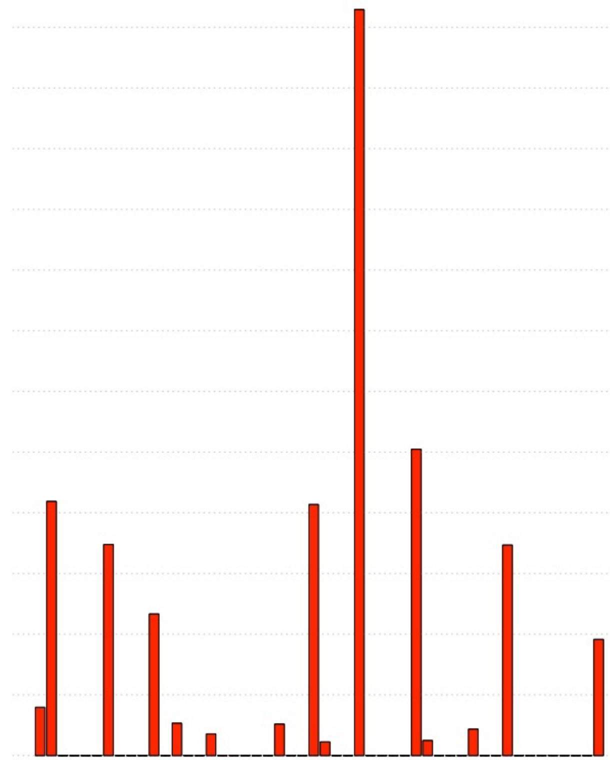
We have directly compared the ABI 3700 sequencer to the ABI 377XL slab gel sequencer by evaluating the sequence data obtained from both machines with human DNA samples. These samples were subcloned into plasmid or m13 phage and prepared and sequenced with our standard protocols for Perkin-Elmer Big Dye Terminator chemistry.

**Fig. 1.** Comparison of read-length histograms for sequences collected with the ABI 3700 capillary machine and the ABI 377XL-96 slab gel machine. The capillary machine under-performs the slab gel machine by about 200 bases. Both sets of reads are from runs with ABI Big Dye Terminator chemistries. Read length is computed as the number of bases per read where the predicted error rate is less than or equal to 1.0% ($Q \geq 20$). The "phred" $Q$ value was recalibrated for each type of read.

The authors are at The Sanger Centre, Wellcome Trust Genome Campus, Hinxton, Cambs, CB10 1SA, UK. E-mail: jcm@sanger.ac.uk



**Topic proportions**

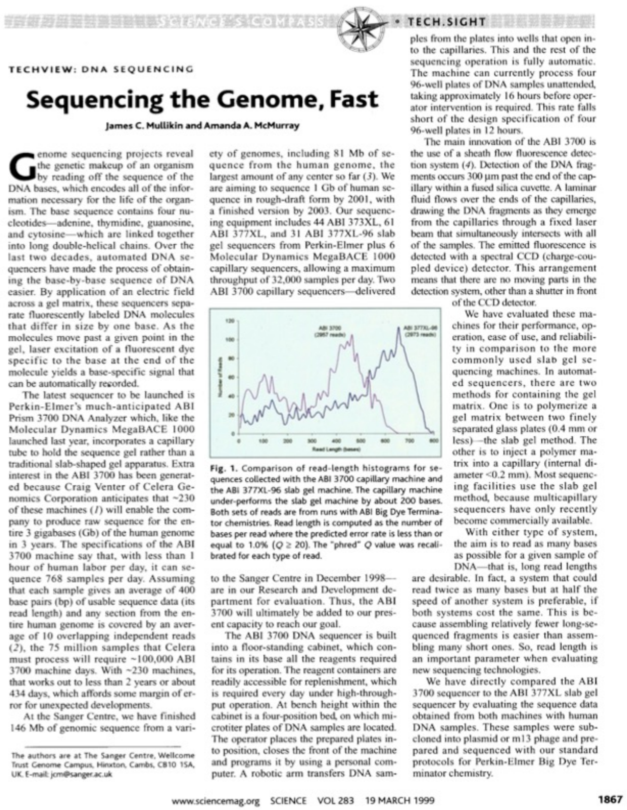




Probabilistic Topic Models, David Blei, 2013

# Dynamic Topic Models

## Original article



SCIENCE'S COMPASS • TECH.SIGHT

TECHVIEW: DNA SEQUENCING

## Sequencing the Genome, Fast

James C. Mullikin and Amanda A. McMurray

Genome sequencing projects reveal the genetic makeup of an organism by reading off the sequence of the DNA bases, which encodes all of the information necessary for the life of the organism. The base sequence contains four nucleotides—adenine, thymidine, guanosine, and cytosine—which are linked together into long double-helical chains. Over the last two decades, automated DNA sequencers have made the process of obtaining the base-by-base sequence of DNA easier. By application of an electric field across a gel matrix, these sequencers separate fluorescently labeled DNA molecules that differ in size by one base. As the molecules move past a given point in the gel, laser excitation of a fluorescent dye specific to the base at the end of the molecule yields a base-specific signal that can be automatically recorded.

The latest sequencer to be launched is Perkin-Elmer's much-anticipated ABI Prism 3700 DNA Analyzer which, like the Molecular Dynamics MegaBACE 1000 launched last year, incorporates a capillary tube to hold the sequence gel rather than a traditional slab-shaped gel apparatus. Extra interest in the ABI 3700 has been generated because Craig Venter of Celera Genomics Corporation anticipates that ~230 of these machines (1) will enable the company to produce raw sequence for the entire 3 gigabases (Gb) of the human genome in 3 years. The specifications of the ABI 3700 machine say that, with less than 1 hour of human labor per day, it can sequence 768 samples per day. Assuming that each sample gives an average of 400 base pairs (bp) of usable sequence data (its read length) and any section from the entire human genome is covered by an average of 10 overlapping independent reads (2), the 75 million samples that Celera must process will require ~100,000 ABI 3700 machine days. With ~230 machines, that works out to less than 2 years or about 434 days, which affords some margin of error for unexpected developments.

At the Sanger Centre, we have finished 146 Mb of genomic sequence from a variety of genomes, including 81 Mb of sequence from the human genome, the largest amount of any center so far (3). We are aiming to sequence 1 Gb of human sequence in rough-draft form by 2001, with a finished version by 2003. Our sequencing equipment includes 44 ABI 373XL, 61 ABI 377XL, and 31 ABI 377XL-96 slab gel sequencers from Perkin-Elmer plus 6 Molecular Dynamics MegaBACE 1000 capillary sequencers, allowing a maximum throughput of 32,000 samples per day. Two ABI 3700 capillary sequencers—delivered to the Sanger Centre in December 1998—are in our Research and Development department for evaluation. Thus, the ABI 3700 will ultimately be added to our present capacity to reach our goal.

The ABI 3700 DNA sequencer is built into a floor-standing cabinet, which contains in its base all the reagents required for its operation. The reagent containers are readily accessible for replenishment, which is required every day under high-throughput operation. At bench height within the cabinet is a four-position bed, on which microtiter plates of DNA samples are located. The operator places the prepared plates into position, closes the front of the machine and programs it by using a personal computer. A robotic arm transfers DNA samples from the plates into wells that open into the capillaries. This and the rest of the sequencing operation is fully automatic. The machine can currently process four 96-well plates of DNA samples unattended, taking approximately 16 hours before operator intervention is required. This rate falls short of the design specification of four 96-well plates in 12 hours.

The main innovation of the ABI 3700 is the use of a sheath flow fluorescence detection system (4). Detection of the DNA fragments occurs 300 μm past the end of the capillary within a fused silica cuvette. A laminar fluid flows over the ends of the capillaries, drawing the DNA fragments as they emerge from the capillaries through a fixed laser beam that simultaneously intersects with all of the samples. The emitted fluorescence is detected with a spectral CCD (charge-coupled device) detector. This arrangement means that there are no moving parts in the detection system, other than a shutter in front of the CCD detector.

We have evaluated these machines for their performance, operation, ease of use, and reliability in comparison to the more commonly used slab gel sequencing machines. In automated sequencers, there are two methods for containing the gel matrix. One is to polymerize a gel matrix between two finely separated glass plates (0.4 mm or less)—the slab gel method. The other is to inject a polymer matrix into a capillary (internal diameter <0.2 mm). Most sequencing facilities use the slab gel method, because multicapillary sequencers have only recently become commercially available.

With either type of system, the aim is to read as many bases as possible for a given sample of DNA—that is, long read lengths are desirable. In fact, a system that could read twice as many bases but at half the speed of another system is preferable, if both systems cost the same. This is because assembling relatively fewer long-sequenced fragments is easier than assembling many short ones. So, read length is an important parameter when evaluating new sequencing technologies.

We have directly compared the ABI 3700 sequencer to the ABI 377XL slab gel sequencer by evaluating the sequence data obtained from both machines with human DNA samples. These samples were subcloned into plasmid or m13 phage and prepared and sequenced with our standard protocols for Perkin-Elmer Big Dye Terminator chemistry.

**Fig. 1.** Comparison of read-length histograms for sequences collected with the ABI 3700 capillary machine and the ABI 377XL-96 slab gel machine. The capillary machine under-performs the slab gel machine by about 200 bases. Both sets of reads are from runs with ABI Big Dye Terminator chemistries. Read length is computed as the number of bases per read where the predicted error rate is less than or equal to 1.0% ($Q \geq 20$). The "phred" $Q$ value was recalibrated for each type of read.

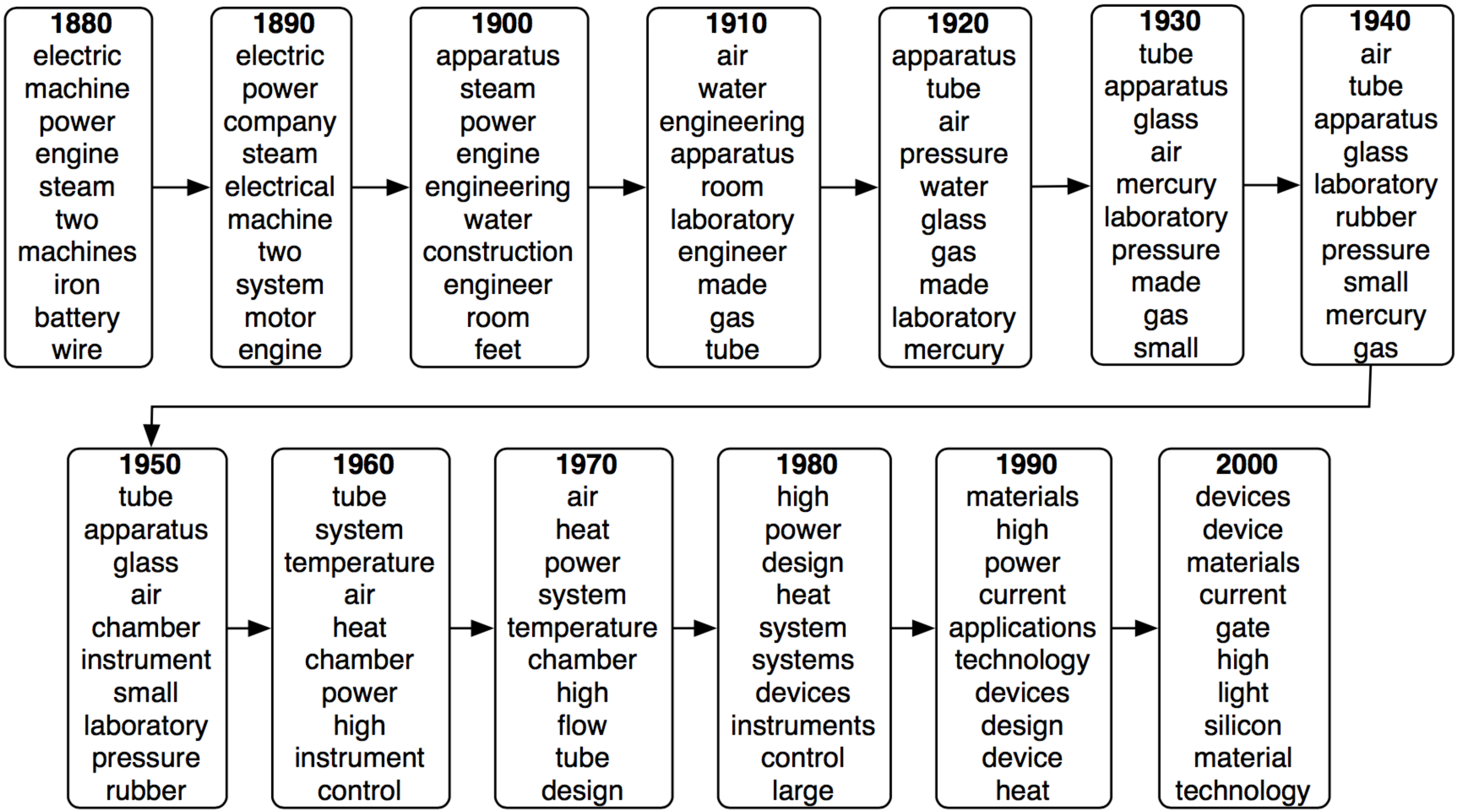
The authors are at The Sanger Centre, Wellcome Trust Genome Campus, Hinxton, Cambs, CB10 1SA, UK. E-mail: jcm@sanger.ac.uk

www.sciencemag.org SCIENCE VOL 283 19 MARCH 1999 1867

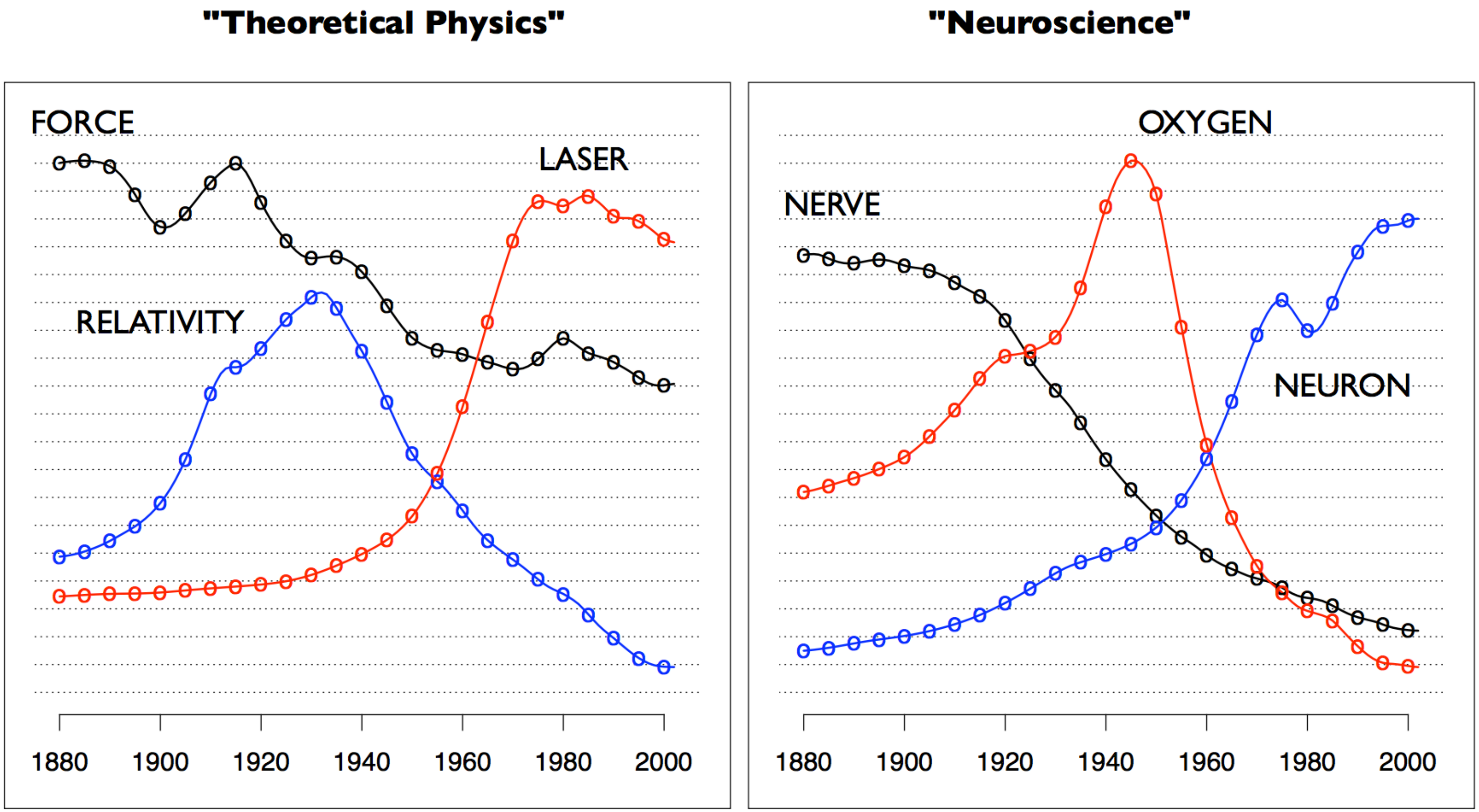
## Most likely words from top topics

| | | |
|---|---|---|
| sequence | devices | data |
| genome | device | information |
| genes | materials | network |
| sequences | current | web |
| human | high | computer |
| gene | gate | language |
| dna | light | networks |
| sequencing | silicon | time |
| chromosome | material | software |
| regions | technology | system |
| analysis | electrical | words |
| data | fiber | algorithm |
| genomic | power | number |
| number | based | internet |

# Dynamic Topic Models



# Dynamic Topic Models




David M. Blei and John D. Lafferty. Dynamic topic models.
In Proc. ICML '06. pp. 113-120. **2006**.

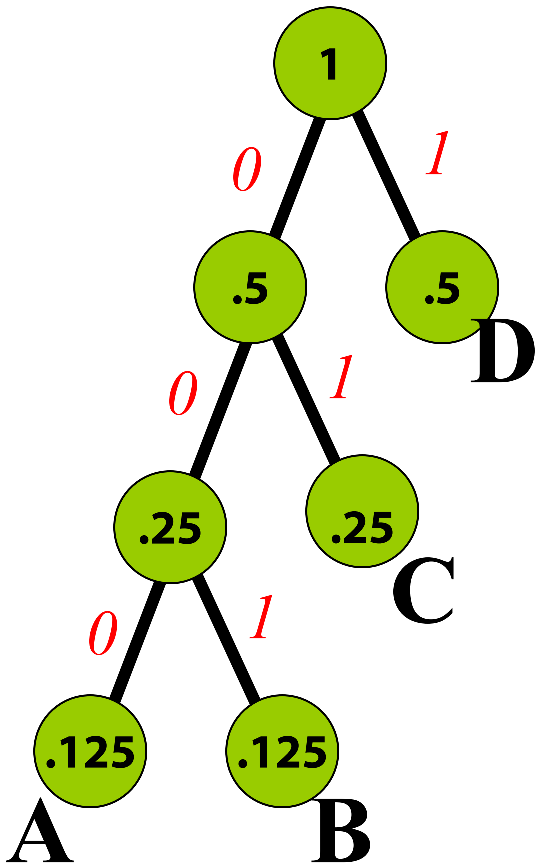
# Dynamic Topic Models

- Understand developments
- Distributions of topics over time
- Discretization of time might be a problem
  - Runtime increases dramatically
  - Continuous dynamic topic models

- Many applications
  - E.g., comparison of science areas, analysis of scientific work
- How can we compare distributions?

Wang, Chong; Blei, David; Heckerman, David. "Continuous Time Dynamic Topic Models". *Proceedings of ICML*'08, **2008**.

# Recap: Huffman code example




| M | code | length | prob | Exp. len |
|---|---|---|---|---|
| A | 000 | 3 | 0,125 | 0,375 |
| B | 001 | 3 | 0,125 | 0,375 |
| C | 01 | 2 | 0,250 | 0,500 |
| D | 1 | 1 | 0,500 | 0,500 |
| average message length | | | | 1,750 |

If we need to send many messages (A,B,C or D) and they have this probability distribution and we use this code, then over time, the average bits/message should approach 1.75

# Recap: Information Theory Background

- Assume that you need to send messages from a repertoire of $n$ messages
- If there are $n$ equally probable possible messages, then the probability $p$ of each is $1/n$ or $n = 1/p$
- Information (number of bits) conveyed by a message is $\log(n) = \log(1/p) = -\log(p)$
- E.g., if there are 16 messages, then $\log(16) = 4$ and we need 4 bits to identify/send each message.
- In general, if we are given a probability distribution

  $P = (p_1, p_2, .., p_n)$
- Expected information induced by distribution P (aka entropy of P):

  $I(P) = -(p_1 * \log(p_1) + p_2 * \log(p_2) + .. + p_n * \log(p_n))$

  $= - \sum_i p_i * \log(p_i) = \sum_i p_i * \log(1/p_i)$
- What if one used an erroneous distribution $q$?
  - One might use too many bits for more frequent messages

# The KL Divergence

- The *cross-entropy*, or *Kullback-Leibler divergence*, between two distributions $\mathbf{p}$ and $\mathbf{q}$ measures the *expected information gain* (reduction in average number of bits per event) due to replacing the "wrong" distribution $\mathbf{q}$ with the "right" distribution $\mathbf{p}$:

$$D^{KL}(\mathbf{p}, \mathbf{q}) \equiv \sum_i p_i(\ln(1/q_i) - \ln(1/p_i)) = \mathbf{E_p}[\ln(\mathbf{p}/\mathbf{q})]$$

- Not symmetric

# Hellinger Distance

- The *Hellinger distance* is a symmetric measure of distance between two distributions that is popular in machine learning applications:

$$D^{HEL}(\mathbf{p}, \mathbf{q}) \equiv \|\sqrt{\mathbf{p}} - \sqrt{\mathbf{q}}\|_2 = \left( \sum_{j=1}^{n} \left( \sqrt{p_j} - \sqrt{q_j} \right)^2 \right)^{1/2}$$

$$\in [0, \sqrt{2}]$$

- Sometimes value should be in [0, 1]

For two discrete probability distributions $P = (p_1, \ldots, p_k)$ and $Q = (q_1, \ldots, q_k)$, their Hellinger distance is defined as

$$H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2},$$

[Wikipedia]

# Dynamic Topic Models

- **Time-corrected similarity** shows a new way of using the posterior
- Consider the expected Hellinger distance between the topic proportions of two documents,

$$d_{ij} = \mathrm{E}\left[\sum_{k=1}^{K}(\sqrt{\theta_{i,k}} - \sqrt{\theta_{j,k}})^2 \,|\, \mathbf{w}_i, \mathbf{w}_j\right]$$

- Uses the latent structure to define similarity
- Time has been factored out because the topics associated to the components are different from year to year
- Similarity of documents based only on topic proportions

# Dynamic Topic Models

## The Brain of the Orang (1880)



*Tribune* in these cases, which were submitted to the authors on the 4th of December last for correction or rejection; no objection being made we printed them in a recent number. After publication Professor Agassiz now writes that the reports under his name are not satisfactory to him. We therefore request our readers to consider them withdrawn.

Professor George F. Barker, Professor O. C. Marsh and Professor J. E. Hilgard are preparing more elaborate reports of their important papers, and promise them at an early day.

THE BRAIN OF THE ORANG.*

BY HENRY C. CHAPMAN, M.D.

The brain of the Orang has been figured by Tiedemann, Sandifort, Schroeder van der Kolk and Vrolik, Gratiolet, Rolleston, etc. On account, however, of the few illustrations extant, and of the importance of the subject, I avail myself of the opportunity of presenting several views of my Orang's brain (Figs. 1 to 5), which was removed from the skull only a few hours after death. The membranes were in a high state of congestion, and a little of the surface of the left hemisphere had been disorganized by disease, otherwise the brain was in good condition. It weighed exactly ten ounces. The brain of the Orang in its general contour resembled that of man more than those of either of the Chimpanzees which I examined. In these the brain was more elongated. The general character of the folds and fissures in

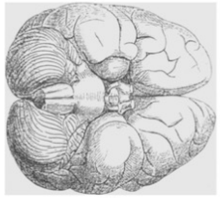


Fig. 1.

the brain of the Orang, Chimpanzee, and man are the same; there are certain minor differences, however, in their disposition in all three. The fissure of Sylvius in the Orang runs up and down the posterior branch pursuing only a slightly backward direction; the anterior branch is small. The fissure of Rolando, or central fissure, quite apparent, is, however, situated slightly more forward in the Orang than in man. It differentiates the frontal from the parietal lobe. The parieto-occipital fispure is well marked; bordered externally by the first occipital fold it descends internally on the mesial side of the hemisphere, separating the parietal from the occipal lobes.

in the Orang, the parieto-occipital fissure does not reach the calcarine, being separated from it by the "deuxième plis de passage interne" of Gratiolet, or "untere innere Scheitelbogen-Windung" of Bischoff. I have noticed this separation as an anomaly more than once in man.

According to Bischoff, this disposition obtains in the Gorilla, and seems to be usual also in the Chimpanzee. In the female Chimpanzee, however, on the left side I found the parieto-occipital fissure passing into the calcarine, as in man. The frontal lobe is easily distinguished from the parietal by the fissure of Rolando, and from the temporal by the fissure of Sylvius. In the Orang it is higher, wider, and more arched than in the Chimpanzee. The anterior central convolution in front of the central fissure runs into the post-central convolution above and below, as in man. It is difficult, however, to identify the three frontal convolutions seen in man and the Chimpanzee, the frontal lobe of the Orang dividing rather into two convolutions, the middle one being badly defined. This is due somewhat to the length of the precentral fissure, which is as long as the fissure of Rolando, extending farther upward than in man. There was nothing particularly noticeable about the base of the frontal lobe; on the mesial surface it ran into the parietal. The part above the calloso-marginal fissure in the Orang is not as distinctly divided into convolutions as in man, though these are not constantly present even in all human brains. The parietal lobe is separated from the frontal by the central fissure, from the occipital and temporal incompletely, by the parieto-occipital and Sylvian fissures. The posterior-central convolution is well defined. The parietal fissure in the Orang is more striking than that of man, resembling the Gorilla's; it is twice as long as the corresponding fissure in the Chimpanzee, extending from the transverse occipital fissure, as is sometimes the case in man, almost into the fissure of Rolando. It is unbridged and without a break, and divides the parietal lobe completely into upper and lower parietal lobules. The upper parietal lobule is bounded externally by the parietal fissure; posteriorly it is separated from the occipital lobe, internally by the parieto-

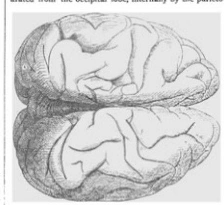
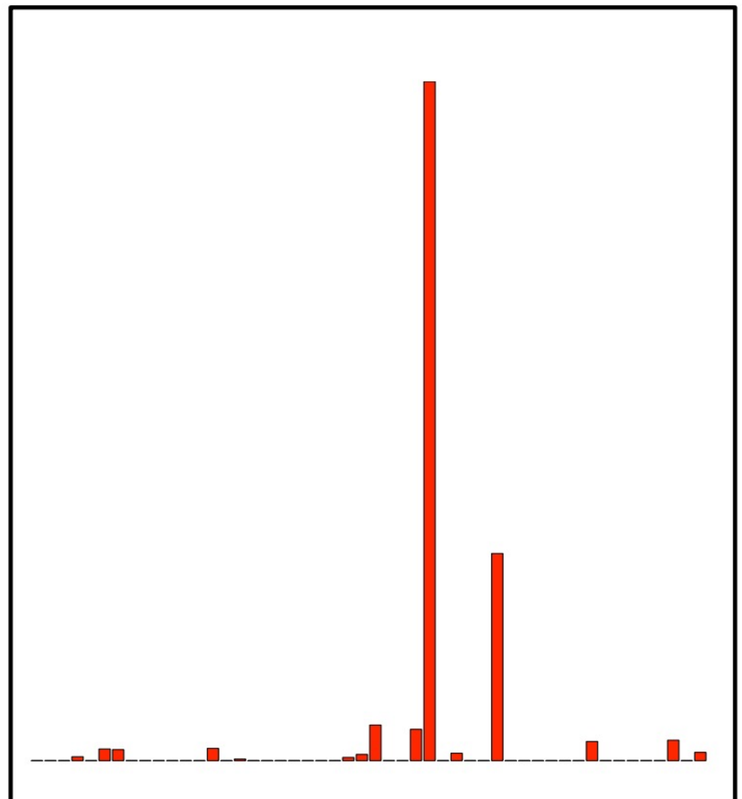


Fig. 2.

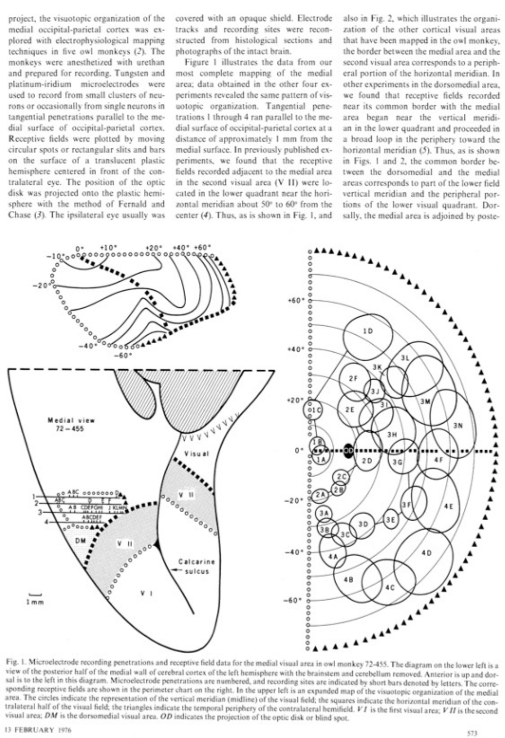
occipital fissure; externally it is continuous with the occipital lobe, as the first occipital gyrus, anteriorly it is separated from the posterior central convolution more completely than in man, by a fissure which runs paralled with the central fissure. There is in the Orang, also, a fissure running paralled with the parietal, which subdivides the upper parietal lobule into inner and outer portions. The precuneus, or the space on the mesial side of the parietal lobe between the parieto-occipital

* From the Proceedings of the Academy of Natural Sciences, Phila., 1880.



# Dynamic Topic Models

**Representation of the Visual Field on the Medial Wall of Occipital-Parietal Cortex in the Owl Monkey (1976)**



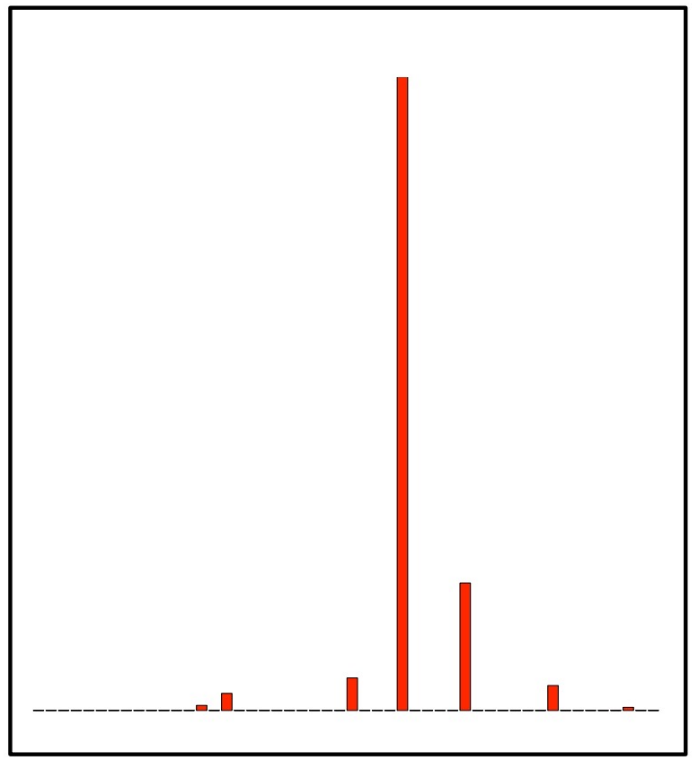
project, the visuotopic organization of the medial occipital-parietal cortex was explored with electrophysiological mapping techniques in five owl monkeys (*2*). The monkeys were anesthetized with urethan and prepared for recording. Tungsten and platinum-iridium microelectrodes were used to record from small clusters of neurons or occasionally from single neurons in tangential penetrations parallel to the medial surface of occipital-parietal cortex. Receptive fields were plotted by moving circular spots or rectangular slits and bars on the surface of a translucent plastic hemisphere centered in front of the contralateral eye. The position of the optic disk was projected onto the plastic hemisphere with the method of Fernald and Chase (*3*). The ipsilateral eye usually was covered with an opaque shield. Electrode tracks and recording sites were reconstructed from histological sections and photographs of the intact brain.

Figure 1 illustrates the data from our most complete mapping of the medial area; data obtained in the other four experiments revealed the same pattern of visuotopic organization. Tangential penetrations 1 through 4 ran parallel to the medial surface of occipital-parietal cortex at a distance of approximately 1 mm from the medial surface. In previously published experiments, we found that the receptive fields recorded adjacent to the medial area in the second visual area (V II) were located in the lower quadrant near the horizontal meridian about 50° to 60° from the center (*4*). Thus, as is shown in Fig. 1, and also in Fig. 2, which illustrates the organization of the other cortical visual areas that have been mapped in the owl monkey, the border between the medial area and the second visual area corresponds to a peripheral portion of the horizontal meridian. In other experiments in the dorsomedial area, we found that receptive fields recorded near its common border with the medial area began near the vertical meridian in the lower quadrant and proceeded in a broad loop in the periphery toward the horizontal meridian (*5*). Thus, as is shown in Figs. 1 and 2, the common border between the dorsomedial and the medial areas corresponds to part of the lower field vertical meridian and the peripheral portions of the lower visual quadrant. Dorsally, the medial area is adjoined by poste-



Fig. 1. Microelectrode recording penetrations and receptive field data for the medial visual area in owl monkey 72-455. The diagram on the lower left is a view of the posterior half of the medial wall of cerebral cortex of the left hemisphere with the brainstem and cerebellum removed. Anterior is up and dorsal is to the left in this diagram. Microelectrode penetrations are numbered, and recording sites are indicated by short bars denoted by letters. The corresponding receptive fields are shown in the perimeter chart on the right. In the upper left is an expanded map of the visuotopic organization of the medial area. The circles indicate the representation of the vertical meridian (midline) of the visual field; the squares indicate the horizontal meridian of the contralateral half of the visual field; the triangles indicate the temporal periphery of the contralateral hemifield. *V I* is the first visual area; *V II* is the second visual area; *DM* is the dorsomedial visual area. *OD* indicates the projection of the optic disk or blind spot.

13 FEBRUARY 1976 573



# Dynamic Topic Models: Summary

- Can model changes of topics (= word distributions) in corpora over time
- Uses a technique for modeling temporal influences
- As a by-product we have discussed techniques for comparing distributions

# Word-Word Associations in Document Retrieval

Recap

- LSI: Documents as vectors, dimension reduction
- Topic Modeling
  - Topic = Word distribution
  - From LDA-Model: $P(Z \mid \mathbf{w})$
  - Assumption: Bag of words model (independence, naïve Bayes, unigram distribution)

Words are not independent of each other

- Word similarity measures
- Extend query with similar words automatically
- Extend query with most frequent followers/predecessors
- Insert words in anticipated gaps in a string query

Need to represent more aspects of word semantics

# Approaches for Representing Word Semantics

Beyond bags of words

**Distributional Semantics (*Count*)**

- Used since the 90's
- Sparse word-context PMI/PPMI matrix
- Decomposed with SVD

**Word Embeddings (*Predict*)**

- Inspired by deep learning
- `word2vec` *(Mikolov et al., 2013)*
- GloVe *(Pennington et al., 2014)*

Underlying Theory: **The Distributional Hypothesis** *(Harris, '54; Firth, '57)*

"Similar words occur in similar contexts"

https://www.tensorflow.org/tutorials/word2vec

https://nlp.stanford.edu/projects/glove/

# Point(wise) Mutual Information: PMI

- Measure of association used in information theory and statistics

$$\operatorname{pmi}(x;y) \equiv \log\frac{p(x,y)}{p(x)p(y)} = \log\frac{p(x|y)}{p(x)} = \log\frac{p(y|x)}{p(y)}$$

- Positive PMI: PPMI(x, y) = max( pmi(x, y), 0 )
- Quantifies the discrepancy between the probability of their coincidence given their joint distribution and their individual distributions, assuming independence
- Finding collocations and associations between words
- Countings of occurrences and co-occurrences of words in a text corpus can be used to approximate the probabilities p(x) or p(y) and p(x,y) respectively

Kenneth Ward Church and Patrick Hanks. "Word association norms, mutual information, and lexicography". Comput. Linguist. 16 (1): 22–29. **1990**.

# PMI – Example

| word 1 | word 2 | count word 1 | count word 2 | count of co-occurrences | PMI |
|---|---|---|---|---|---|
| puerto | rico | 1938 | 1311 | 1159 | 10.0349081703 |
| hong | kong | 2438 | 2694 | 2205 | 9.72831972408 |
| los | angeles | 3501 | 2808 | 2791 | 9.56067615065 |
| carbon | dioxide | 4265 | 1353 | 1032 | 9.09852946116 |
| prize | laureate | 5131 | 1676 | 1210 | 8.85870710982 |
| san | francisco | 5237 | 2477 | 1779 | 8.83305176711 |
| nobel | prize | 4098 | 5131 | 2498 | 8.68948811416 |
| ice | hockey | 5607 | 3002 | 1933 | 8.6555759741 |
| star | trek | 8264 | 1594 | 1489 | 8.63974676575 |
| car | driver | 5578 | 2749 | 1384 | 8.41470768304 |
| it | the | 283891 | 3293296 | 3347 | -1.72037278119 |
| are | of | 234458 | 1761436 | 1019 | -2.09254205335 |
| this | the | 199882 | 3293296 | 1211 | -2.38612756961 |
| is | of | 565679 | 1761436 | 1562 | -2.54614706831 |
| and | of | 1375396 | 1761436 | 2949 | -2.79911817902 |
| a | and | 984442 | 1375396 | 1457 | -2.92239510038 |
| in | and | 1187652 | 1375396 | 1537 | -3.05660070757 |
| to | and | 1025659 | 1375396 | 1286 | -3.08825363041 |
| to | in | 1025659 | 1187652 | 1066 | -3.12911348956 |
| of | and | 1761436 | 1375396 | 1190 | -3.70663100173 |

- Counts of pairs of words getting the most and the least PMI scores in the first 50 millions of words in Wikipedia (dump of October 2015)
- Filtering by 1,000 or more co-occurrences.
- The frequency of each count can be obtained by dividing its value by 50,000,952. (Note: natural log is used to calculate the PMI values in this example, instead of log base 2)

[Wikipedia]

# The Contributions of Word Embeddings

**Novel Algorithms**

*(objective + training method)*

- Skip Grams + Negative Sampling
- CBOW + Hierarchical Softmax
- Noise Contrastive Estimation
- GloVe
- …

**New Hyperparameters**

*(preprocessing, smoothing, etc.)*

- Subsampling of Frequent Words
- Dynamic Context Windows
- Context Distribution Smoothing
- Adding Context Vectors
- …

What's really improving performance?

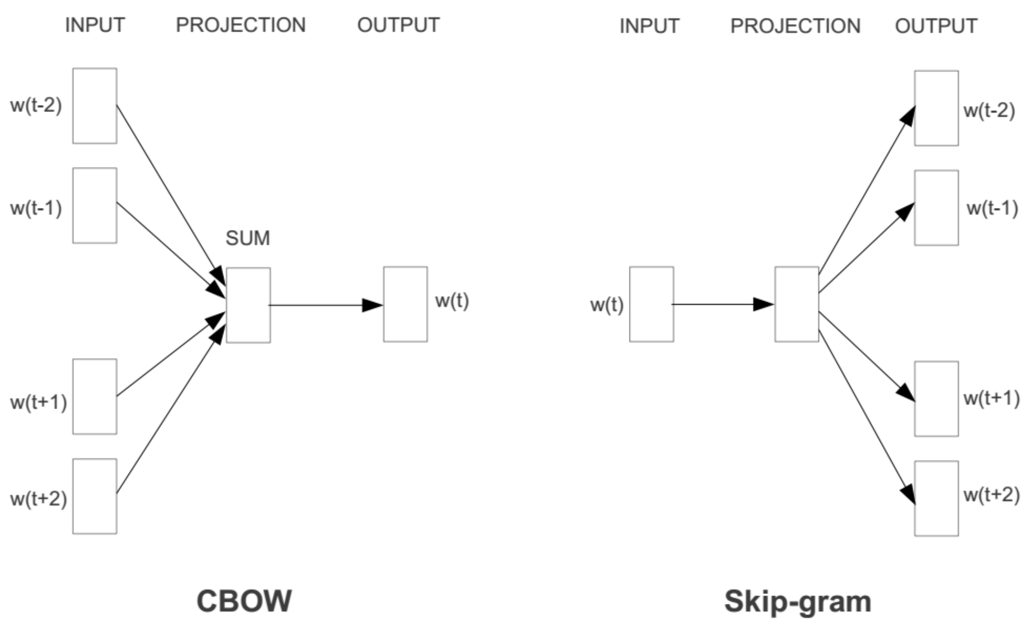
Improving Distributional Similarity with Lessons Learned from Word Embeddings, Omer Levy, Yoav Goldberg, Ido Dagan, Transactions of the Association for Computational Linguistics, Volume 3, **2015**.

# Embedding Approaches

- Represent each word with a low-dimensional vector
- Word similarity = vector similarity
- Key idea: Predict surrounding words of every word
- Faster and can easily incorporate a new sentence/document or add a word to the vocabulary

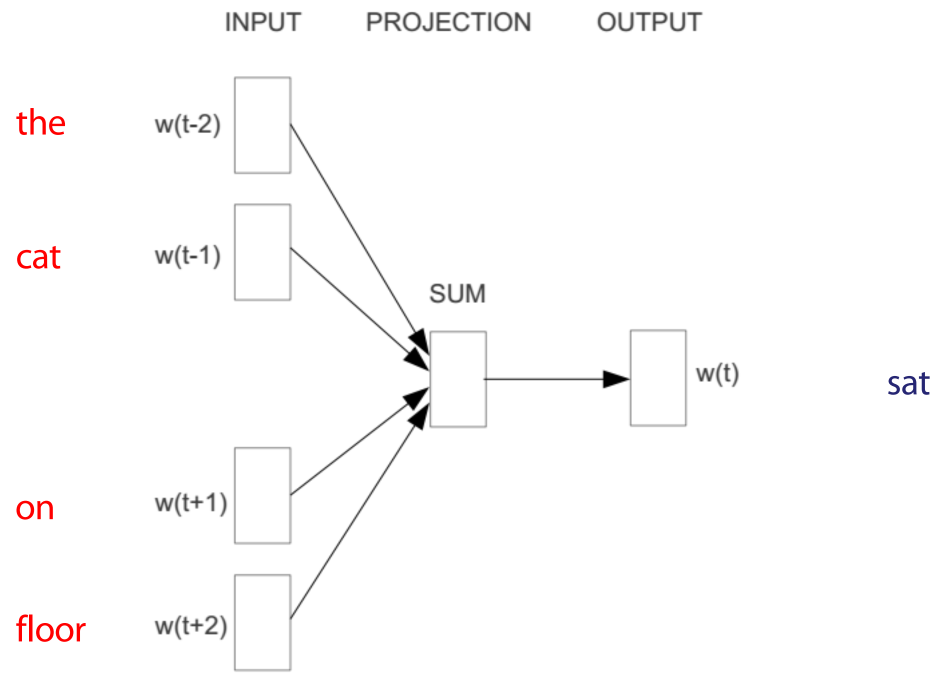
# **Represent** the meaning of a **word** – word2vec

- 2 basic network models:
  - Continuous Bag of Word (CBOW): use a window to predict the middle word
  - Skip-gram (SG): use a word to predict the surrounding ones in window.

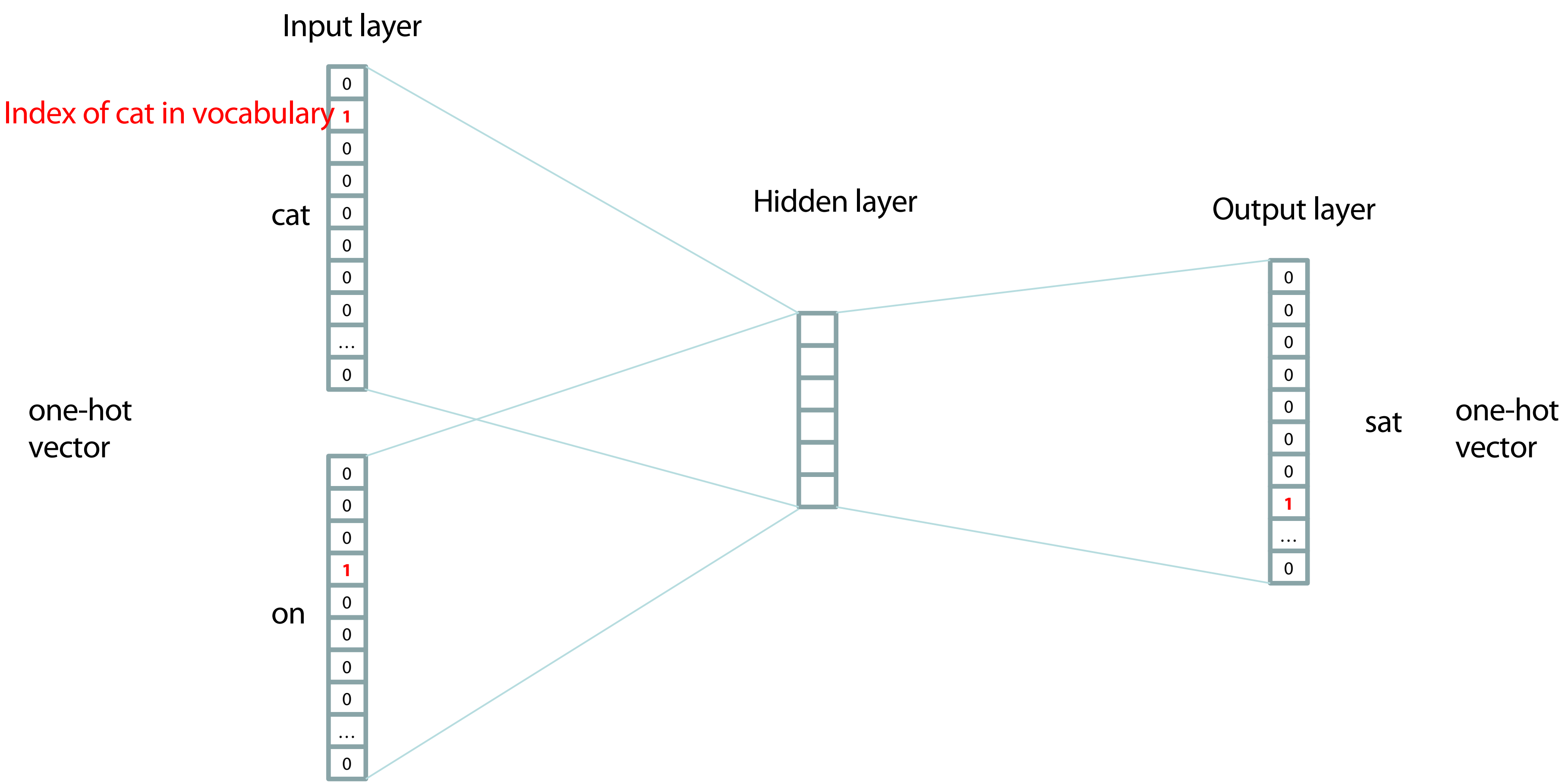


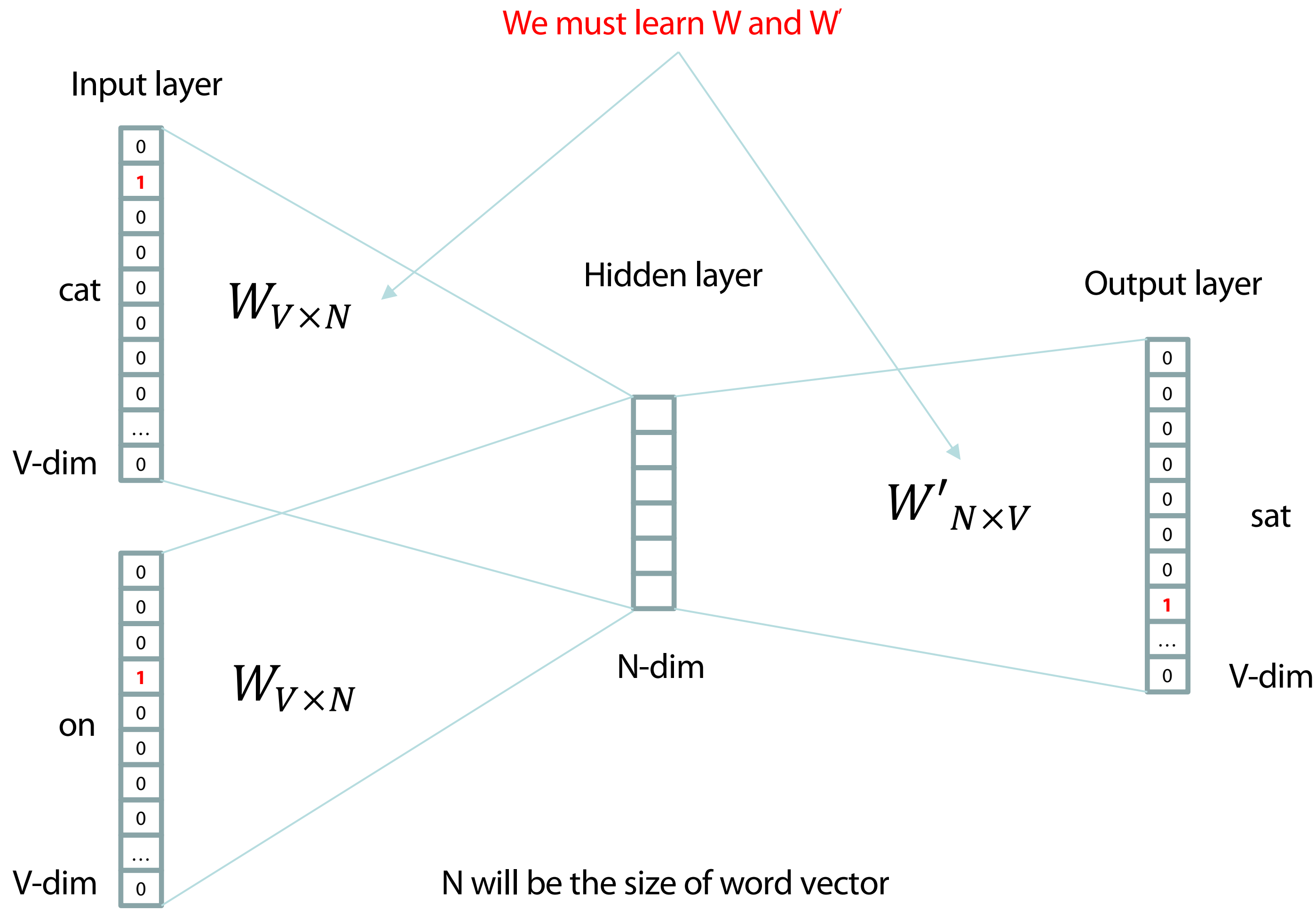
# Word2vec – Continuous Bag of Words

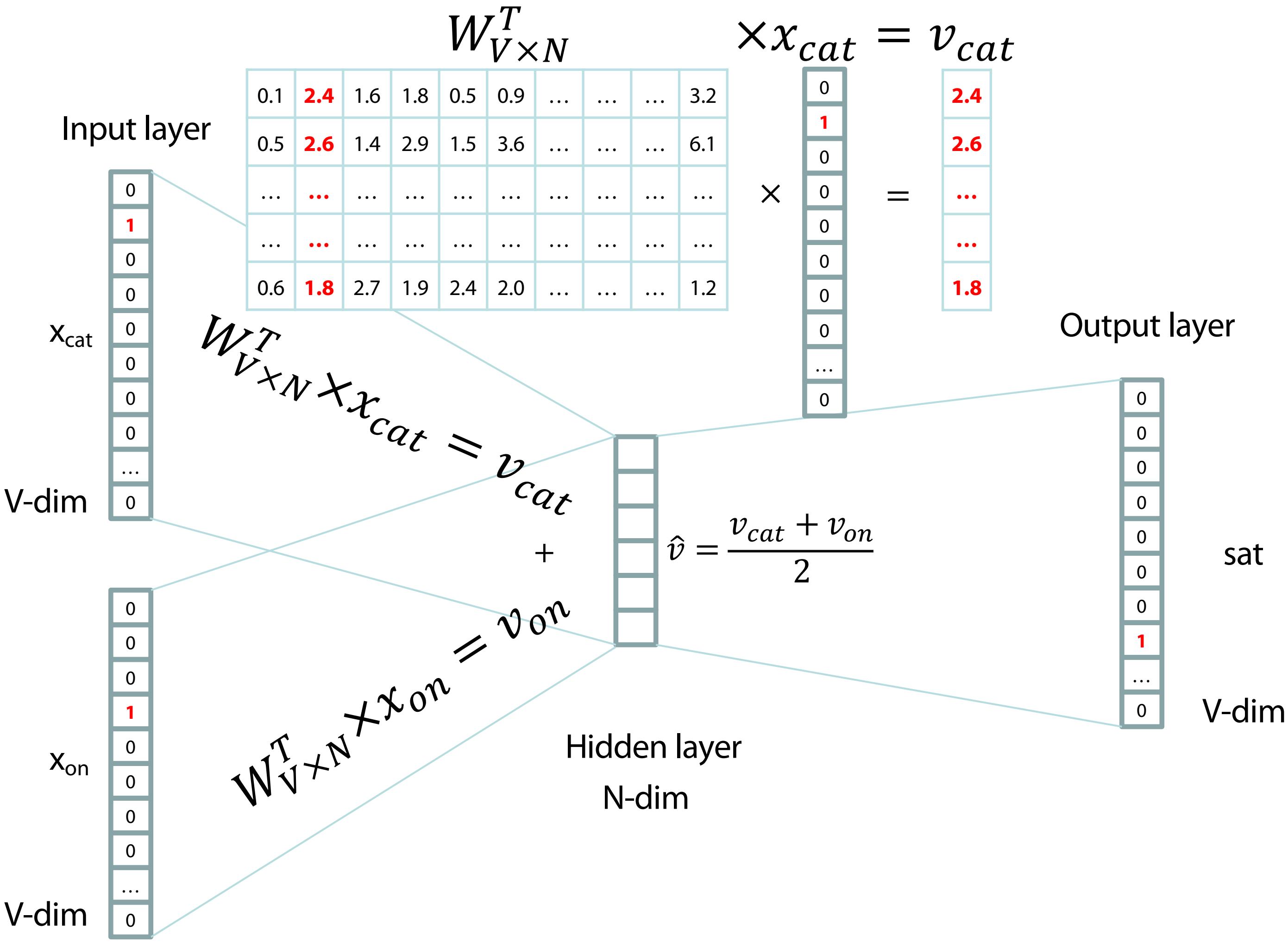
- E.g. "The cat sat on floor"
  - Window size = 2




Distributed Representations of Words and Phrases and their Compositionality
Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean, NIPS **2013**


Input layer
Index of cat in vocabulary
cat
0
1
0
0
0
0
0
0
...
0
one-hot
vector
0
0
0
1
0
0
0
0
...
0
on
Hidden layer
Output layer
0
0
0
0
0
0
0
1
...
0
sat
one-hot
vector


We must learn W and W′
Input layer
cat
V-dim
on
V-dim
$W_{V\times N}$
$W_{V\times N}$
Hidden layer
N-dim
$W'_{N\times V}$
Output layer
sat
V-dim
N will be the size of word vector




$$W^T_{V\times N} \times x_{cat} = v_{cat}$$

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.1 | **2.4** | 1.6 | 1.8 | 0.5 | 0.9 | … | … | … | 3.2 |
| 0.5 | **2.6** | 1.4 | 2.9 | 1.5 | 3.6 | … | … | … | 6.1 |
| … | **…** | … | … | … | … | … | … | … | … |
| … | **…** | … | … | … | … | … | … | … | … |
| 0.6 | **1.8** | 2.7 | 1.9 | 2.4 | 2.0 | … | … | … | 1.2 |

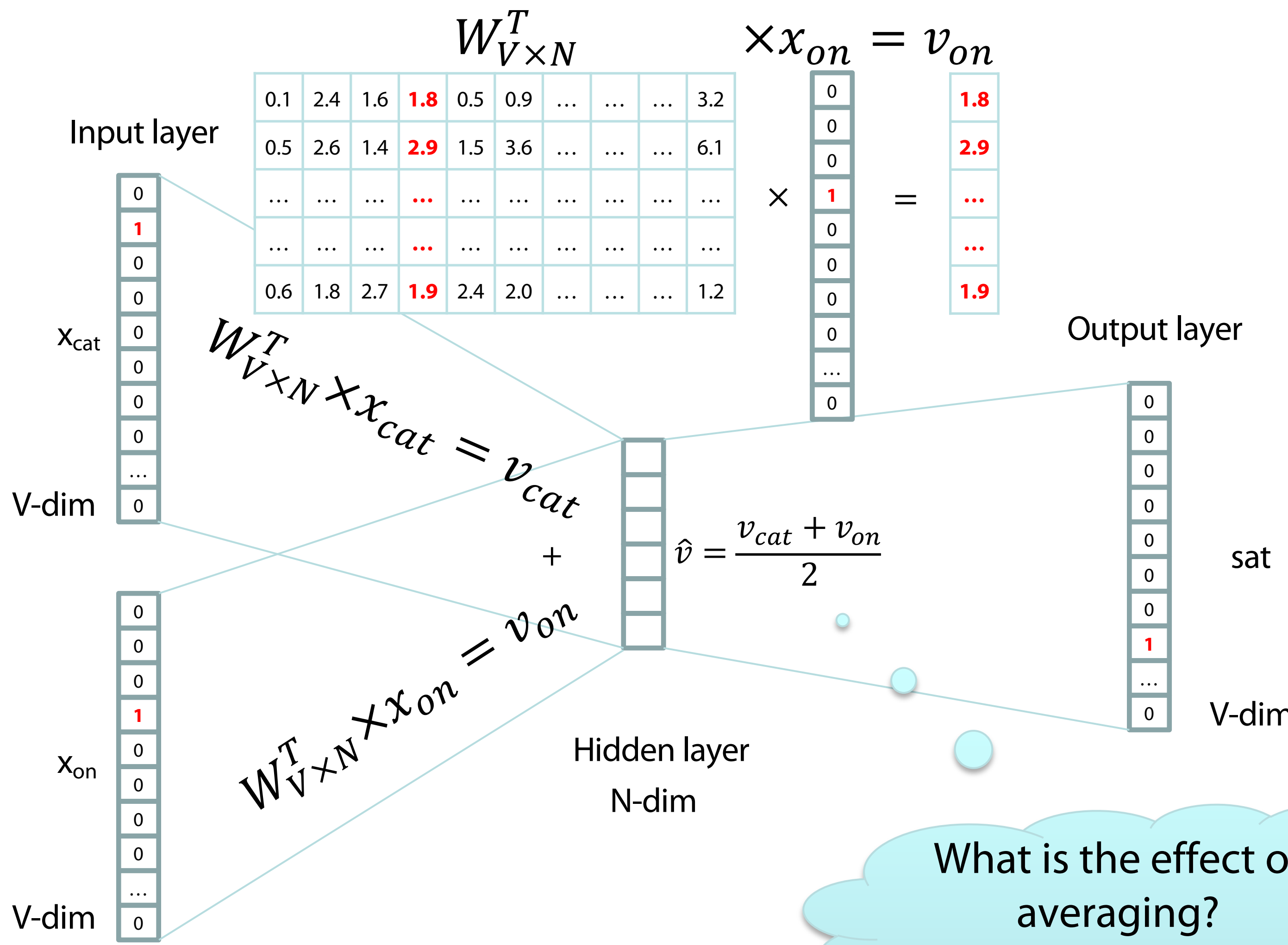
$$\times \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \dots \\ 0 \end{pmatrix} = \begin{pmatrix} 2.4 \\ 2.6 \\ \dots \\ \dots \\ 1.8 \end{pmatrix}$$

$$W^T_{V\times N} \times x_{cat} = v_{cat}$$

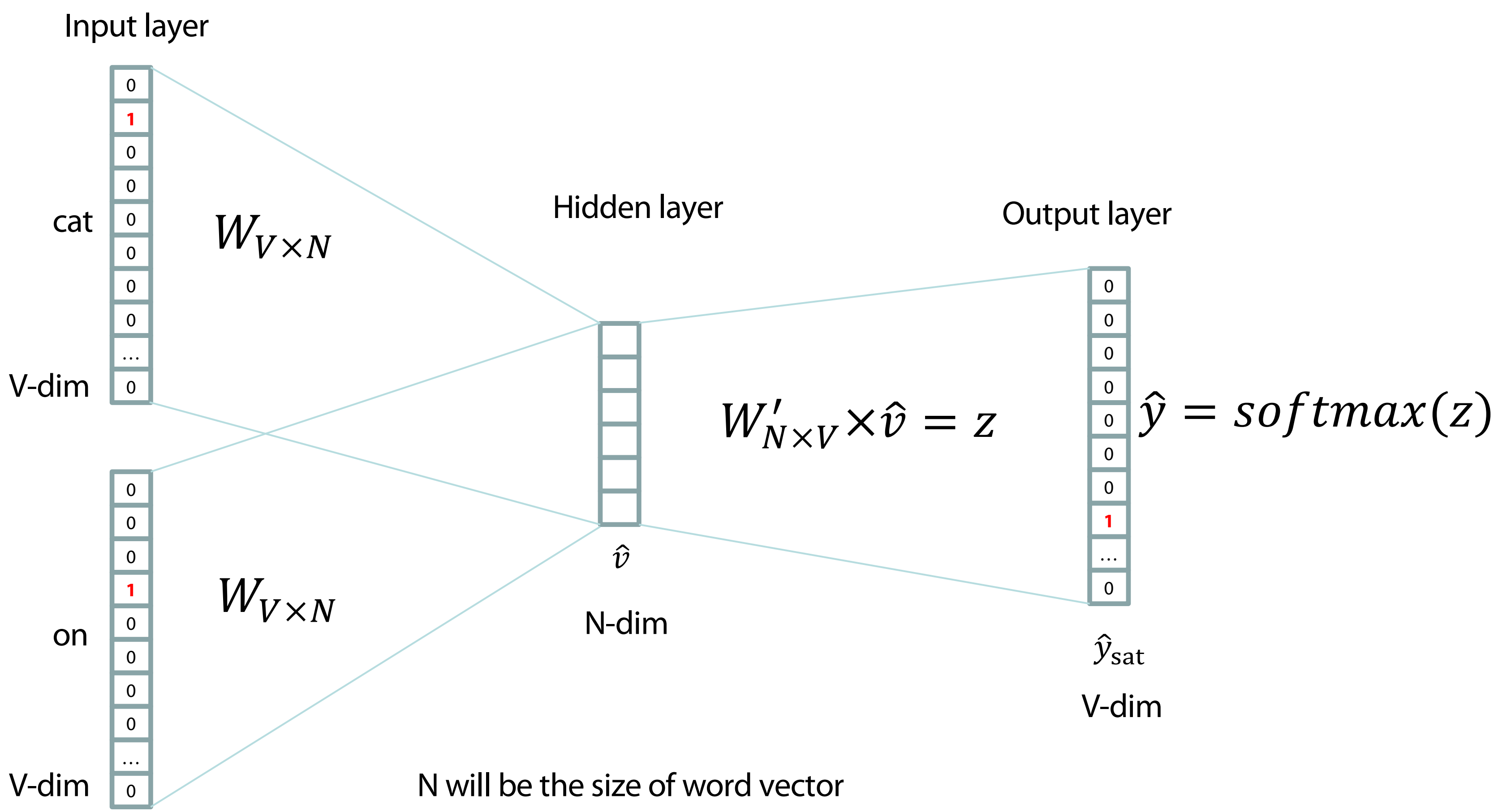
$$+$$

$$W^T_{V\times N} \times x_{on} = v_{on}$$

$$\hat{v} = \frac{v_{cat} + v_{on}}{2}$$




What is the effect of averaging? Concatenation?




N will be the size of word vector

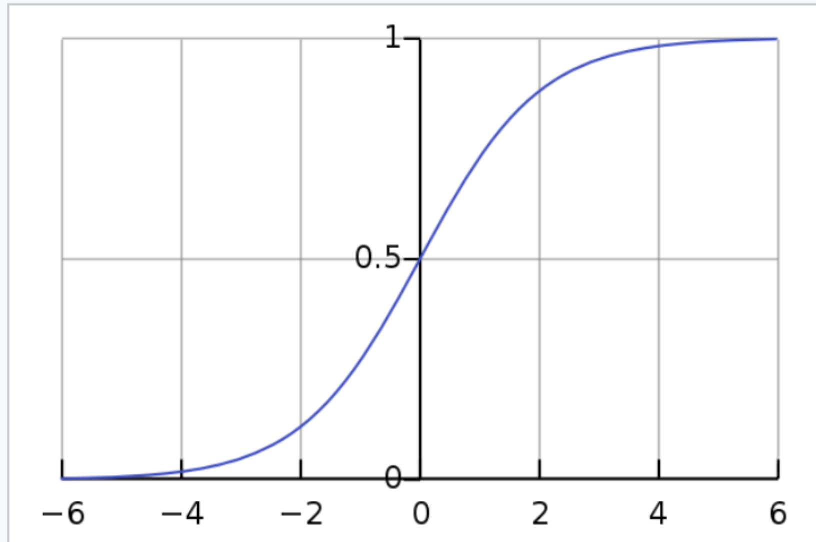
# Logistic function

A **logistic function** or **logistic curve** is a common "S" shape (sigmoid curve), with equation:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

where

- $e$ = the natural logarithm base (also known as Euler's number),
- $x_0$ = the $x$-value of the sigmoid's midpoint,
- $L$ = the curve's maximum value, and
- $k$ = the steepness of the curve.[1]




Standard logistic sigmoid function i.e. $L = 1, k = 1, x_0 = 0$

[Wikipedia]

# softmax($\mathbf{z}$)

The **softmax function**, or **normalized exponential function**, is a generalization of the logistic function that "squashes" a $K$-dimensional vector $\mathbf{z}$ of arbitrary real values to a $K$-dimensional vector $\sigma(\mathbf{z})$ of real values in the range [0, 1] that add up to 1. The function is given by

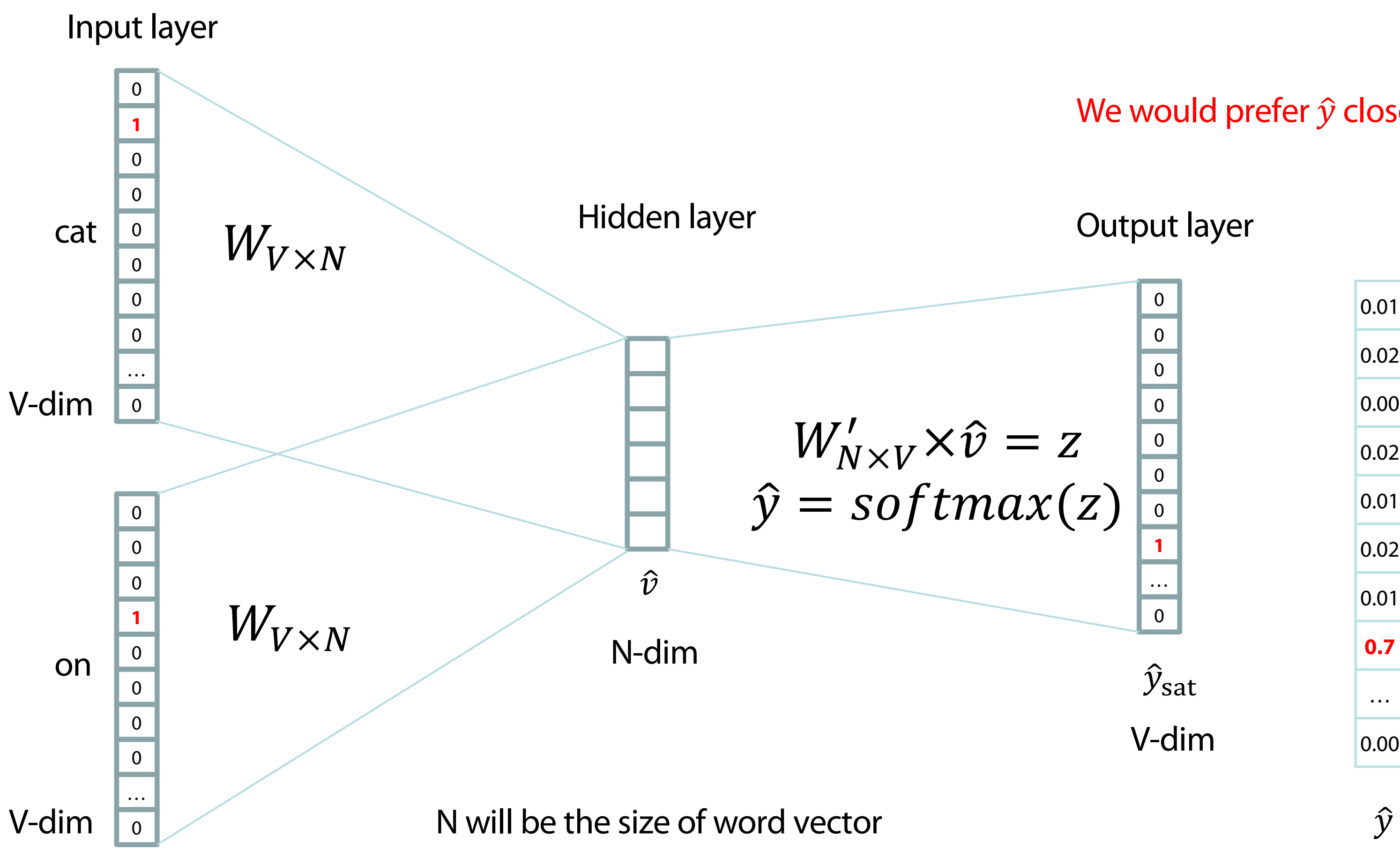
$$\sigma : \mathbb{R}^K \to [0,1]^K$$

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

In probability theory, the output of the softmax function can be used to represent a categorical distribution – that is, a probability distribution over $K$ different possible outcomes.

[Wikipedia]

We would prefer $\hat{y}$ close to $\hat{y}_{sat}$




N will be the size of word vector

# CBOW Model

Objective: Given $w_{c-k}$ … , $w_{c-1}, w_{c+1}$, … , $w_{c+k}$, predict $w_c$

Training data: Given sequence of words $< w_1, w_2, \ldots w_n >$, extract context and target: $(w_{c-k} \ldots, w_{c-1}, w_{c+1}, \ldots, w_{c+k}; w_c)$

Knowns:

- Training data $\{(w_{c-k} \ldots, w_{c-1}, w_{c+1}, \ldots, w_{c+k}; w_c)\}$
- Vocabulary $\{w_1, w_2, \ldots w_V\}$ of the training corpus

Unknowns:

- Word embedding matrices $W_{V \times N}$ and $W'_{N \times V}$ with $N$ being a hyperparameter

# Loss Function for Learning

- How to determine word embedding matrices?
- Cross entropy for comparing probability distributions
  - $H(\hat{y}, y) = -\sum_{j=1}^{V} y_j \, log(\hat{y}_j)$
- $y$ is a one-hot vector with a "one" at position $c$
  - $H(\hat{y}, y) = -y_c \, log(\hat{y}_c) = -log(\hat{y}_c)$

In this formulation, $c$ is the index where the correct word's one hot vector is 1. We can now consider the case where our prediction was perfect and thus $\hat{y}_c = 1$. We can then calculate $H(\hat{y}, y) = -1\log(1) = 0$. Thus, for a perfect prediction, we face no penalty or loss. Now let us consider the opposite case where our prediction was very bad and thus $\hat{y}_c = 0.01$. As before, we can calculate our loss to be $H(\hat{y}, y) = -1\log(0.01) \approx 4.605$. We can thus see that for probability distributions, cross entropy provides us with a good measure of distance.

# CBOW: Derivation of Learning Procedure

$Minimize \quad -\log P(w_c | w_{c-k}, \ldots, w_{c-1}, w_{c+1}, \ldots, w_{c+k})$

$= -\log P(W'[c] \mid \hat{v})$ (and due to the softmax)

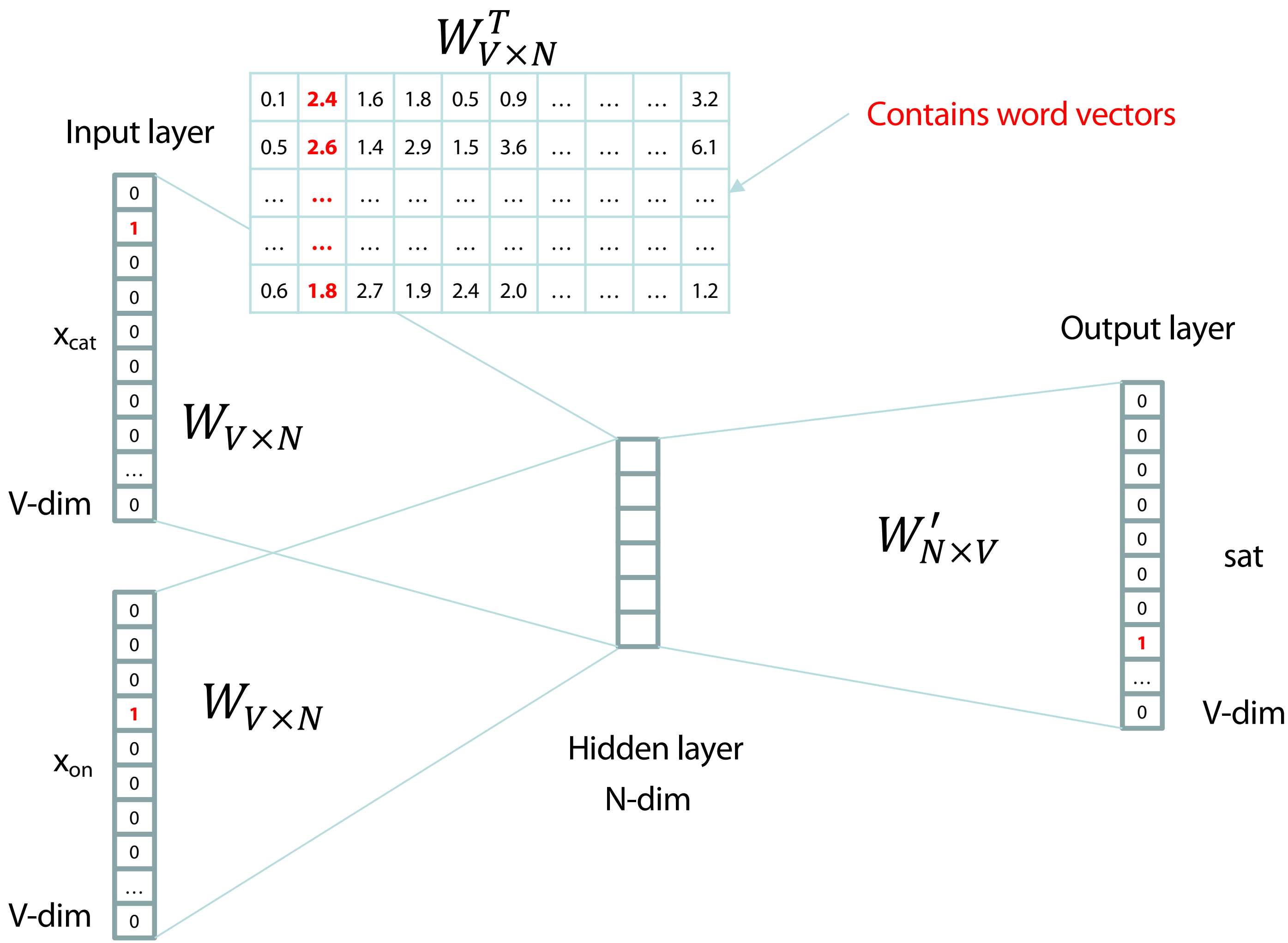
$= -\log \frac{e^{W'[c]^T \hat{v}}}{\sum_{j=1}^{V} e^{W'[c]^T{}_j \hat{v}}}$

$= -W'[c]^T \hat{v} + \log \sum_{j=1}^{V} e^{W'[j]^T \hat{v}}$

where

$\hat{v} = (2k)^{-1} \sum_{i=-k}^{k} W^T w_{c+i}$

Use gradient descent to update word vectors




| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.1 | **2.4** | 1.6 | 1.8 | 0.5 | 0.9 | … | … | … | 3.2 |
| 0.5 | **2.6** | 1.4 | 2.9 | 1.5 | 3.6 | … | … | … | 6.1 |
| … | **…** | … | … | … | … | … | … | … | … |
| … | **…** | … | … | … | … | … | … | … | … |
| 0.6 | **1.8** | 2.7 | 1.9 | 2.4 | 2.0 | … | … | … | 1.2 |

We can consider either W or W' as the word's representation.
Or even take the average.

# Word Analogies

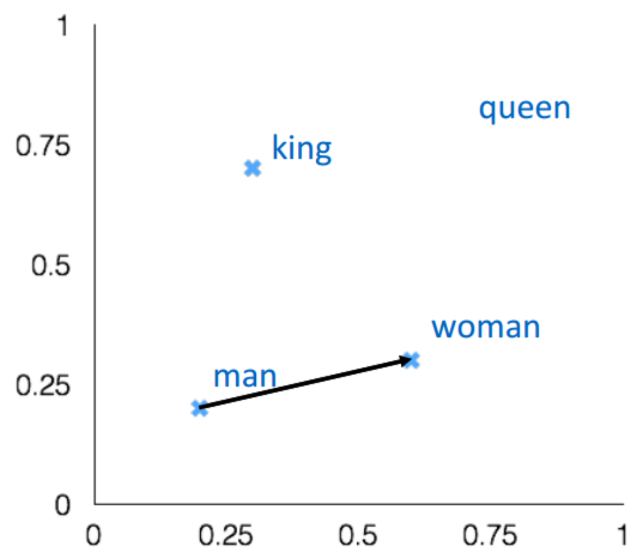
Test for linear relationships, examined by Mikolov et al. (2014)

a:b :: c:?

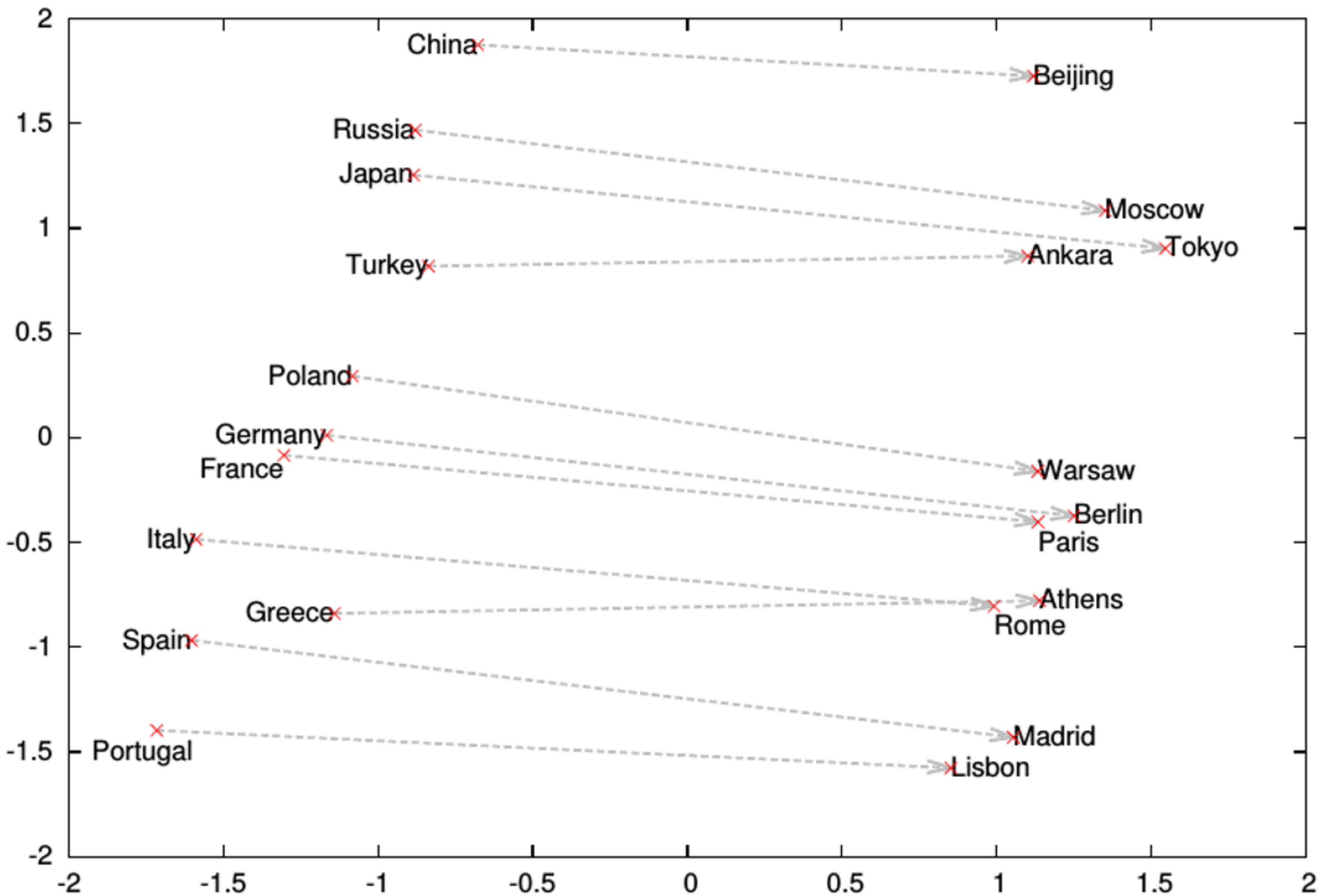
$$d = \arg\max_x \frac{(w_b - w_a + w_c)^T w_x}{||w_b - w_a + w_c||}$$

man:woman :: king:?

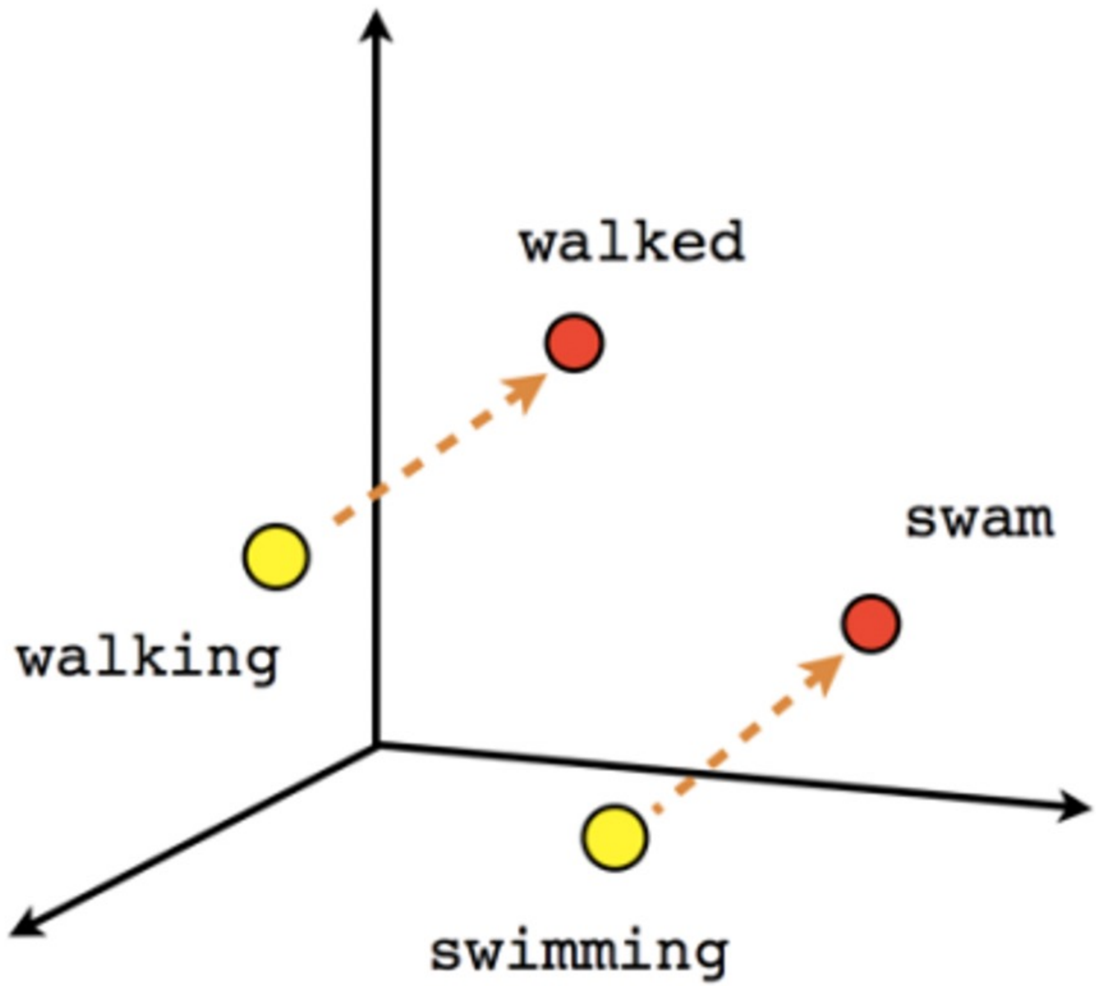
| | | |
|---|---|---|
| + | king | [ 0.30 0.70 ] |
| - | man | [ 0.20 0.20 ] |
| + | woman | [ 0.60 0.30 ] |
| | queen | [ 0.70 0.80 ] |



# Word analogies



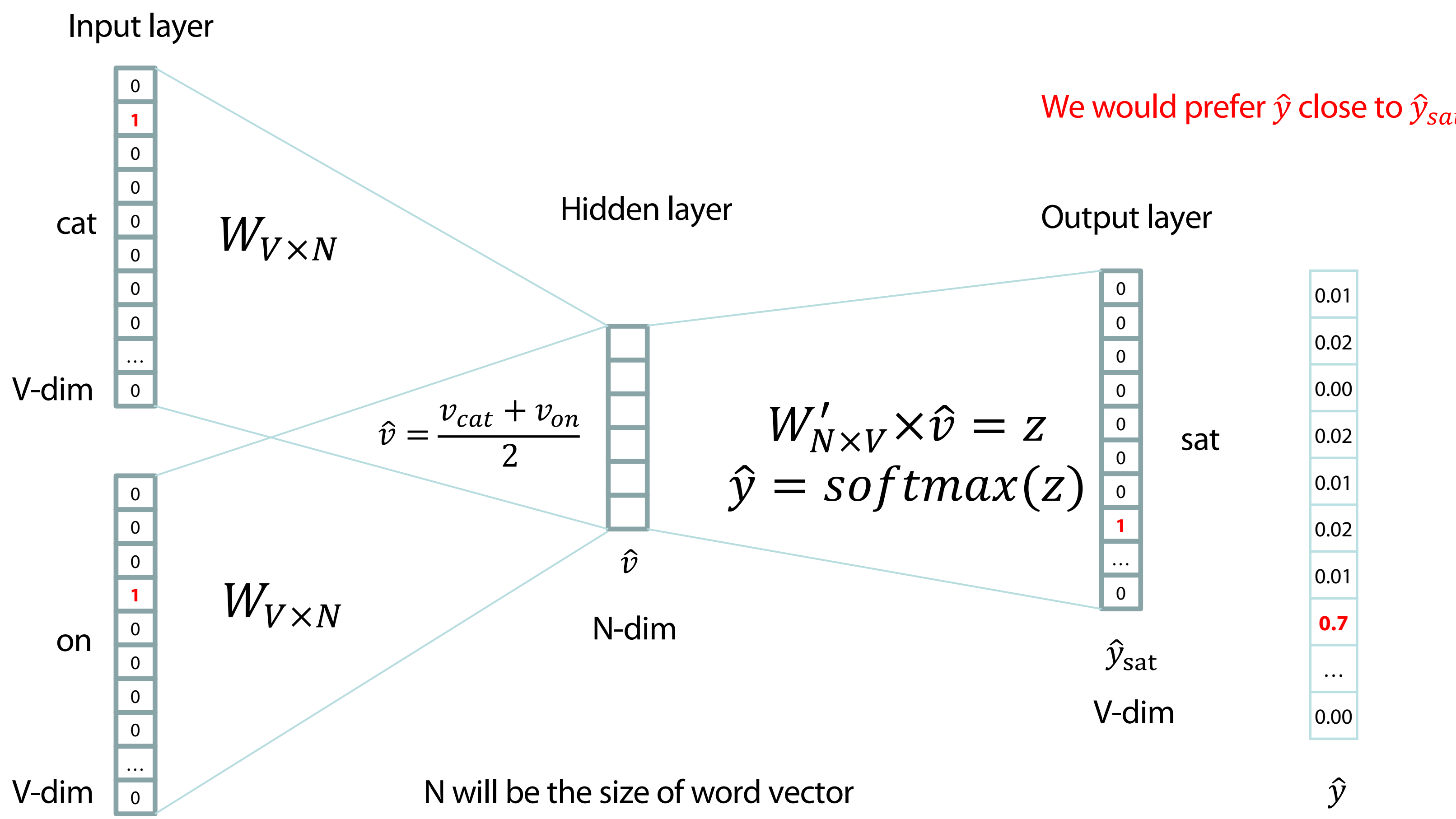
# Word Analogies (Tense)



# Extrinsic Evaluation

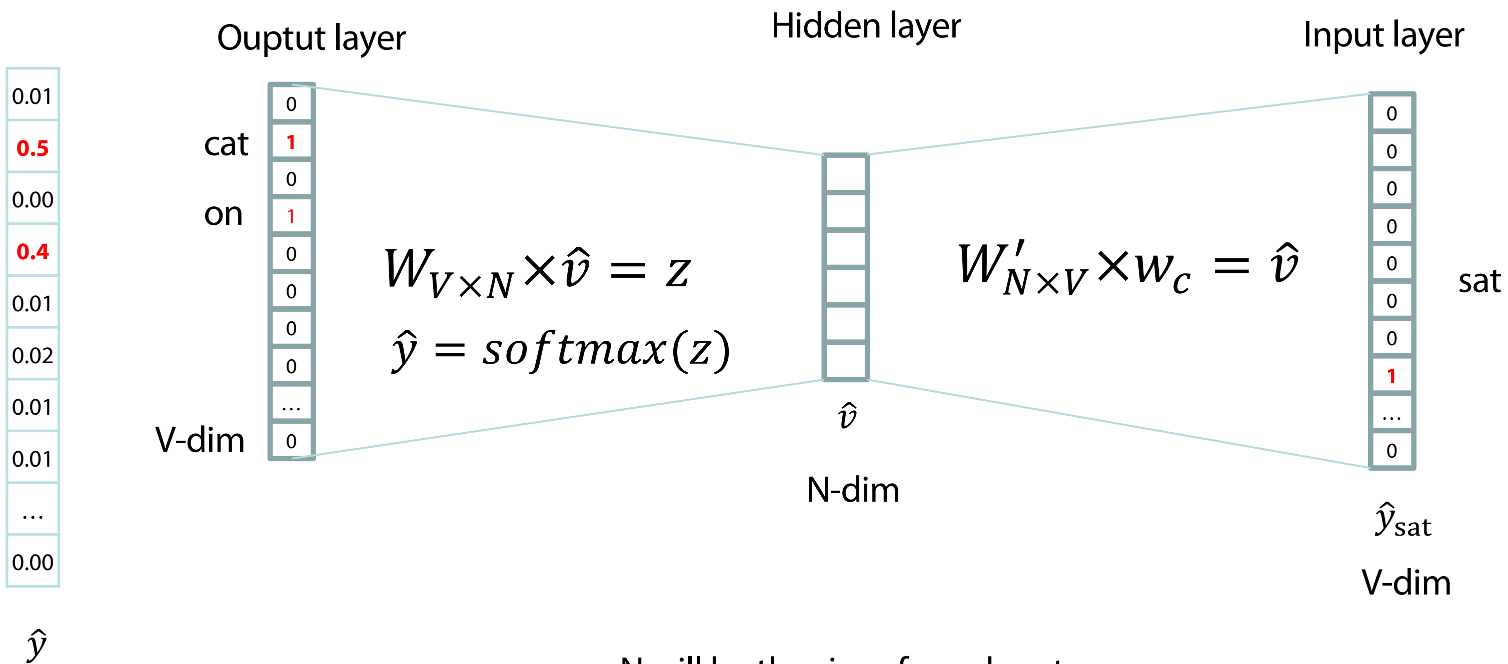
- Evaluate in applications
  - Sentiment analysis
  - …

# CBOW


Input layer
cat
V-dim
on
V-dim
$W_{V\times N}$
$W_{V\times N}$
Hidden layer
$\hat{v} = \frac{v_{cat} + v_{on}}{2}$
$\hat{v}$
N-dim
Output layer
$W'_{N\times V}\times\hat{v} = z$
$\hat{y} = softmax(z)$
sat
$\hat{y}_{\text{sat}}$
V-dim
We would prefer $\hat{y}$ close to $\hat{y}_{sat}$
0.01
0.02
0.00
0.02
0.01
0.02
0.01
0.7
…
0.00
$\hat{y}$
N will be the size of word vector

## Skip-Gram

We would prefer $\hat{y}$ close to $z$




N will be the size of word vector

# Skip-Gram Model

Objective: Given $w_c$, predict $w_{c-k}\ \ldots, w_{c-1}, w_{c+1}, \ldots, w_{c+k}$

Training data: Given sequence of words $< w_1, w_2, \ldots w_n >$, extract input and output: $(w_c\ ;\ w_{c-k}\ \ldots, w_{c-1}, w_{c+1}, \ldots, w_{c+k})$

Knowns:

- Training data $\{(w_c\ ;\ w_{c-k}\ \ldots, w_{c-1}, w_{c+1}, \ldots, w_{c+k})\}$
- Vocabulary $\{w_1, w_2, \ldots w_V\}$ of the training corpus

Unknowns:

- Word embedding matrices $W_{V\,\mathrm{x}\,N}$ and $W'_{N\,\mathrm{x}\,V}$ with $N$ being a hyperparameter

# Skip-Gram: Derivation of Learning Procedure

$$Minimize \quad -\log P(w_{c-m}, \dots, w_{c-1}, w_{c+1}, \dots, w_{c+m} | \, w_c)$$

$$= -\log \prod_{j=0, j\neq m}^{2m} P\left(w_{c-m+j} \,\middle|\, v_c\right) \quad \text{(and due to softmax)}$$

$$= -\log \prod_{j=0, j\neq m}^{2m} \frac{e^{W_{c-m+j}\, v_c}}{\sum_{k=1}^{V} e^{W_k\, v_c}}$$

$$= -(\sum_{j=0, j\neq m}^{2m} W_{c-m+j}\; v_c) \quad + 2m \log \sum_{k=1}^{V} e^{W_k\, v_c}$$

Use stochastic gradient decent to minimize and then to update word vectors

where $v_c = W' w_c$
(no averaging for skip-gram)

word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method, Yoav Goldberg and Omer Levy, arxiv, **2014**.

# What is `word2vec`?

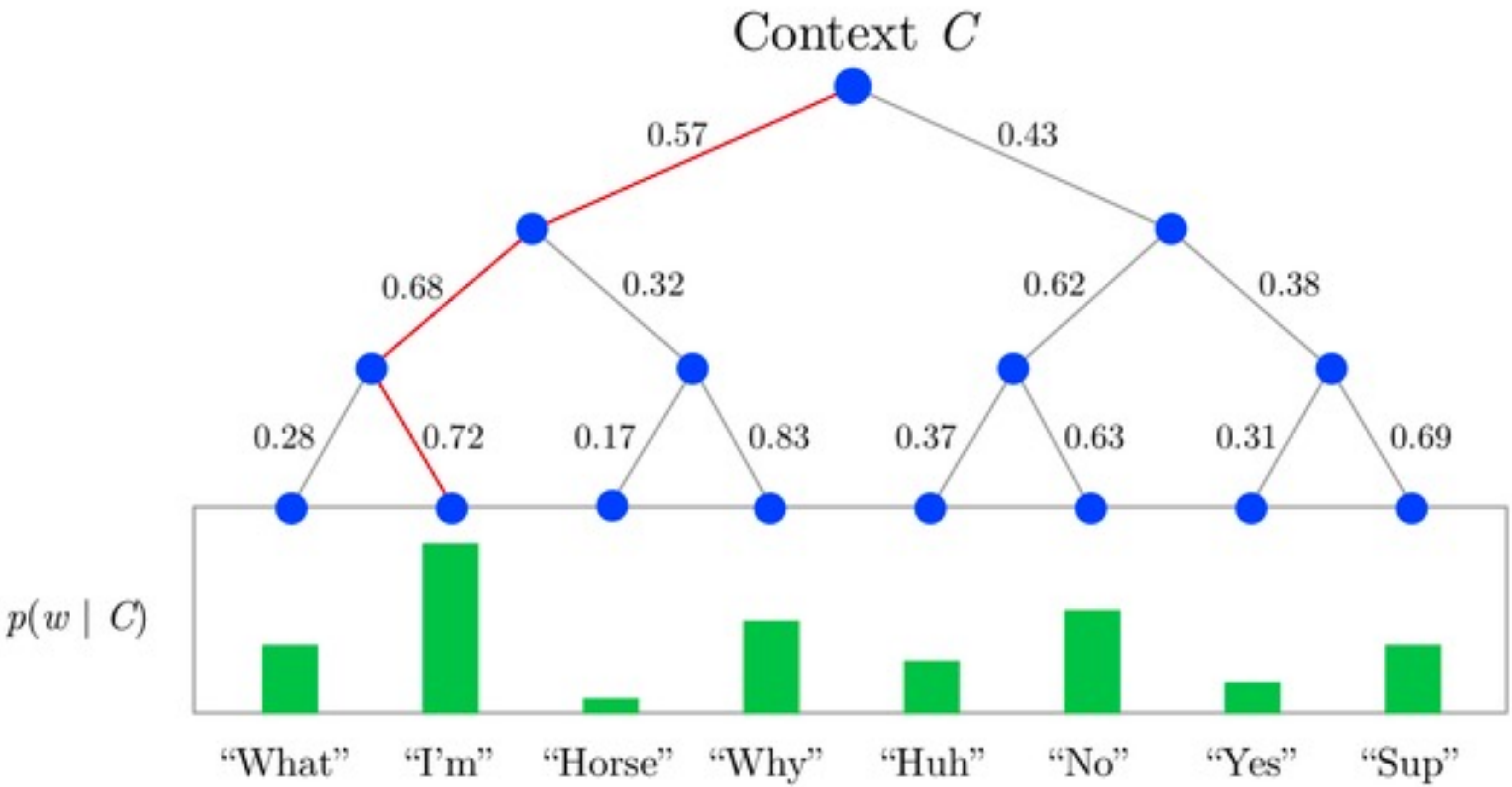
- `word2vec` is **not** a single algorithm
- It is a **software package** for representing words as vectors, containing:
  - Two distinct models
    - CBoW
    - Skip-Gram
  - Various training methods
    - Softmax is a bottleneck (discussed next)
  - A rich preprocessing pipeline
    - Dynamic Context Windows
    - Subsampling of Frequent Words
    - Deleting Rare Words (left out)

# Softmax is a Bottleneck (CBOW and Skip-Gram)

- The denominator is a sum across entire vocabulary
- $-(\sum_{j=0, j\neq m}^{2m} W_{c-m+j}\ v_c) \quad + 2m \log \sum_{k=1}^{V} e^{W_k\ v_c}$
- To be computed for every window
  - Too expensive
  - Single update of parameters requires iteration of entire vocabulary (which usually is in millions)
- Various optimized training methods
  - Hierarchical Softmax (use binary tree)
    - Probability of a word is calculated through the product of probabilities on each edge on the path to that node
  - Noise Contrastive Estimation (left out)
  - Negative Sampling

Rong, X. word2vec Parameter Learning Explained (cite arxiv:1411.2738). **2014**.

# Tree for Computing Word Probabilities



# Skip-Grams with Negative Sampling (SGNS)

Marco saw a furry little wampimuk hiding in the tree.

Distributed Representations of Words and Phrases and their Compositionality
Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean, NIPS **2013**

# Skip-Grams with Negative Sampling (SGNS)

Marco saw a furry little wampimuk hiding in the tree.

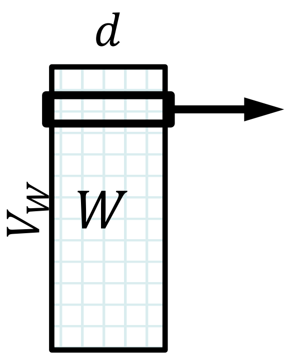
| **words** | **contexts** |
|---|---|
| wampimuk | furry |
| wampimuk | little |
| wampimuk | hiding |
| wampimuk | in |
| … | … |

$D$ (data)

“word2vec Explained…”
Goldberg & Levy, arXiv 2014

# Skip-Grams with Negative Sampling (SGNS)

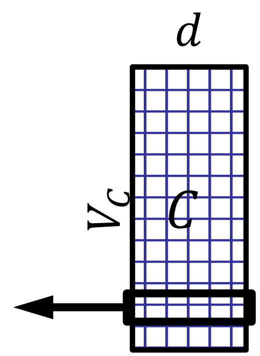
- SGNS finds a vector $\vec{w}$ for each word $w$ in our vocabulary $V_W$
- Each such vector has $d$ latent dimensions (e.g. $d = 100$)
- Effectively, it learns a matrix $W$ whose rows represent $V_W$
- **Key point:** it also learns a similar auxiliary matrix $C$ of context vectors
- In fact, each word has two embeddings




$w$:wampimuk $=$
$(-3.1, 4.15, 9.2, -6.5, \dots)$

$\neq$

$c$:wampimuk $=$
$(-5.6, 2.95, 1.4, -1.3, \dots)$




$d$ was called $N$ before

"word2vec Explained…"
Goldberg & Levy, arXiv 2014

# Coming back to Negative Sampling

- Given $(w, c)$: word and context
- Let $P(D = 1|\, w, c)$ be the probability that $(w, c)$ came from the corpus data
- $P(D = 0|\, w, c)$ = probability that $(w, c)$ are not from the corpus data
- Let us model $P(D = 1|\, w, c)$ with $sigmoid$
- $P(D = 1|\, w, c) = sigmoid(u_w^T v_c) = \frac{1}{1+e^{-u_w^T v_c}}$

  $u_v = Ww \quad v_c = Cc$
- Objective:
  - Maximize $P(D = 1|\, w, c)$ if $(w, c)$ is in the corpus data
  - Minimize$P(D = 1|\, w, c)$ if $(w, c)$ not in the corpus data

Distributed Representations of Words and Phrases and their Compositionality
Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean, NIPS 2013

# Skip-Grams with Negative Sampling (SGNS)

- **Maximize:** $\sigma(\vec{w} \cdot \vec{c})$
  - $c$ was **observed** with $w$

| **words** | **contexts** |
|---|---|
| wampimuk | furry |
| wampimuk | little |
| wampimuk | hiding |
| wampimuk | in |

"word2vec Explained…"
Goldberg & Levy, arXiv 2014

# Skip-Grams with Negative Sampling (SGNS)

- **Maximize:** $\sigma(\vec{w} \cdot \vec{c})$
  - $c$ was **observed** with $w$

| **words** | **contexts** |
|---|---|
| wampimuk | furry |
| wampimuk | little |
| wampimuk | hiding |
| wampimuk | in |

- **Minimize:** $\sigma(\vec{w} \cdot \vec{c}')$
  - $c'$ was **hallucinated** with $w$

| **words** | **contexts** |
|---|---|
| wampimuk | Australia |
| wampimuk | cyber |
| wampimuk | the |
| wampimuk | 1985 |

"word2vec Explained…"
Goldberg & Levy, arXiv 2014

# Math behind Negative Sampling

Maximum Likelihood approach for learning $\theta = (\mathrm{W}, \mathrm{C})$

$$
\begin{aligned}
\theta &= \arg\max_{\theta} \prod_{(w,c)\in D} P(D=1|w,c,\theta) \prod_{(w,c)\in \tilde{D}} P(D=0|w,c,\theta) \\
&= \arg\max_{\theta} \prod_{(w,c)\in D} P(D=1|w,c,\theta) \prod_{(w,c)\in \tilde{D}} (1-P(D=1|w,c,\theta)) \\
&= \arg\max_{\theta} \sum_{(w,c)\in D} \log P(D=1|w,c,\theta) + \sum_{(w,c)\in \tilde{D}} \log(1-P(D=1|w,c,\theta)) \\
&= \arg\max_{\theta} \sum_{(w,c)\in D} \log \frac{1}{1+\exp(-u_w^T v_c)} + \sum_{(w,c)\in \tilde{D}} \log(1-\frac{1}{1+\exp(-u_w^T v_c)}) \\
&= \arg\max_{\theta} \sum_{(w,c)\in D} \log \frac{1}{1+\exp(-u_w^T v_c)} + \sum_{(w,c)\in \tilde{D}} \log(\frac{1}{1+\exp(u_w^T v_c)})
\end{aligned}
$$

$$u_v = Ww \quad v_c = Cc$$

Distributed Representations of Words and Phrases and their Compositionality
Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean, NIPS **2013**

# Math behind Negative Sampling

- Maximize log likelihood = minimize –log likelihood

$$-\sum_{(w,c)\in D} \log \frac{1}{1+\exp(-u_w^T v_c)} - \sum_{(w,c)\in \tilde{D}} \log\left(\frac{1}{1+\exp(u_w^T v_c)}\right)$$

Sigmoid

- $\tilde{D}$ is the negative corpus with wrong contexts
- Generate $\tilde{D}$ on the fly by randomly sampling from the vocabulary
- New objective function for observing context word $w_{c-m+j}$ $(j = 0..2m)$ given the center word $w_c$ would be

$$-\log \sigma(u_{c-m+j}^T \cdot v_c) - \sum_{k=1}^{\boxed{K}} \log \sigma(-\tilde{u}_k^T \cdot v_c)$$

$$-u_{c-m+j}^T v_c + \log \sum_{k=1}^{\boxed{|V|}} \exp(u_k^T v_c)$$

regular softmax loss for skip-gram

$$-\sum_{j=0,j\neq m}^{2m} W_{c-m+j}\, v_c \quad + 2m \log \sum_{k=1}^{|V|} e^{W_k v_c}$$

# Skip-Grams with Negative Sampling (SGNS)

- “Negative Sampling”
- SGNS samples $k$ contexts $c'$ **at random** as **negative examples**
- “Random” = unigram distribution

$$P\ (c) = \frac{\#c}{\sum_{c' \in V_C} (\#c')}$$

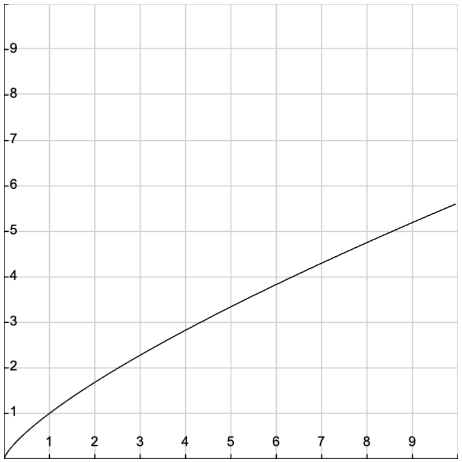
- Changing this distribution has a significant effect

Distributed Representations of Words and Phrases and their Compositionality
Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean, NIPS **2013**

# Context Distribution Smoothing

- In practice, it's a **smoothed** unigram distribution

$$P^{0.75}(c) = \frac{(\#c)^{0.75}}{\sum_{c' \in V_C} (\#c')^{0.75}}$$




- This little change makes a big difference



Distributed Representations of Words and Phrases and their Compositionality
Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean, NIPS **2013**

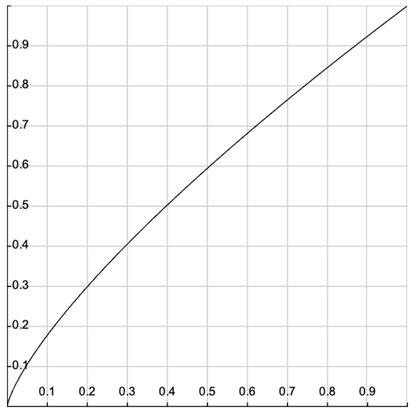
# Context Distribution Smoothing

- We can **adapt** context distribution smoothing to PMI!

- Replace $P(c)$ with $P^{0.75}(c)$

$$PMI^{0.75}(w,c) = \log \frac{P(w,c)}{P(w) \cdot \boldsymbol{P^{0.75}(c)}}$$

- Consistently improves **PMI** on **every task**
- **Always use Context Distribution Smoothing!**



# Math behind CBOW with Negative Sampling

- Likewise for CBOW $$\hat{v} = \frac{v_{c-m} + v_{c-m+1} + \ldots + v_{c+m}}{2m}$$
- Objective:

$$-\log \sigma(u_c^T \cdot \hat{v}) - \sum_{k=1}^{\boxed{K}} \log \sigma(-\tilde{u}_k^T \cdot \hat{v})$$

where $\{\tilde{u}_k \mid k = 1..\mathrm{K}\}$ is sampled from vocabulary (also use context distribution smoothing)

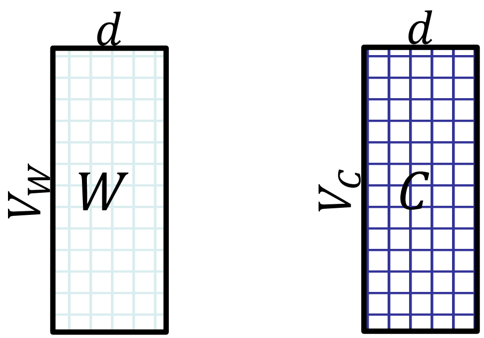
- Rather than:

$$-u_c^T \hat{v} + \log \sum_{j=1}^{\boxed{|V|}} \exp(u_j^T \hat{v})$$

regular softmax loss for CBOW

# What is SGNS learning?

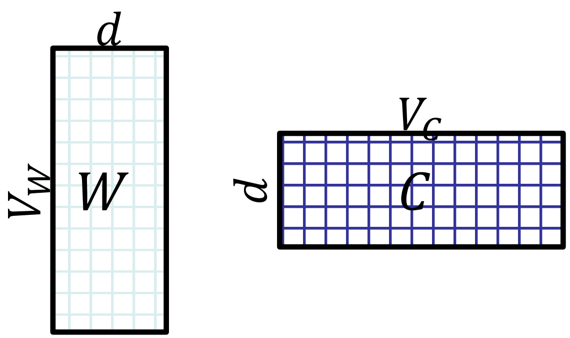
- Take SGNS's embedding matrices ($W$ and $C$)




"Neural Word Embeddings as Implicit Matrix Factorization"
Levy & Goldberg, NIPS **2014**

# What is SGNS learning?

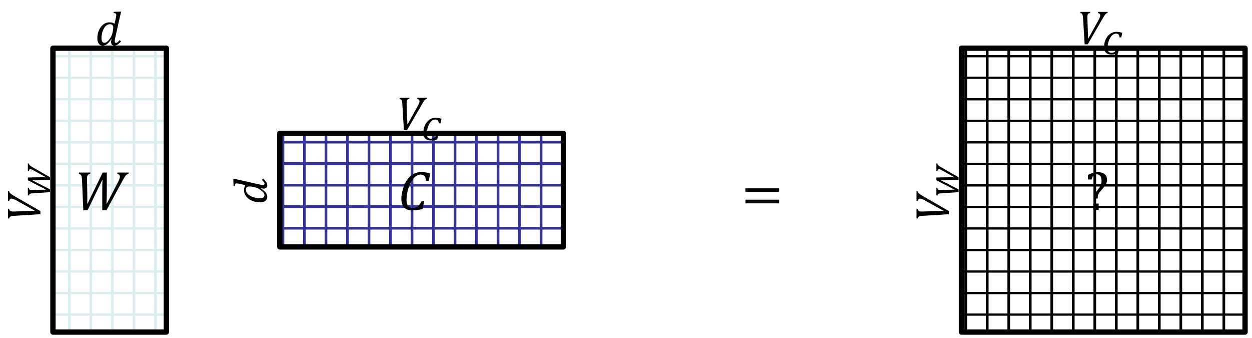
- Take SGNS's embedding matrices ($W$ and $C$)
- Multiply them
- What do you get?



# What is SGNS learning?

- A $V_W \times V_C$ matrix
- Each cell describes the relation between a specific word-context pair

$$\vec{w} \cdot \vec{c} = ?$$

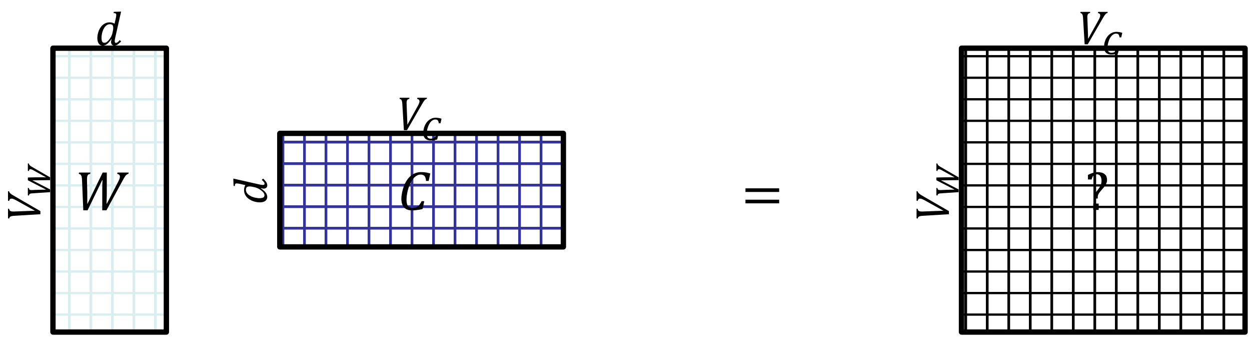





"Neural Word Embeddings as Implicit Matrix Factorization"
Levy & Goldberg, NIPS **2014**

# What is SGNS learning?

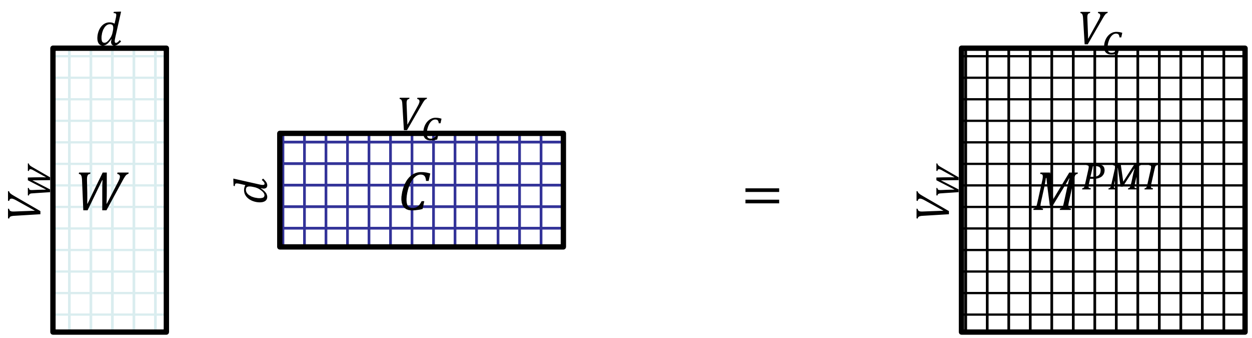
- Levy&Goldberg [2014] **proved** that for large enough $d$ and enough iterations …




“Neural Word Embeddings as Implicit Matrix Factorization”
Levy & Goldberg, NIPS **2014**

# What is SGNS learning?

- Levy&Goldberg [2014] **proved** that for large enough $d$ and enough iterations …
- … one obtains the word-context PMI matrix

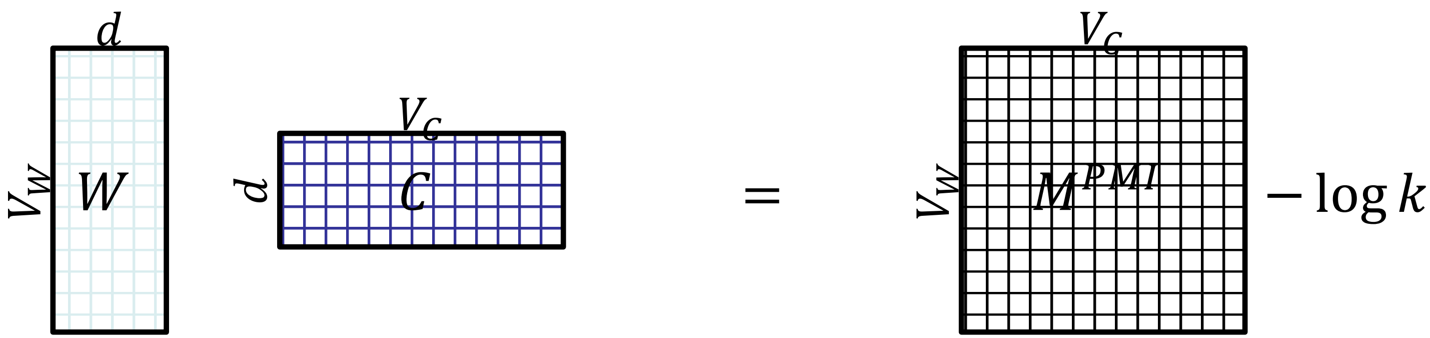



“Neural Word Embeddings as Implicit Matrix Factorization”
Levy & Goldberg, NIPS **2014**

# What is SGNS learning?

- Levy&Goldberg [2014] **proved** that for large enough $d$ and enough iterations …
- … one obtains the word-context PMI matrix …
- shifted by a global constant

$$Opt(\vec{w} \cdot \vec{c}) = PMI(w, c) - \log k$$




where $k$ is the number of negative examples



"Neural Word Embeddings as Implicit Matrix Factorization"
Levy & Goldberg, NIPS **2014**

# What is SGNS learning?

- SGNS is doing something very similar to the older approaches

- SGNS factorizes the traditional word-context PMI matrix

- So does SVD!

# But embeddings are still better, right?

- Plenty of evidence that embeddings outperform traditional methods
  - "Don't Count, Predict!" (Baroni et al., ACL 2014)
  - GloVe (Pennington et al., EMNLP 2014)

- How does this fit with our story?


Marco Baroni, Georgiana Dinu, Germán Kruszewski. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proc. ACL-14, 238–247, **2014**.

Jeffrey Pennington, Richard Socher, Christopher Manning. GloVe: Global Vectors for Word Representation. In: Proc. EMNLP-.14, 1532–1543, **2014**.

# The Big Impact of “Small” Hyperparameters

- `word2vec` & GloVe are more than just algorithms…
- Introduce **new hyperparameters**
- May seem minor, but **make a big difference** in practice

# New Hyperparameters

- **Preprocessing** **(word2vec)**
  - Dynamic Context Windows
  - Subsampling of Frequent Words
  - Deleting Rare Words

- **Postprocessing** **(GloVe)**
  - Adding Context Vectors

- **Association Metric** **(SGNS)**
  - Shifted PMI
  - Context Distribution Smoothing

# Dynamic Context Windows

Marco saw a furry little wampimuk hiding in the tree.

# Dynamic Context Windows

saw a furry little wampimuk hiding in the tree

# Dynamic Context Windows

saw a furry little wampimuk hiding in the tree

Word2vec: $\frac{1}{4}$ $\frac{2}{4}$ $\frac{3}{4}$ $\frac{4}{4}$ $\frac{4}{4}$ $\frac{3}{4}$ $\frac{2}{4}$ $\frac{1}{4}$

GloVe: $\frac{1}{4}$ $\frac{1}{3}$ $\frac{1}{2}$ $\frac{1}{1}$ $\frac{1}{1}$ $\frac{1}{2}$ $\frac{1}{3}$ $\frac{1}{4}$

Aggressive: $\frac{1}{8}$ $\frac{1}{4}$ $\frac{1}{2}$ $\frac{1}{1}$ $\frac{1}{1}$ $\frac{1}{2}$ $\frac{1}{4}$ $\frac{1}{8}$

**The Word-Space Model** *(Sahlgren, 2006)*

Magnus Sahlgren, The Word-Space Model,
Dissertation, Stockholm Univ., **2006**.

# Subsampling of Frequent Words

- Counter imbalance of rare and frequent words
- Each word in the training set is discarded with a probability computed by

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

- where $f(w_i)$ is the number of occurrences of word $w_i$ and $t$ is a chosen threshold

Distributed Representations of Words and Phrases and their Compositionality
Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean, NIPS **2013**

# Adding Context Vectors

- SGNS creates word vectors $\vec{w}$
- SGNS creates auxiliary context vectors $\vec{c}$
  - So do GloVe and SVD

- Instead of just $\vec{w}$
- Represent a word as: $\vec{w} + \vec{c}$

- Introduced by Pennington et al. (2014)
- Only applied to GloVe

# Don’t Count, Predict! ?

- “word2vec is better than count-based methods” [Baroni et al., 2014]

- **Hyperparameter settings** account for most of the reported gaps in count-based approaches

- Embeddings do **not** really outperform count-based methods

- No unique conclusion available

Marco Baroni, Georgiana Dinu, Germán Kruszewski. Don’t count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proc. ACL-14, 238–247, **2014**.

# Problem

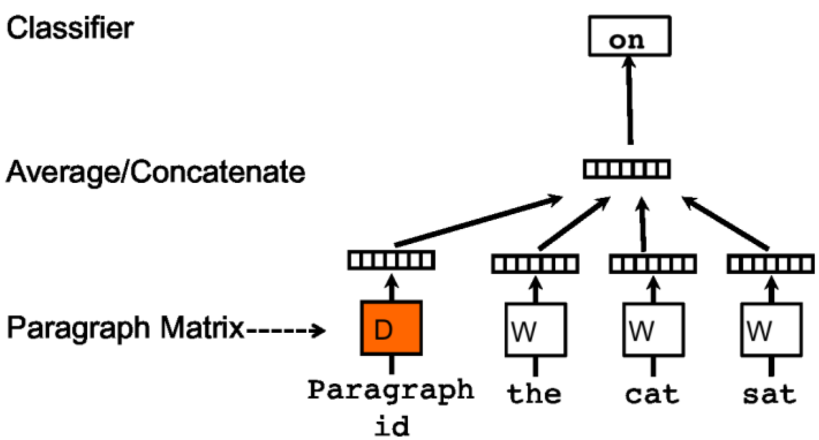
- Learn low-dimensional, dense representations (or embeddings) for documents.
- Document embeddings can be used off-the-shelf to solve many IR applications such as,
  - Document Classification
  - Document Retrieval
  - Document Ranking

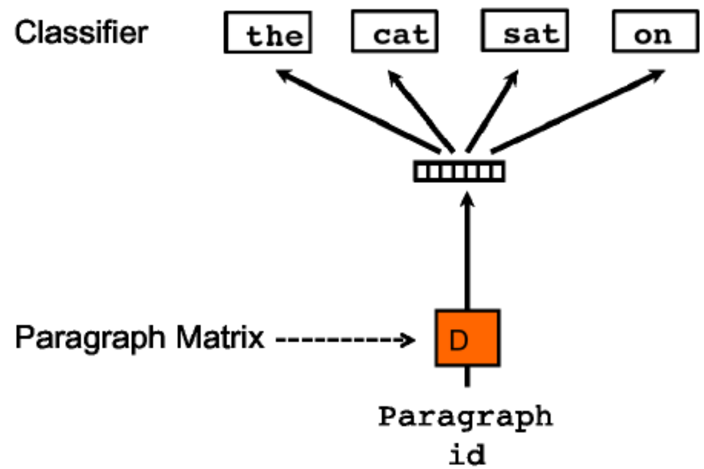
# Power of 2Vec Representations

- Bag-of-words (BOW) or Bag-of-n-grams
  - Data sparsity
  - High dimensionality
  - Not/hardly capturing word order
- Latent Dirichlet Allocation (LDA)
  - Computationally inefficient for larger dataset.
- Paragraph Vector
  - Dense representation
  - Compact representation
  - Captures word order
  - Efficient to estimate

# Represent the meaning of sentence/paragraph/doc

- Paragraph Vector (Le and Mikolov, 2014)
  - Extend word2vec to text level
  - Also two models: add paragraph vector as the input




Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In Proceedings ICML'14. **2014**.

# Paragraph Vector

- Learn document embedding by predicting the next word in the document using the context of the word and the ('unknown') document vector as features.
- Resulting vector captures the topic of the document.
- Update the document vectors, but not the word vectors [Le et al.]
- Update the document vectors, along with the word vectors [Dai et al.]
  - Improvement in the accuracy for document similarity tasks.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In Proceedings ICML'14. **2014**.

Dai, A.M., Olah, C., Le, Q.V., Corrado, G.S.: Document embedding with paragraph vectors. In: NIPS Deep Learning Workshop. **2014**

# Doc2Sent2Vec Idea - Being granular helps

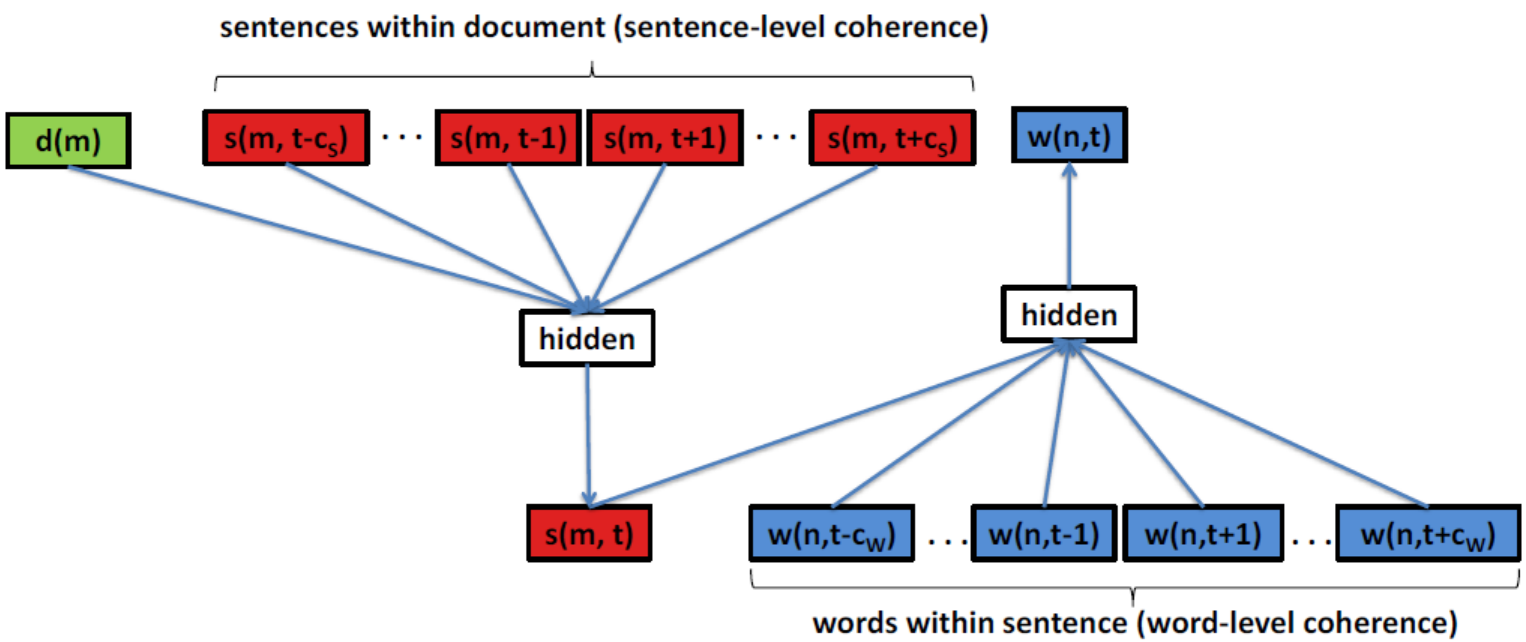
- Should we learn the document embedding from the word context directly?
- Can we learn the document embedding from the sentence context?
  - Explicitly exploit the sentence-level and word-level coherence to learn document and sentence embedding respectively.

# Notation

- Document Set: $D = \{d_1, d_2, \ldots, d_M\}$; 'M' documents;
- Document: $d_m = \{s(m,1), s(m,2), \ldots, s(m,T_m)\}$; '$T_m$' sentences;
- Sentence: $s(m,n) = \{w(n,1), w(n,2), \ldots, w(n,T_n)\}$; '$T_n$' words;
- Word: $w(n,t)$;

*Doc2Sent2Vec's goal is to learn low-dimensional representations of words, sentences and documents as a continuous feature vector of dimensionality $D_w$, $D_s$ and $D_d$ respectively.*

# Architecture Diagram



# Phase 1: Learn Sentence Embedding

Idea: Learn sentence representation from the word sequence within the sentence.

Input Features:

- Context words for target word w(n,t): $w(n,t-c_w), \ldots, w(n,t-1), w(n,t+1), \ldots, w(n,t+c_w)$ (where '$c_w$' is the word context size)
- Target Sentence: s(m,n) (where 'm' is the document id)

Output: w(n,t)

Task: Predict the target word using the concatenation of word vectors of context words along with the sentence vector as features.

- Maximize the word likelihood:

$$L_{word} = P(w(n,t) \mid w(n,t-c_w), \ldots, w(n,t-1), w(n,t+1), \ldots, w(n,t+c_w), s(m,n))$$

# Phase 2: Learn Document Embedding

**Idea:** Learn document representation from the sentence sequence within the document.

**Input Features:**

- Context sentences for target sentence s(m,t): $s(m,t-c_S), \ldots, s(m,t-1), s(m,t+1), \ldots, s(m,t+c_S)$ (where '$c_S$' is the sentence context size)
- Target Document: d(m)

**Output:** s(m,t)

**Novel Task:** Predict the target sentence using the concatenation of sentence vectors of context sentences along with the document vector as features.

- Maximize the sentence likelihood:
  $L_{sent} = P(s(m,t) \mid s(m,t-c_S), \ldots, s(m,t-1), s(m,t+1), \ldots, s(m,t+c_S), d(m))$

# Training

- Overall objective function: $L = L_{word} + L_{sent}$
- Use Stochastic Gradient Descent (SGD) to learn parameters
- Use Hierarchical Softmax (Mikolov et al.) to facilitate faster training

# Latent Relational Structures

Processing natural language data:

- ✓ Tokenization/Sentence Splitting
- ✓ Part-of-speech (POS) tagging
- Phrase chunking
- Named entity recognition
- Coreference resolution
- Semantic role labeling

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In Proceedings ICML '08. pp. 160–167. **2008**.

# Phrase Chunking

- Identifies phrase-level constituents in sentences

  [NP Boris] [ADVP regretfully] [VP told] [NP his wife] [SBAR that] [NP their child] [VP could not attend] [NP night school] [PP without] [NP permission] .

- Useful for filtering: identify e.g. only noun phrases, or only verb phrases
- Used as source of features, e.g. distance, (abstracts away determiners, adjectives, for example), sequence,…
  - More efficient to compute than full syntactic parse
  - Applications in e.g. Information Extraction – getting (simple) information about concepts of interest from text documents
- Hand-crafted chunkers (regular expressions/finite automata)
- HMM/CRF-based chunk parsers derived from training data

# Named Entity Recognition

- Identifies and classifies strings of characters representing proper nouns
- **[PER Neil A. Armstrong]** , the 38-year-old civilian commander, radioed to earth and the mission control room here: "**[LOC Houston]** , **[ORG Tranquility]** Base here; the Eagle has landed."
- Useful for filtering documents
  - "I need to find news articles about organizations in which Bill Gates might be involved…"
- Disambiguate tokens: "Chicago" (team) vs. "Chicago" (city)
- Source of abstract features
  - E.g. "Verbs that appear with entities that are Organizations"
  - E.g. "Documents that have a high proportion of Organizations"

# Named Entity Recogniton: Definition

- NE involves identification of *proper names* in texts, and classification into a set of predefined categories of interest
  - Three universally accepted categories: person, location and organisation
  - Other common tasks: recognition of date/time expressions, measures (percent, money, weight etc), email addresses etc.
  - Other domain-specific entities: names of drugs, medical conditions, names of ships, bibliographic references etc
- NER ist not easy

# Named Entity Classification

- Category definitions are intuitively quite clear, but there are many grey areas.
- Many of these grey areas are caused by metonymy.
  - Person vs. Artefact: "The ham sandwich wants his bill." vs "Bring me a ham sandwich."
  - Organisation vs. Location : "England won the World Cup" vs. "The World Cup took place in England".
  - Company vs. Artefact: "shares in MTV" vs. "watching MTV"
  - Location vs. Organisation: "she met him at Heathrow" vs. "the Heathrow authorities"

# Basic Problems in NE

- Variation of NEs – e.g. John Smith, Mr Smith, John.
- Ambiguity of NE types
  - John Smith (company vs. person)
  - May (person vs. month)
  - Washington (person vs. location)
  - 1945 (date vs. time)
- Ambiguity with common words, e.g. “may”

# More complex problems in NER

- Issues of style, structure, domain, genre etc.
  - Punctuation, spelling, spacing, formatting, ….all have an impact

Dept. of Computing and Maths
Manchester Metropolitan University
Manchester
United Kingdom

> Tell me more about Leonardo
> Da Vinci

# List Lookup Approach

- System that recognises only entities stored in its lists (gazetteers).
- Advantages - Simple, fast, language independent, easy to retarget
- Disadvantages – collection and maintenance of lists, cannot deal with name variants, cannot resolve ambiguity

# Shallow Parsing Approach

- Internal evidence – names often have internal structure. These components can be either stored or guessed.

**location:**

CapWord + {City, Forest, Center}

*e.g. Sherwood Forest*

Cap Word + {Street, Boulevard, Avenue, Crescent, Road}

e.g. *Portobello Street*

# Shallow Parsing Approach

- External evidence - names are often used in very predictive local contexts

**Location:**

"to the" COMPASS "of" CapWord

e.g. *to the south of* ***Loitokitok***

"based in" CapWord

e.g. *based in* ***Loitokitok***

CapWord "is a" (ADJ)? GeoWord

e.g. ***Loitokitok*** *is a friendly city*

# Difficulties in Shallow Parsing Approach

- **Ambiguously capitalised words** (first word in sentence)

  [All American Bank] vs. All [State Police]

- **Semantic ambiguity**

  "John F. Kennedy" = airport (location)

  "Philip Morris" = organisation

- **Structural ambiguity**

  [Cable and Wireless] vs. [Microsoft] and [Dell]

  [Center for Computational Linguistics] vs. message from [City Hospital] for [John Smith].

# Coreference

- Identify all phrases that refer to each entity of interest – i.e., group mentions of concepts
- **[Neil A. Armstrong]** , **[the 38-year-old civilian commander]**, radioed to **[earth]**. **[He]** said the famous words, "**[the Eagle]** has landed"."
- The Named Entity Recognizer only gets us part-way…
- …if we ask, "what actions did Neil Armstrong perform?", we will miss many instances (e.g., "He said…")
- Coreference resolver **abstracts over different ways of referring to the same person**
  - Useful in feature extraction, information extraction

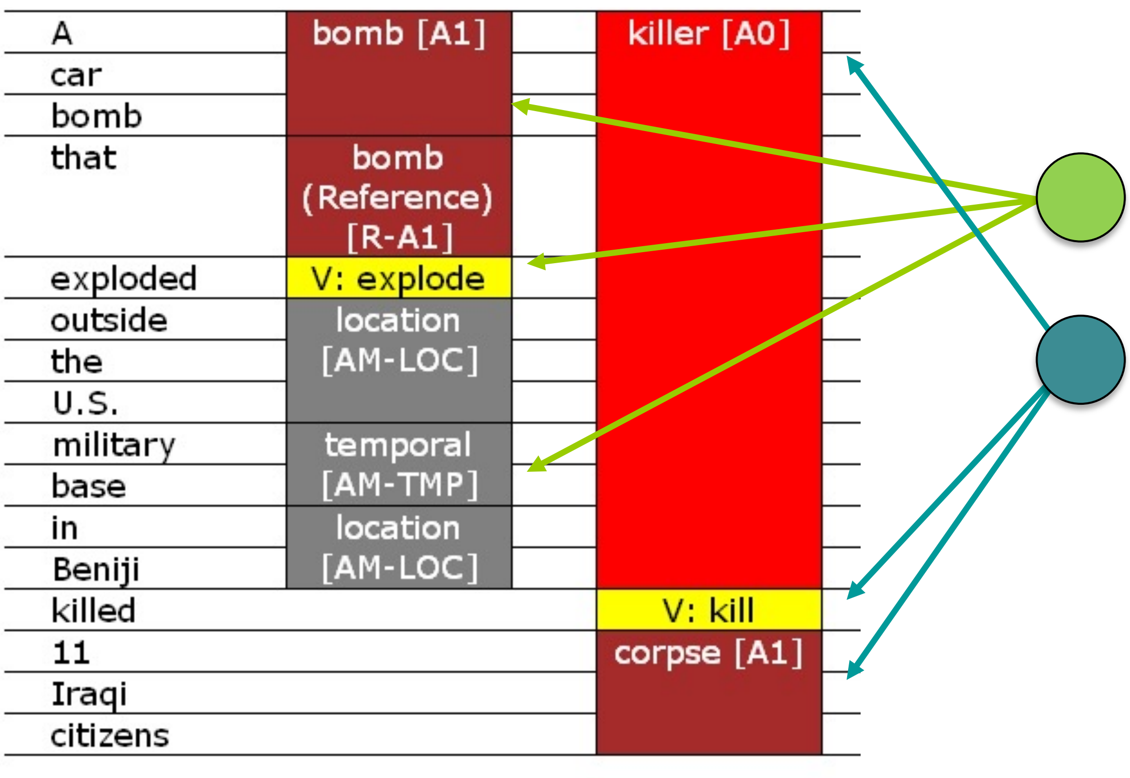
# Semantic Role Labeling (SRL)

**Input Text:**

A car bomb that exploded outside the U.S. military base in Beniji killed 11 Iraqi citizens.

**Result: Complete!**

⊞ General Explanation of Argument Labels




- SRL reveals relations and arguments in the sentence (where relations are expressed as verbs)
- Cannot abstract over variability of expressing the relations – e.g. kill vs. murder vs. slay…

# Why is SRL Important – *Applications*

- Question Answering
  - Q: When was Napoleon defeated?
  - Look for: [PATIENT Napoleon] [PRED **defeat-synset**] [ARGM-TMP *ANS*]

- Machine Translation

| English (SVO) | Farsi (SOV) | |
|---|---|---|
| [AGENT The little boy] | [AGENT pesar koocholo] | boy-little |
| [PRED **kicked**] | [THEME toop germezi] | ball-red |
| [THEME the red ball] | [ARGM-MNR moqtam] | hard-adverb |
| [ARGM-MNR hard] | [PRED **zaad-e**] | hit-past |

- Document Summarization
  - Predicates and Heads of Roles summarize content

- Information Extraction
  - SRL can be used to construct useful rules for IE

# Some History

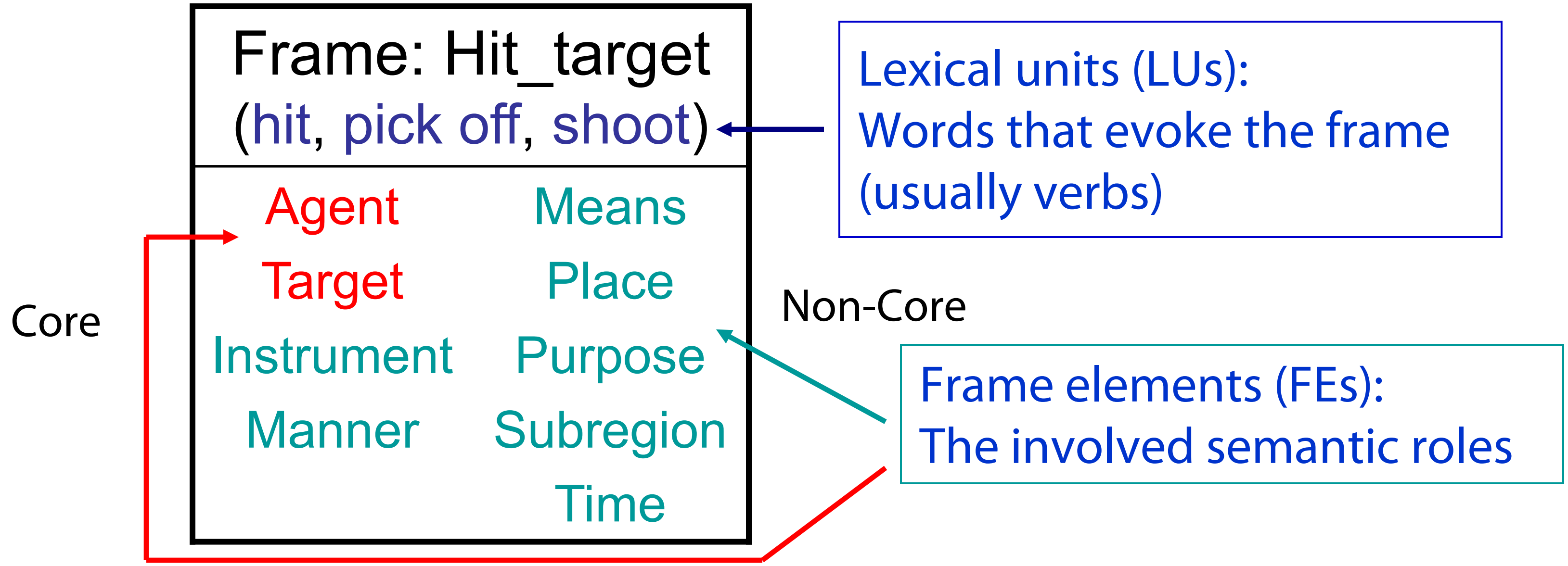
- Minsky 74, Fillmore 1976: *Frames* describe events or situations
  - Multiple participants, "props", and "conceptual roles"
  - E.g., agent, instrument, target, time, …
- Levin 1993: verb class defined by sets of frames (meaning-preserving alternations) a verb appears in
  - {*break,shatter,..*}: *Glass X's easily; John Xed the glass, …*
  - *Cut* is different: *The window broke; *The window cut.*
- FrameNet, late '90s: based on Levin's work: large corpus of sentences annotated with *frames*
- PropBank


Marvin Minky. A Framework for Representing Knowledge Marvin Minsky, MIT-AI Laboratory Memo 306, June, **1974**.

Charles J. Fillmore, Frame semantics and the nature of language Annals of the New York Academy of Sciences 280(1):20 – 32, **1976**.

Levin, B. English Verb Classes and Alternations: A Preliminary Investigation, University of Chicago Press, Chicago, IL. **1993**.

# FrameNet




[$_{\text{Agent}}$ *Kristina*] ***hit*** [$_{\text{Target}}$ *Scott*] [$_{\text{Instrument}}$ *with a baseball*] [$_{\text{Time}}$ *yesterday* ].

Gildea, Daniel; Jurafsky, Daniel. "Automatic Labeling of Semantic Roles”. Computational Linguistics. 28 (3): 245–288. **2002**.

# Proposition Bank (PropBank)

- Transfer sentences to propositions
  - Kristina hit Scott → hit(Kristina,Scott)
- Penn TreeBank → PropBank
  - Add a semantic layer on Penn TreeBank
  - Define a set of semantic roles for each verb
  - Each verb's roles are numbered

…[**A0** the company] to … *offer* [A1 a 15% to 20% stake] [**A2** to the public]
…[**A0** Sotheby's] … *offered* [**A2** the Dorrance heirs] [A1 a money-back guarantee]
…[A1 an amendment] *offered* [**A0** by Rep. Peter DeFazio] …
…[**A2** Subcontractors] will be *offered* [A1 a settlement] …



Palmer M, Kingsbury P, Gildea D. "The Proposition Bank: An Annotated Corpus of Semantic Roles". Computational Linguistics. 31 (1): 71–106. **2005**.

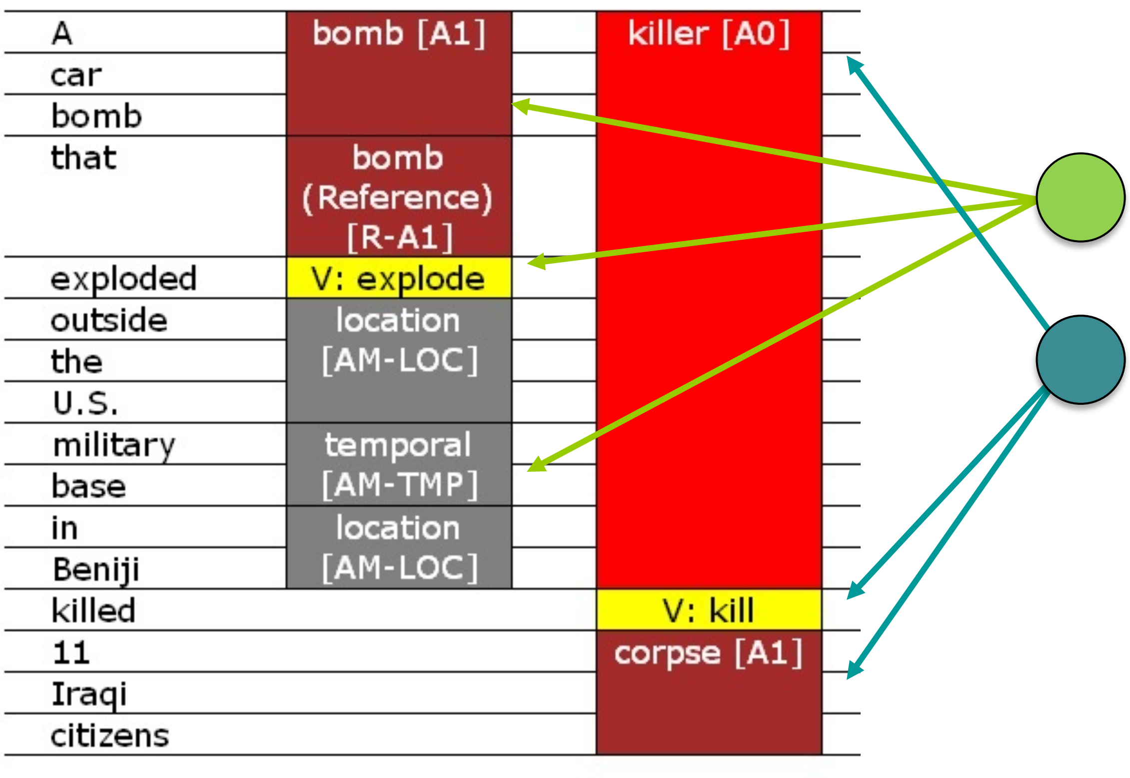
# Semantic Role Labeling (SRL)

**Input Text:**

A car bomb that exploded outside the U.S. military base in Beniji killed 11 Iraqi citizens.

**Result: Complete!**

⊞ General Explanation of Argument Labels




- SRL reveals relations and arguments in the sentence (where relations are expressed as verbs)
- Cannot abstract over variability of expressing the relations – e.g. kill vs. murder vs. slay…