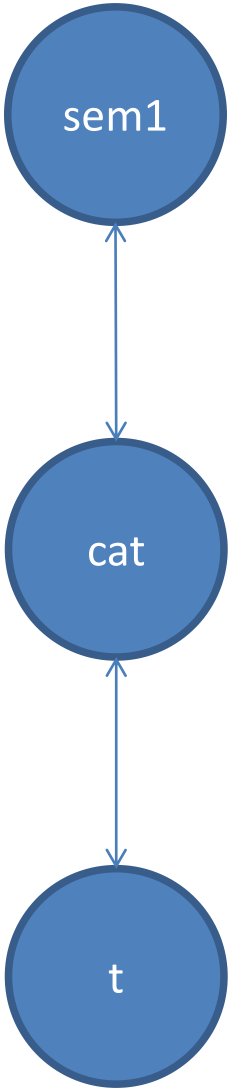
Discussion of Dell & Gordon’s Formula

Comparison of the matrix representation with the mathematics formula

Tengchao Zhou

May 18, 2010

Let us focus on a small part of the model in order to see how the MATLAB file model2matrix.m works.



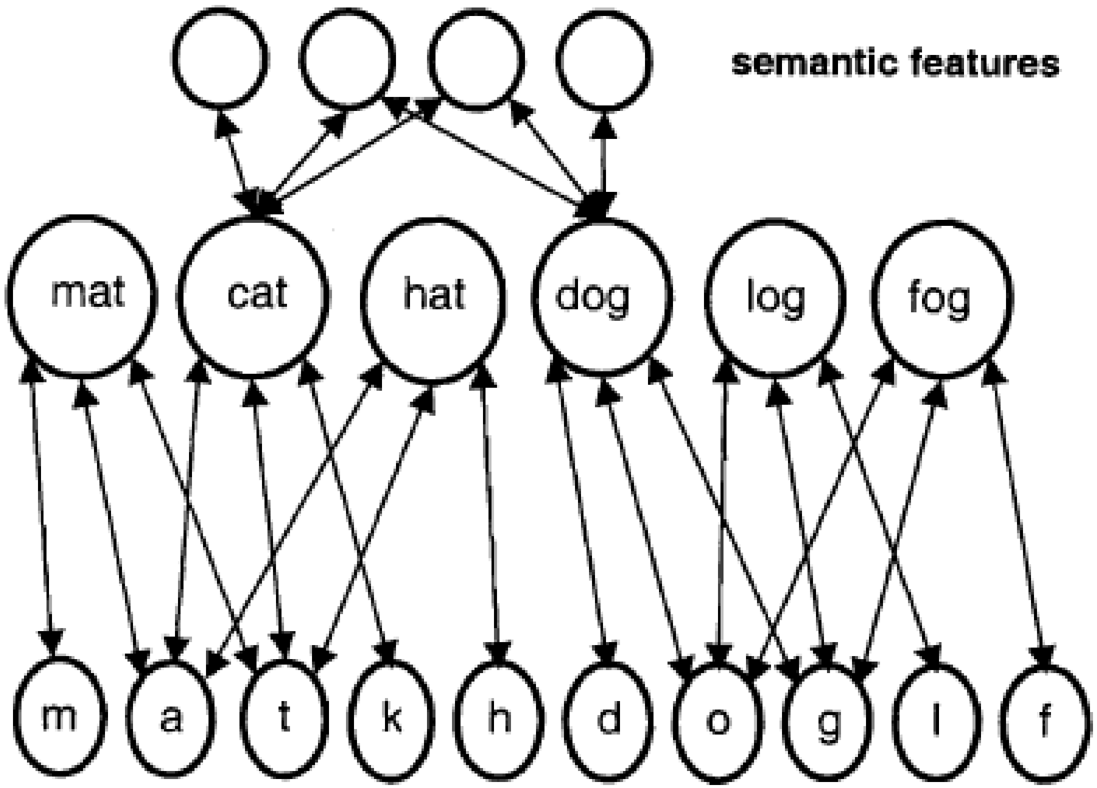
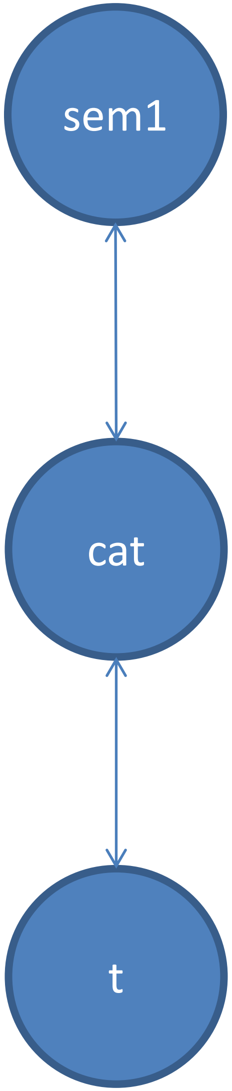


Figure 1. The Architecture of the Two-Step Interactive-Activation Model



We can write MATLAB code for this small part of the model.

Set up a 3-by-3 matrix here.

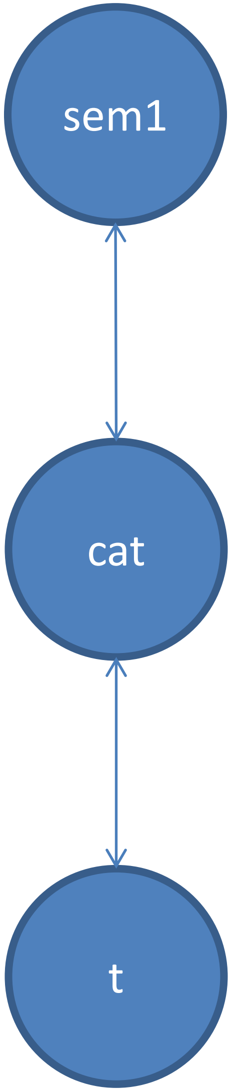
```
matrix = zeros(3, 3);
```

	col1	col2	col3
row1			
row2			
row3			

Assign col1 & row1 to represent all possible connections of sem1.
Assign col2 & row2 to represent all possible connections of cat.
Assign col3 & row3 to represent all possible connections of t.

```
sem1=1;
cat=2;
t=3;
```

	sem1	cat	t
sem1			
cat			
t			



	sem1	cat	t
sem1			
cat	***		
t			

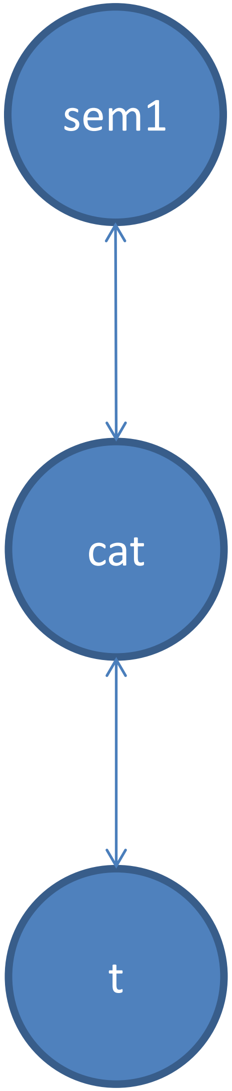
The entry with *** represents the connection from sem1 to cat.

	sem1	cat	t
sem1		***	
cat			
t			

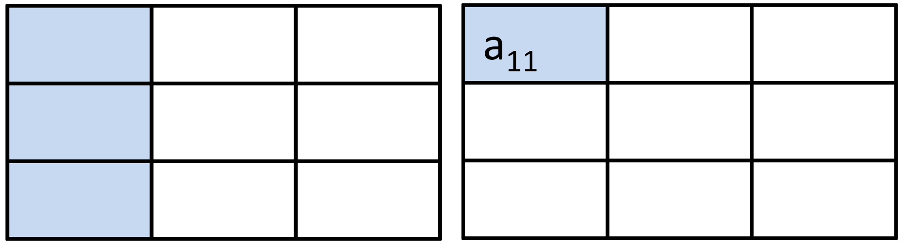
The entry with *** represents the connection from cat to sem1.

	sem1	cat	t
sem1	***		
cat			
t			

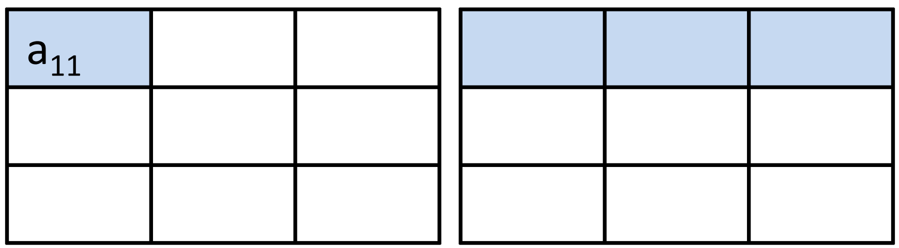
The entry with *** does not present any connection, but it can be looked as the representation of the node sem1.



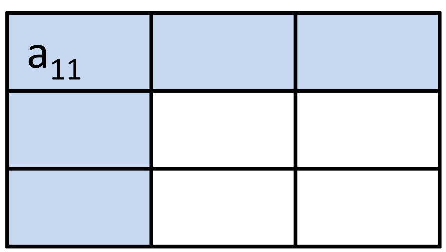
When two matrices multiply, the a_{11}-entry of the later matrix will influence the first column of the former matrix.

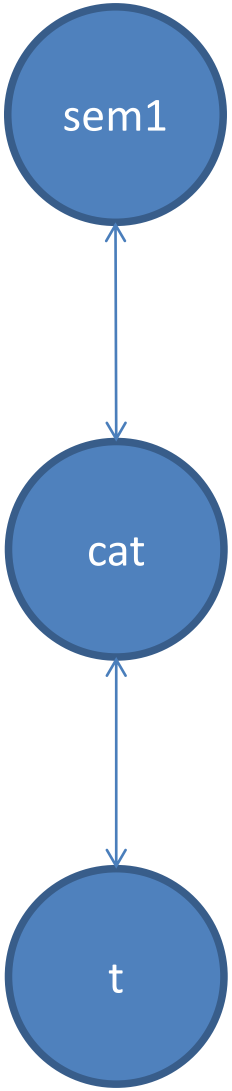


When two matrices multiply, the a_{11}-entry of the former matrix will influence the first row of the later matrix.

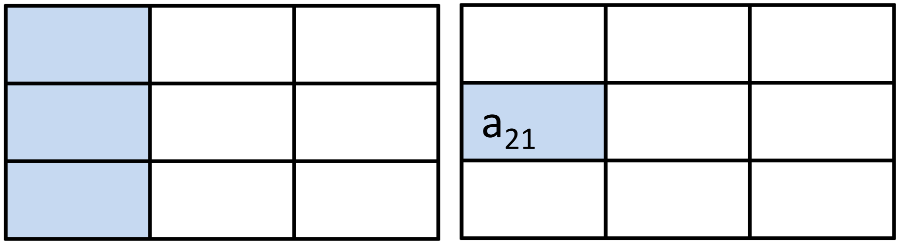


So the first column and the first row of the product matrix will be influence by the a_{11}-entry of both matrices.

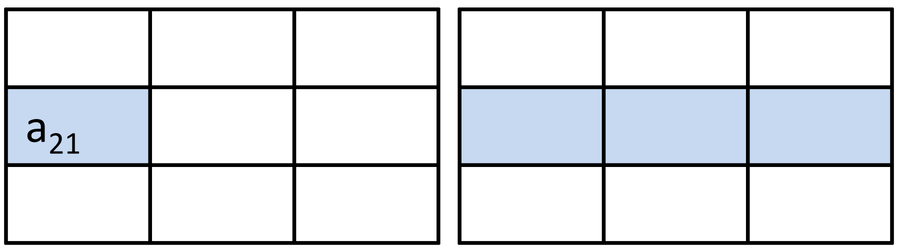




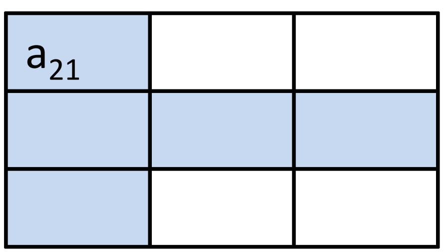
When two matrices multiply, the a_{21}-entry of the later matrix will influence the first column of the former matrix.

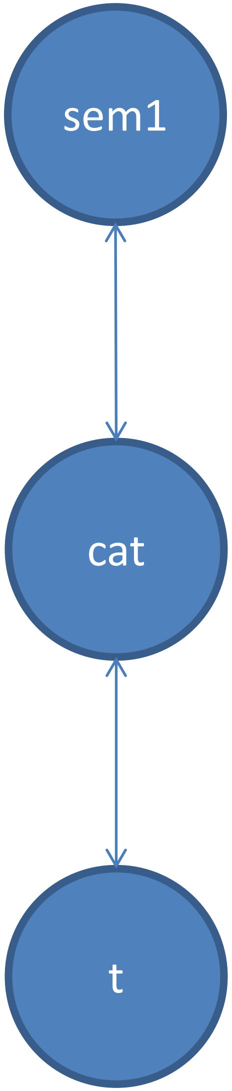


When two matrices multiply, the a_{21}-entry of the former matrix will influence the second row of the later matrix.

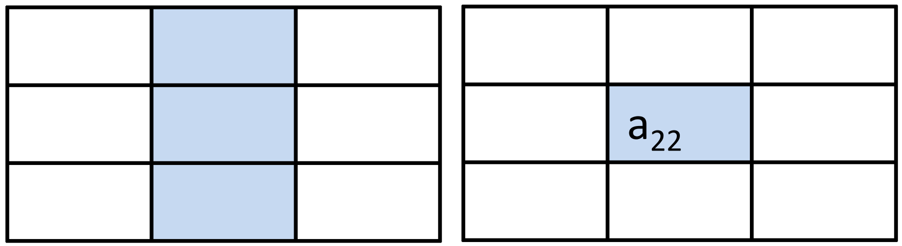


So the first column and the second row of the product matrix will be influence by the a_{21}-entry of both matrices.

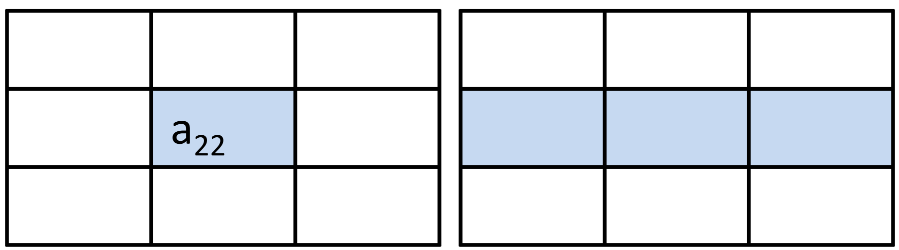




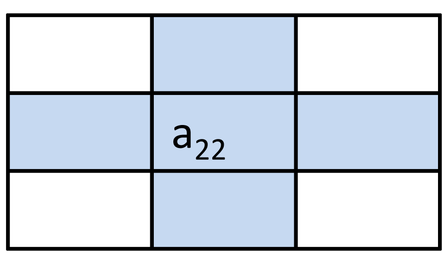
When two matrices multiply, the a_{22}-entry of the later matrix will influence the first column of the former matrix.

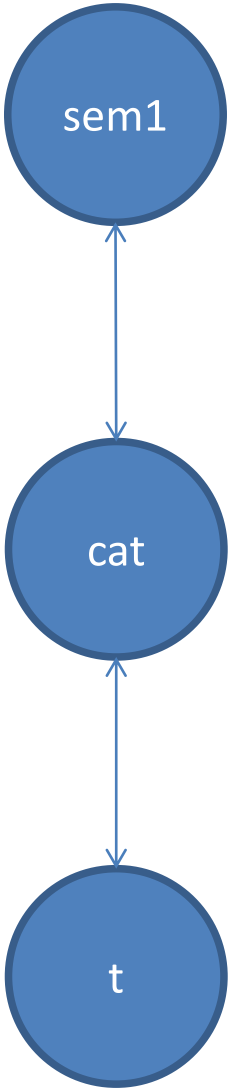


When two matrices multiply, the a_{22}-entry of the former matrix will influence the second row of the later matrix.



So the first column and the second row of the product matrix will be influence by the a_{22}-entry of both matrices.





According to Dell Gordon, the activation of, for example, cat will decay as time step t increases at the rate of q (q is a real number between 0 and 1), so 1-q means the amount of activation that will be preserved from the previous time step of activation.

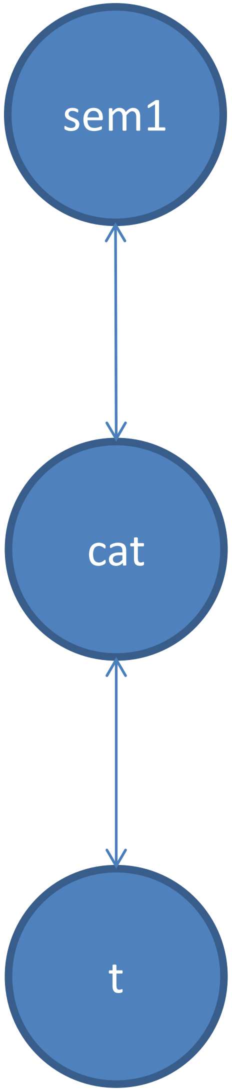
$$(1)\ A(j,t) = A(j,t-1)\,(1-q) + \sum w(i,j)A(i,t-1) + \text{noise}$$

A(j,t) is the activation of unit j at time step t, q is the rate with which activation decays, and w(i,j) is the connection weight between unit i and unit j. During each time step, each unit's activation level is perturbed by normally distributed noise with a mean of zero and a standard deviation proportional to the unit's activation.

The property of matrix multiplication (presented in previous pages) ensures that putting 1-q on the diagonal can actually simulate the decay process happening in activation process. Therefore, we put (1-q) on the diagonal of the matrix.

```
q = .5;
for i=1:20,
   matrix(i,i) = 1-q;
end
```

	sem1	cat	t
sem1	1-q		
cat		1-q	
t			1-q



According to Dell Gordon, the activation of, for example, cat will be influenced by its neighbors: sem1 & t. A part of (multiply by w, which is a real number from 0 to 1) the previous step of activation of sem1 & t will be added into the current step of activation of cat.

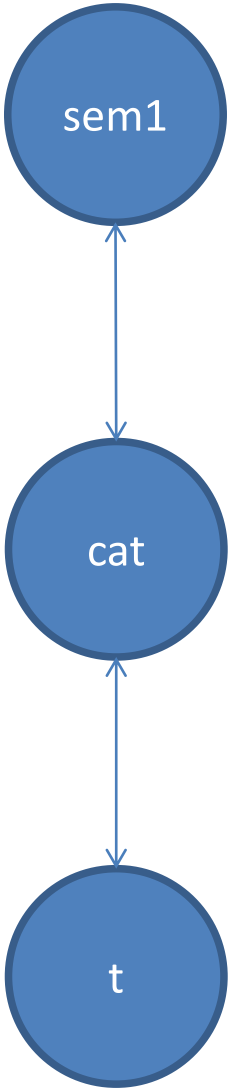
$$(1)\ A(j,t) = A(j,t-1)\,(1-q) + \sum w(i,j)A(i,t-1) + \text{noise}$$

A(j,t) is the activation of unit j at time step t, q is the rate with which activation decays, and w(i,j) is the connection weight between unit i and unit j. During each time step, each unit's activation level is perturbed by normally distributed noise with a mean of zero and a standard deviation proportional to the unit's activation.

Therefore, put w into respective entries.

```
weight=.1;
matrix(sem1,cat)=weight;
matrix(cat,sem1)=weight;
matrix(t,cat)=weight;
matrix(cat,t)=weight;
```

	sem1	cat	t
sem1	1-q	w	
cat	w	1-q	w
t		w	1-q



Now we get the following matrix.

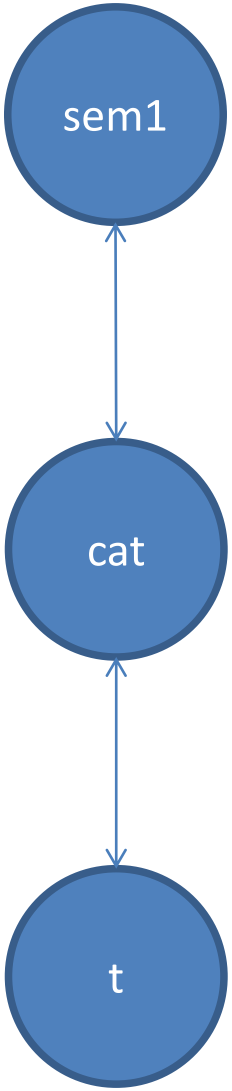
	sem1	cat	t
sem1	1-q	w	
cat	w	1-q	w
t		w	1-q

	sem1	cat	t
sem1	.5	.1	0
cat	.1	.5	.1
t	0	.1	.5

We multiply the matrix by itself for seven times, which can be seen as enduring seven time steps.

```
tempmatrix=matrix^7;
```

	sem1	cat	t
sem1	.015264	.015523	.0074515
cat	.015523	.022716	.015523
t	.0074515	.015523	.015264



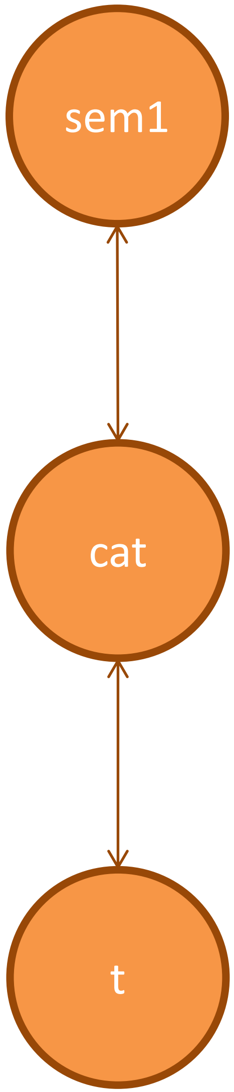
Then we select the first column of the matrix, which means we are going to determine which word is most relevant to sem1; we achieve the process by multiply the matrix by a vector called jolt.

```
jolt=zeros(3,1);
jolt(1)=100;
result=tempmatrix*jolt;
```

	sem1	cat	t		jolt		result
sem1	.015264	.015523	.0074515		100		1.5264
cat	.015523	.022716	.015523	multiply	0	equal	1.5523
t	.0074515	.015523	.015264		0		.74515

The unit of maximum value in the vector, result, is the word to be produce. So in this case cat (with 1.553) is the word to be produced.

```
max(result);
```



But actually the matrix method presented before is not identical to the formula below (for symbol notation about the formula, refer to previous pages).

$$(1)\ A(j,t) = A(j,t\text{-}1)\,(1\text{-}q) + \sum w(i,j)A(i,t\text{-}1) + \text{noise}$$

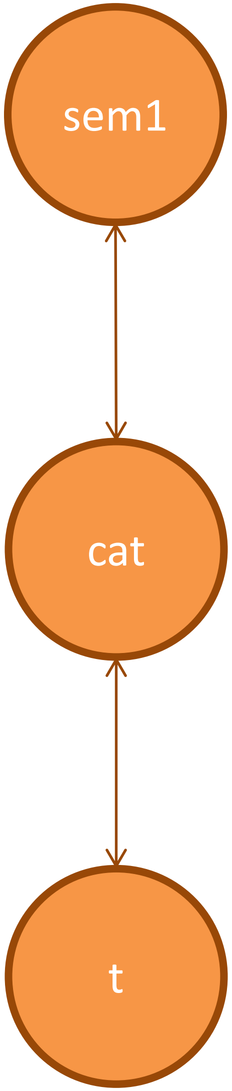
Strictly sticking to the formula(with q=.5, W=.1), the mathematics process should be as follows (we do not discuss noise here).

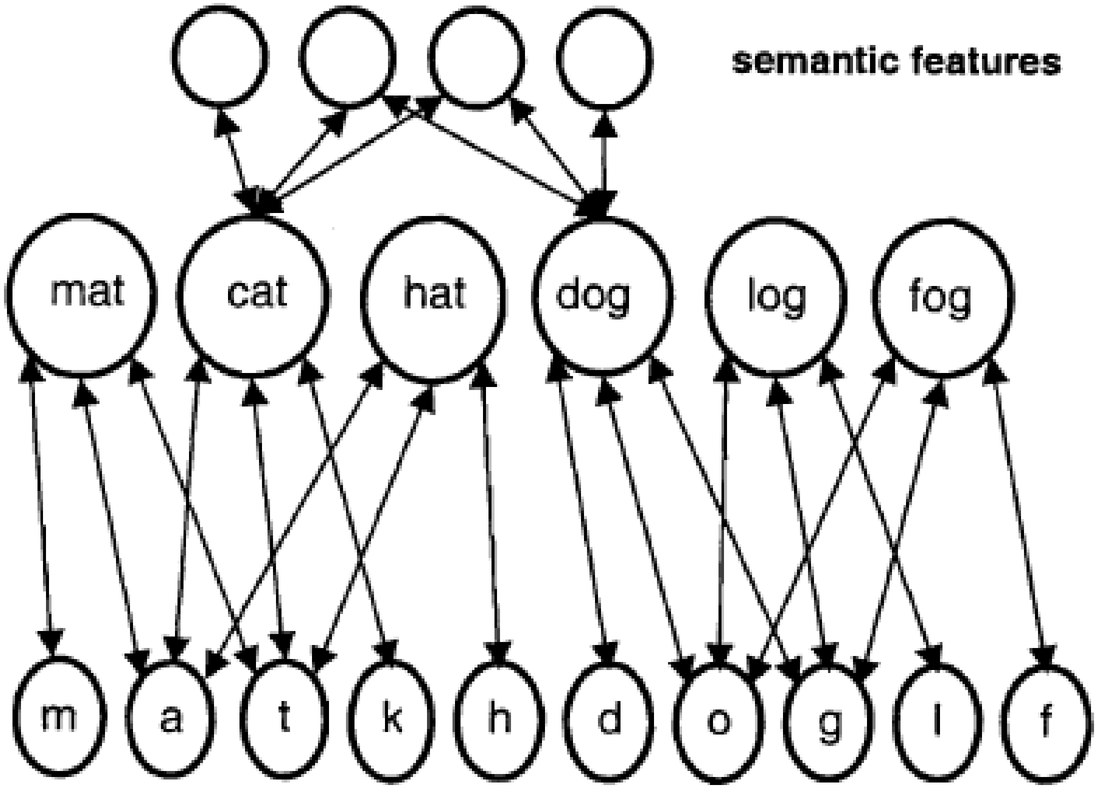
A(1,1)=100
A(2,1)=0
A(3,1)=0

A(1,2)=A(1,1)(1-q)+w(2,1)A(2,1)=100*.5+.1*0=50
A(2,2)=A(2,1)(1-q)+w(1,2)A(1,1)+w(3,2)A(3,1)=0*.5+.1*100+.1*0=10
A(3,2)=A(3,1)(1-q)+w(2,3)A(2,1)=0*.5+.1*0=0

A(1,3)=A(1,2)(1-q)+w(2,1)A(2,2)=50*.5+.1*10=26
A(2,3)=A(2,2)(1-q)+w(1,2)A(1,2)+w(3,2)A(3,2)=10*.5+.1*50+.1*0=10
A(3,3)=A(3,2)(1-q)+w(2,3)A(2,2)=0*.5+.1*10=1

……





Use Java to simulate the process. It shows that the word cat is the most possible word to be produced.

A(1,8)= 5.3302966989576826E-11

A(2,8)= 2.7556665008887645E-10 (cat)

A(3,8)= 9.38192941248417E-11

A(4,8)= 2.363944426178933E-10 (dog)

A(5,8)= 7.146284542977811E-11

A(6,8)= 7.690532691776754E-11