

# Information Extraction

# Document-term Matrix

# Document-Term Matrix

- Bag-of-Words (TF-IDF): Document-Term Matrix

car road gas

truck road

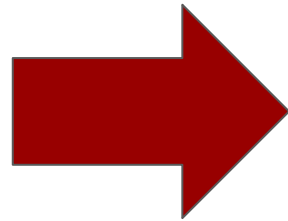
car

gas oil

gas

oil

Toy Corpus: Six Doc.



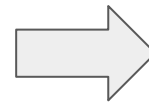
	car	gas	oil	road	truck
0	1	1	0	1	0
1	0	0	0	1	1
2	1	0	0	0	0
3	0	1	1	0	0
4	0	1	0	0	0
5	0	0	1	0	0

# Document-term Matrix

- The shape of C matrix is  $n$  by  $m$
- $m$  is vocab. size
- $n$  is number of documents
- **High-dimensionality**, **Sparse**

# Just Counting No Semantic

	car	gas	oil	road	truck
0	1	1	0	1	0
1	0	0	0	1	1
2	1	0	0	0	0
3	0	1	1	0	0
4	0	1	0	0	0
5	0	0	1	0	0



Cosine  
Similarity is zero

# Feature Selection

- Based on measures such as mutual information, keep the top ranked K features. *choose a subset of the features*

	car	gas	oil	road	truck
0	1	1	0	1	0
1	0	0	0	1	1
2	1	0	0	0	0
3	0	1	1	0	0
4	0	1	0	0	0
5	0	0	1	0	0

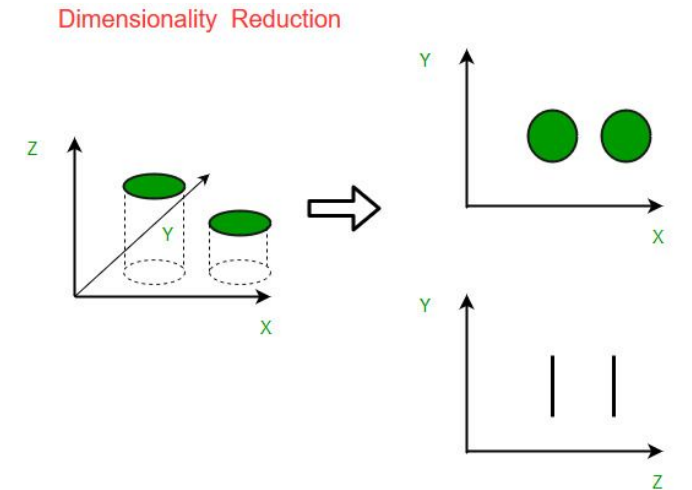


	car	gas
0	1	1
1	0	0
2	1	0
3	0	1
4	0	1
5	0	0

Still Sparse and discard lots of information

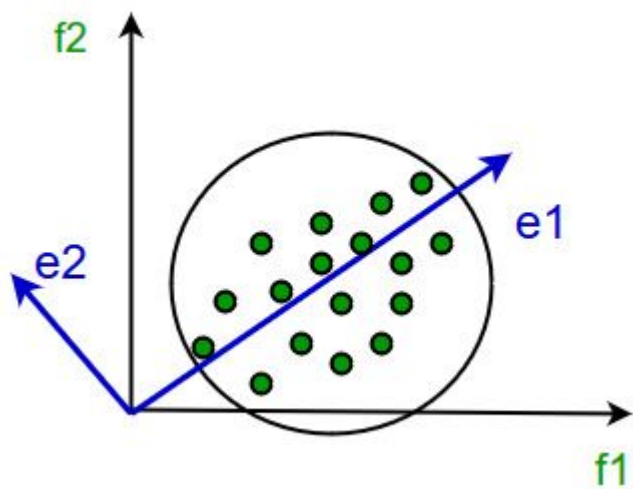
# Dimensionality Reduction

- New features will be learned by combining old features
- This is:  $\mathbb{R}^M \rightarrow \mathbb{R}^d$  and  $d < m$
- Algorithms:
  - PCA
  - NMF
  - Auto-encoder
  - T-sNE
  - Kernel PCA
  - Manifold learning
  - ....

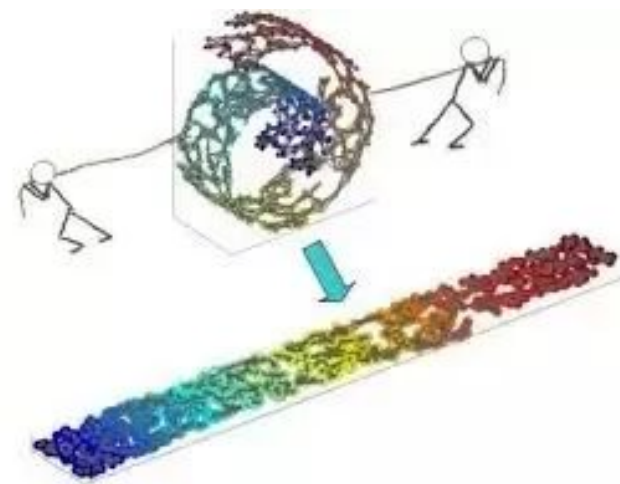


<https://www.geeksforgeeks.org/dimensionality-reduction/>

# Linear vs Nonlinear



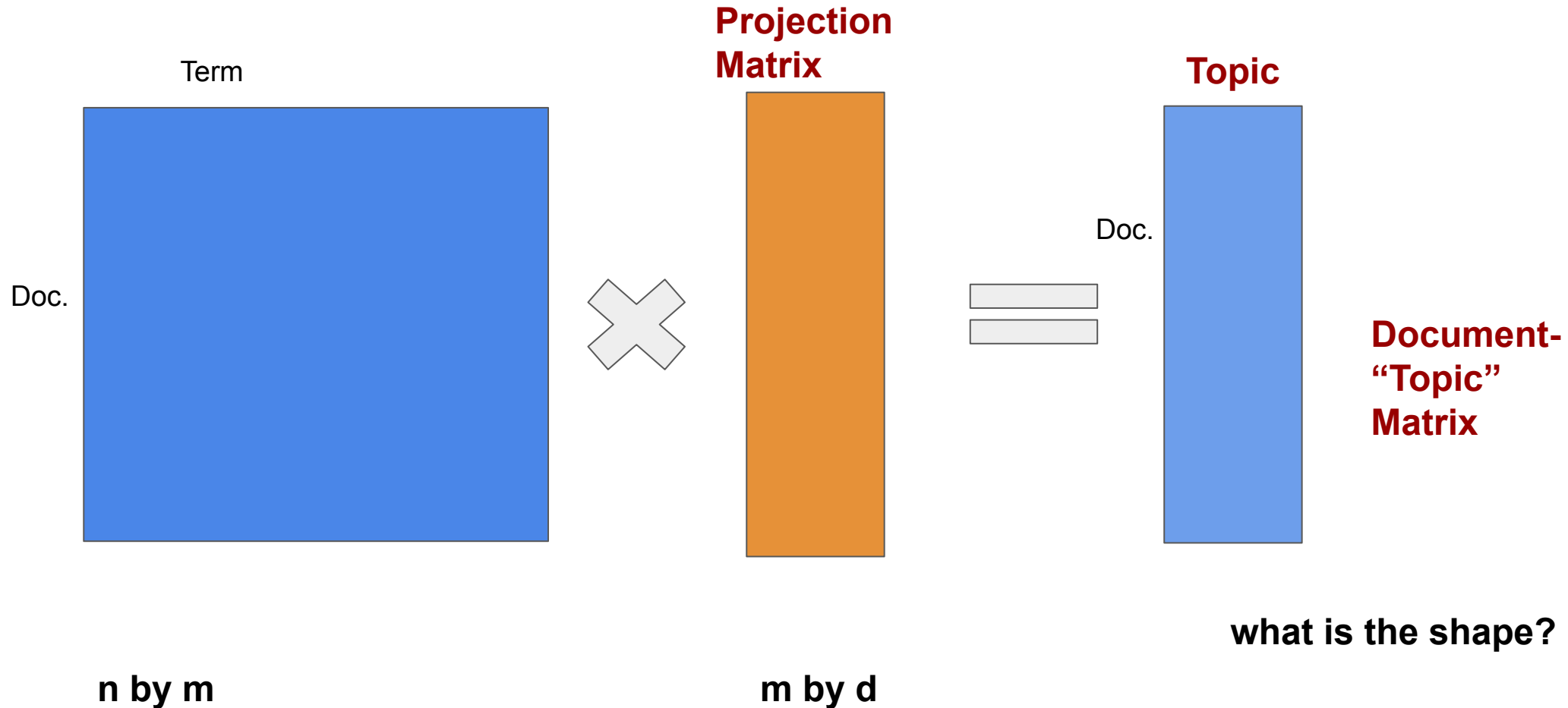
PCA



Manifold Learning (From Quora)



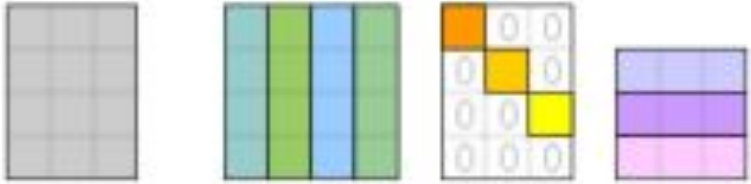
# Linear Projection for Term Document Matrix




# How to find project matrix

- It is also called matrix decomposition in linear algebra
- **Latent Semantic Analysis: SVD**
- Non-negative Matrix Factorization
- .....


# Full SVD



$$\mathbf{M}_{m \times n} = \mathbf{U}_{m \times m} \mathbf{\Sigma}_{m \times n} \mathbf{V}^*_{n \times n}$$



$$\mathbf{U} \mathbf{U}^* = \mathbf{I}_m$$



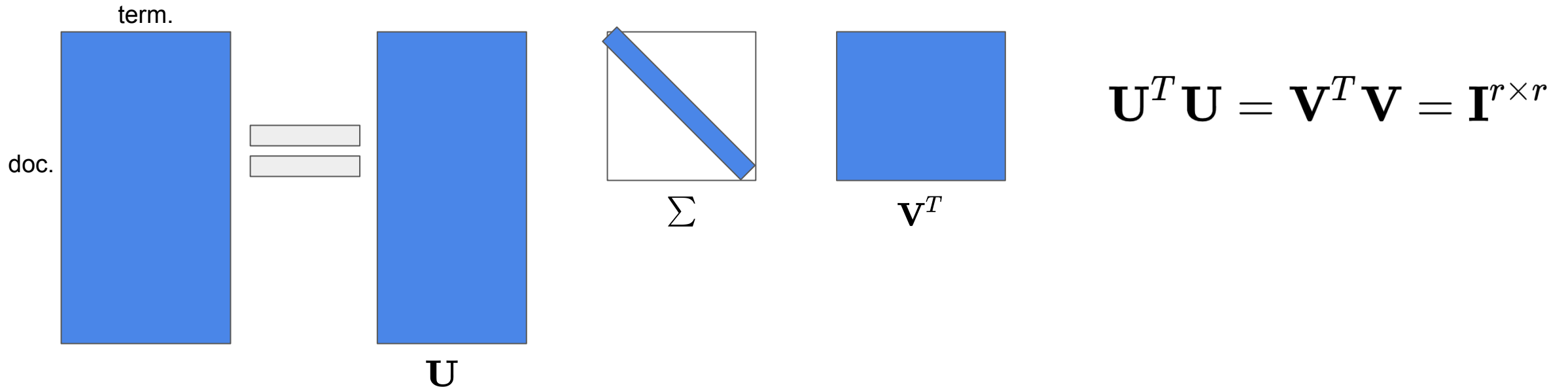
$$\mathbf{V} \mathbf{V}^* = \mathbf{I}_n$$

from wiki

How about  $m < n$ ?

# Reduced SVD

- SVD decomposes the matrix ( n by m matrix) into 3 parts  $\mathbf{C} = \mathbf{U} * \Sigma * \mathbf{V}^T$ 
  - $\mathbf{U}$  is a matrix of size n by r
  - $\Sigma$  is a diagonal matrix whose diagonal entries are known as the singular values and the size is r by r
  - $\mathbf{V}^T$  is a matrix of size of r by m
  - r is the rank of the matrix, which is usually the min value of m and n



# Look at the previous example

	car	gas	oil	road	truck
0	1	1	0	1	0
1	0	0	0	1	1
2	1	0	0	0	0
3	0	1	1	0	0
4	0	1	0	0	0
5	0	0	1	0	0



	topic1	topic2	topic3	topic4	topic5
0	-0.748623	0.286454	-0.279712	-1.703299e-17	0.528459
1	-0.279712	0.528459	0.748623	-6.485281e-16	-0.286454
2	-0.203629	0.185761	-0.446563	5.773503e-01	-0.625521
3	-0.446563	-0.625521	0.203629	5.088081e-16	-0.185761
4	-0.325096	-0.219880	-0.121467	-5.773503e-01	-0.405641
5	-0.121467	-0.405641	0.325096	5.773503e-01	0.219880



	topic1	topic2	topic3	topic4	topic5
topic1	2.162501	0.000000	0.000000	0.0	0.000000
topic2	0.000000	1.594382	0.000000	0.0	0.000000
topic3	0.000000	0.000000	1.27529	0.0	0.000000
topic4	0.000000	0.000000	0.000000	1.0	0.000000
topic5	0.000000	0.000000	0.000000	0.0	0.393915



	car	gas	oil	road	truck
topic1	-0.440347	-0.703020	-0.262673	-4.755303e-01	-1.293463e-01
topic2	0.296174	-0.350572	-0.646747	5.111152e-01	3.314507e-01
topic3	-0.569498	-0.154906	0.414592	3.676900e-01	5.870217e-01
topic4	0.577350	-0.577350	0.577350	-2.493650e-16	-6.963379e-16
topic5	-0.246402	-0.159788	0.086614	6.143584e-01	-7.271970e-01

representation  
of documents

importance of the semantic  
dimensions

representation  
of terms

# New Representation for Documents

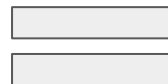
	topic1	topic2	topic3	topic4	topic5
0	-0.748623	0.286454	-0.279712	-1.703299e-17	0.528459
1	-0.279712	0.528459	0.748623	-6.485281e-16	-0.286454
2	-0.203629	0.185761	-0.446563	5.773503e-01	-0.625521
3	-0.446563	-0.625521	0.203629	5.088081e-16	-0.185761
4	-0.325096	-0.219880	-0.121467	-5.773503e-01	-0.405641
5	-0.121467	-0.405641	0.325096	5.773503e-01	0.219880

**representation  
of documents**



	topic1	topic2	topic3	topic4	topic5
topic1	2.162501	0.000000	0.000000	0.0	0.000000
topic2	0.000000	1.594382	0.000000	0.0	0.000000
topic3	0.000000	0.000000	1.27529	0.0	0.000000
topic4	0.000000	0.000000	0.000000	1.0	0.000000
topic5	0.000000	0.000000	0.000000	0.0	0.393915

**importance of the semantic  
dimensions**



	topic1	topic2	topic3	topic4	topic5
0	-1.618898	0.456717	-0.356713	-1.703299e-17	0.208168
1	-0.604877	0.842566	0.954712	-6.485281e-16	-0.112839
2	-0.440347	0.296174	-0.569498	5.773503e-01	-0.246402
3	-0.965693	-0.997319	0.259686	5.088081e-16	-0.073174
4	-0.703020	-0.350572	-0.154906	-5.773503e-01	-0.159788
5	-0.262673	-0.646747	0.414592	5.773503e-01	0.086614

**final representation of  
documents  $U \Sigma$**

Vectors for documents are dense in the new learned topic space. However, the similarity between doc4 and doc5 are still tiny

```
] print(cosine_similarity(dense_matrix[4].reshape(1, -1), dense_matrix[5].reshape(1, -1))  
[[-6.10622664e-16]]
```

# Projection Matrix is V

	car	gas	oil	road	truck
topic1	-0.440347	-0.703020	-0.262673	-4.755303e-01	-1.293463e-01
topic2	0.296174	-0.350572	-0.646747	5.111152e-01	3.314507e-01
topic3	-0.569498	-0.154906	0.414592	3.676900e-01	5.870217e-01
topic4	0.577350	-0.577350	0.577350	-2.493650e-16	-6.963379e-16
topic5	-0.246402	-0.159788	0.086614	6.143584e-01	-7.271970e-01

$V^T$

	topic1	topic2	topic3	topic4	topic5
car	-0.440347	0.296174	-0.569498	5.773503e-01	-0.246402
gas	-0.703020	-0.350572	-0.154906	-5.773503e-01	-0.159788
oil	-0.262673	-0.646747	0.414592	5.773503e-01	0.086614
road	-0.475530	0.511115	0.367690	-2.493650e-16	0.614358
truck	-0.129346	0.331451	0.587022	-6.963379e-16	-0.727197

$V$

$$C = U * \Sigma * V^T \quad \rightarrow \quad C * \underset{\substack{\text{projection matrix}}}{V} = U * \Sigma$$

Topic 1 = -0.44 \* **car** - 0.70 \* **gas** - 0.26 \* **oil** - 0.47 \* **road** - 0.13 \* **truck**

Each topic is regarded as the **linear** combination of words

# For a new document

car	gas	oil	road	truck
0	0	1	1	1



	topic1	topic2	topic3	topic4	topic5
car	-0.440347	0.296174	-0.569498	5.773503e-01	-0.246402
gas	-0.703020	-0.350572	-0.154906	-5.773503e-01	-0.159788
oil	-0.262673	-0.646747	0.414592	5.773503e-01	0.086614
road	-0.475530	0.511115	0.367690	-2.493650e-16	0.614358
truck	-0.129346	0.331451	0.587022	-6.963379e-16	-0.727197



topic1	topic2	topic3	topic4	topic5
-0.867549	0.195819	1.369303	0.57735	-0.026225

V

Sparse



Dense



# How to reduce dimensionality

**Abandon unimportant topics**

# Reduce Dimensionality

- Each singular value in  $\Sigma$  tells us how important its dimension is.
- **By setting less important dimensions to zero, we keep the important information, but get rid of the details**
- The details may
  - be noise - in that case, reduced LSI is better representation because it is less noisy
  - make things dissimilar that should be similar - again, the reduced LSI representation is a better representation because it represents similarity better.

# Truncated SVD

- Zeroing out but these two largest singular values

	topic1	topic2	topic3	topic4	topic5
topic1	2.162501	0.000000	0.000000	0.0	0.000000
topic2	0.000000	1.594382	0.000000	0.0	0.000000
topic3	0.000000	0.000000	1.27529	0.0	0.000000
topic4	0.000000	0.000000	0.000000	1.0	0.000000
topic5	0.000000	0.000000	0.000000	0.0	0.393915



	topic1	topic2	topic3	topic4	topic5
topic1	2.162501	0.000000	0.0	0.0	0.0
topic2	0.000000	1.594382	0.0	0.0	0.0
topic3	0.000000	0.000000	0.0	0.0	0.0
topic4	0.000000	0.000000	0.0	0.0	0.0
topic5	0.000000	0.000000	0.0	0.0	0.0

# New Representation for Documents in 2 dimensions

	topic1	topic2	topic3	topic4	topic5
0	-0.748623	0.286454	-0.279712	-1.703299e-17	0.528459
1	-0.279712	0.528459	0.748623	-6.485361e-16	-0.286454
2	-0.203629	0.185761	-0.446563	5.773503e-01	-0.625521
3	-0.446563	-0.625521	0.203629	5.088081e-16	-0.185761
4	-0.325096	-0.219880	-0.121467	-5.773503e-01	-0.405641
5	-0.121467	-0.405641	0.325096	5.773503e-01	0.219880

representation  
of documents

$U_d$



	topic1	topic2	topic3	topic4	topic5
topic1	2.162501	0.000000	0.0	0.0	0.0
topic2	0.000000	1.594382	0.0	0.0	0.0
topic3	0.000000	0.000000	0.0	0.0	0.0
topic4	0.000000	0.000000	0.0	0.0	0.0
topic5	0.000000	0.000000	0.0	0.0	0.0

importance of the semantic  
dimensions

$\sum_d$



	topic1	topic2	topic3	topic4	topic5
0	-1.618898	0.456717	-0.0	-0.0	0.0
1	-0.604877	0.842566	0.0	0.0	-0.0
2	-0.440347	0.296174	-0.0	0.0	-0.0
3	-0.965693	-0.997319	0.0	0.0	-0.0
4	-0.703020	-0.350572	-0.0	-0.0	-0.0
5	-0.262673	-0.646747	0.0	0.0	0.0

The new feature space is  
2

Now, we can compute the similarity for doc  
4 and doc 5

```
# compute the new similarity
print(cosine_similarity(lowdim_dense_matrix[4].reshape(1, -1), lowdim_dense_matrix[5].reshape(1, -1)))

[[0.75020516]]
```

# For a new document

						topic1	topic2	topic3	topic4	topic5	
						car	-0.440347	0.296174	-0.569498	5.773503e-01	-0.246402
						gas	-0.703020	-0.350572	-0.154906	-5.773503e-01	-0.159788
						oil	-0.262673	-0.646747	0.414592	5.773503e-01	0.086614
						road	-0.475530	0.511115	0.367690	-2.493650e-16	0.614358
						truck	-0.129346	0.331451	0.587022	-6.963379e-16	-0.727197
car	gas	oil	road	truck							
0	0	1	1	1							

			topic1	topic2
0	-0.867549	0.195819		

$$\mathbf{V}_d = \mathbf{V}[:, 0:d]$$

**Here,  $d$  is less than the original matrix rank.**

# Why we use LSA as text vectors

- LSA try to capture semantic information (talk about the same topic)
  - do not need documents use same words
  - project doc in a reduced vector space
- Try to addresses the linguistic characteristic: **synonymy** and **semantic relatedness**.

# How LSA addresses synonymy and semantic relatedness

- The dimensionality reduction forces us to ignore a lot of details.
- We try to map different words (original vector space) to the same dimension in the reduced space.
- The “cost” of mapping synonyms to the same dimensions is much less than the cost of collapsing unrelated words.
- SVD selects the “least costly” mapping. (Eckart-Young theorem)

	topic1	topic2	topic3	topic4	topic5
<b>car</b>	-0.440347	0.296174	-0.569498	5.773503e-01	-0.246402
<b>gas</b>	-0.703020	-0.350572	-0.154906	-5.773503e-01	-0.159788
<b>oil</b>	-0.262673	-0.646747	0.414592	5.773503e-01	0.086614
<b>road</b>	-0.475530	0.511115	0.367690	-2.493650e-16	0.614358
<b>truck</b>	-0.129346	0.331451	0.587022	-6.963379e-16	-0.727197

# Implementation

- Given a corpus, get the document-term matrix
- Compute SVD of the matrix  $\mathbf{C} = \mathbf{U} * \Sigma * \mathbf{V}^T$
- The original corpus are represented in the reduced space (dim is d):

$$\mathbf{U}_d \Sigma_d$$

- The new documents can be firstly transformed in the original vector space  $\mathbf{q}$
- Map them into the reduced space  $\mathbf{qV}_d$



# Other approaches for document representation

- Other Matrix Decomposition methods:
  - NMF
- Probabilistic Model:
  - Topic Model: Latent Dirichlet Allocation
- Deep learning based Model:
  - RNN, CNN

# Information Extraction

# Goals of Information Extraction

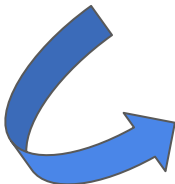
- An important research area for natural language processing and text mining is **the extraction and formatting of information from unstructured text**.
- Computers can be used to sift through a large amount of text and extract restricted forms of useful information, which can be **represented in a tabular format**.
- Information extraction can be regarded as a restricted form of full natural language understanding, where we know in advance what kind of **semantic information** we are looking for.
- The main task is then to extract parts of text to **fill in slots in a predefined template**.

# Goals of Information Extraction

- A task defined as **executive position changes**:

One of the many differences between *Robert L. James, chairman and chief executive officer of McCann-Erickson*, and *John J. Dooner, Jr.*, the agency's president and chief operating officer, is quite telling: Mr. James enjoys sailboating, while Mr. Dooner owns a powerboat.

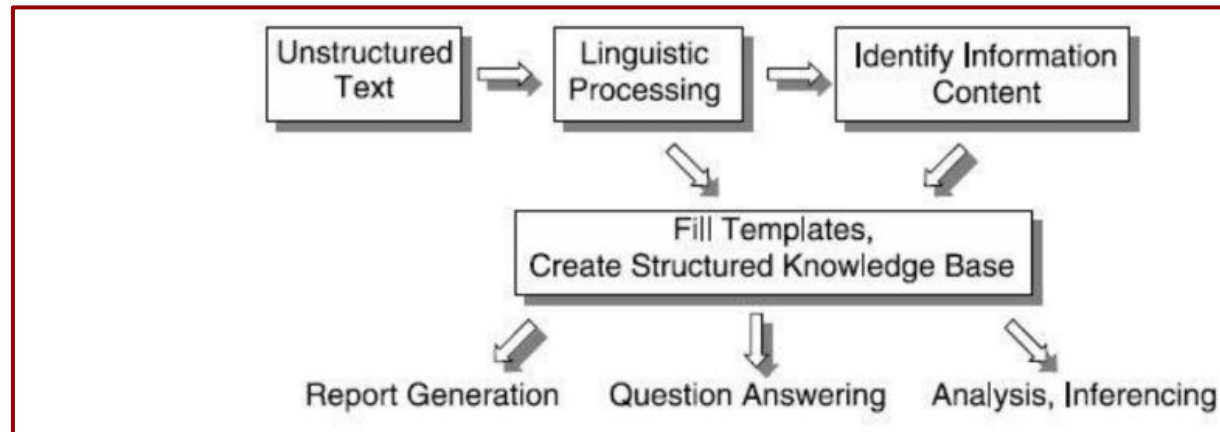
Now, Mr. James is preparing to sail into the sunset, and Mr. Dooner is poised to rev up the engines to guide *Interpublic Group's McCann-Erickson* into the 21st century. Yesterday, *McCann* made official what had been widely anticipated: *Mr. James, 57 years old*, is stepping down as chief executive officer on *July 1* and will retire as chairman at the *end of the year*. He will be succeeded by *Mr. Dooner, 45 ...*



Predefined Domain	Extracted Information
Organization	<i>McCann-Erickson</i>
Position	<i>Chief executive officer</i>
Date	<i>July 1</i>
Outgoing person name	<i>Robert L. James</i>
Outgoing person age	<i>57</i>
Incoming person name	<i>John J. Dooner, Jr.</i>
Incoming person age	<i>45</i>

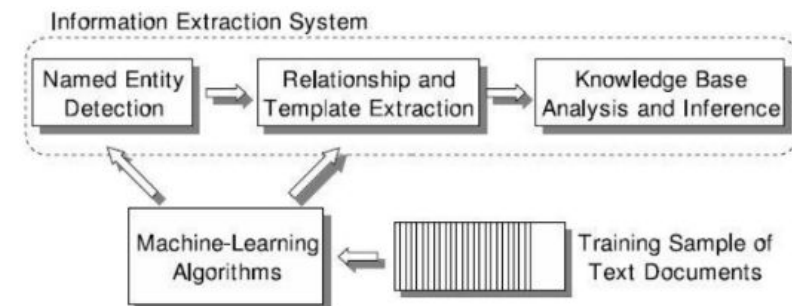
# Goals of Information Extraction

- A general information extraction system is illustrated in the blow figure.
- The task of information extraction naturally decomposes into a sequence of processing steps, typically including **tokenization**, **sentence segmentation**, **part-of-speech assignment**, **named entity identification**, **phrasal parsing**, **sentential parsing**, **semantic interpretation**, **template filling**, and **merging**.



# Goals of Information Extraction

- The most accurate information extraction systems often involve human effort: handcrafted language processing modules.
  - People's names have prefixes such as Mr., Mrs., Miss., Dr., Jr.,
  - People's names are recognized by phrases such as “according to..” or “...said”
- The application of machine-learning techniques to information extraction is motivated by the time-consuming process needed to handcraft these systems.
- The general architecture of a machine-learning-based information extraction system is given as below:



# Goals of Information Extraction

- There are typically two main modules involved in such a system.
- The purpose of the first module is to annotate the text document and find portions of the text that interest us (**Name Entity extraction**)
  - For example, we want to identify the string *Robert L. James* as a **person** and the string *McCann-Erickson* as an **organization**. human effort: handcrafted language processing modules.
- Once such entity mentions are extracted, another module is invoked to extract high-level information based on the entity mentions (**Relationship extraction**).
  - In the example of Fig. 6.1, we want to identify that the person *Robert L. James* **belongs to** the organization *McCann-Erickson*, and his age is **57**.
- The information is then filled into slots of a predefined template.

# Example of Information Extraction

- As a task: Filling slots in a database (template) from corpus

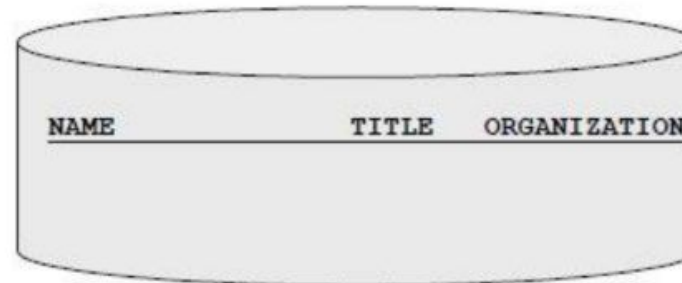
October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying...





# Example of Information Extraction

- As a task: Filling slots in a database (template) from corpus

October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying...



NAME	TITLE	ORGANIZATION
Bill Gates	CEO	Microsoft
Bill Veghte	VP	Microsoft
Richard Stallman	founder	Free Soft..

# Example of Information Extraction

- As a family of techniques:

**Information Extraction = Segmentation + Classification + Association + Clustering**

October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying...

**Microsoft Corporation**

**CEO**

**Bill Gates**

**Microsoft**

**Gates**

**Microsoft**

**Bill Veghte**

**Microsoft**

**VP**

**Richard Stallman**

**founder**

**Free Software Foundation**

aka "named entity  
extraction"

# Example of Information Extraction

- As a family of techniques:

Information Extraction = Segmentation + Classification + Association + Clustering

October 14, 2002, 4:00 a.m. PT

For years, **Microsoft Corporation** **CEO** **Bill Gates** railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, **Microsoft** claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. **Gates** himself says **Microsoft** will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said **Bill Veghte**, a **Microsoft** **VP**. "That's a super-important shift for us in terms of code access."

**Richard Stallman**, **founder** of the **Free Software Foundation**, countered saying...

**Microsoft Corporation**  
**CEO**  
**Bill Gates**  
**Microsoft**  
**Gates**  
**Microsoft**  
**Bill Veghte**  
**Microsoft**  
**VP**  
**Richard Stallman**  
**founder**  
**Free Software Foundation**

# Example of Information Extraction

- As a family of techniques:

Information Extraction = Segmentation + Classification + Association + Clustering

October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying...

Microsoft Corporation  
CEO  
Bill Gates

Microsoft  
Gates

Microsoft  
Bill Veghte  
Microsoft  
VP

Richard Stallman  
founder  
Free Software Foundation



# Example of Information Extraction

- As a family of techniques:

Information Extraction = Segmentation + Classification + Association + Clustering

October 14, 2002, 4:00 a.m. PT

For years, [Microsoft Corporation](#) [CEO](#) [Bill Gates](#) railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, [Microsoft](#) claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. [Gates](#) himself says [Microsoft](#) will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said [Bill Veghte](#), a [Microsoft](#) [VP](#). "That's a super-important shift for us in terms of code access."

[Richard Stallman](#), [founder](#) of the [Free Software Foundation](#), countered saying...

*	<a href="#">Microsoft Corporation</a> <a href="#">CEO</a> <a href="#">Bill Gates</a>
*	<a href="#">Microsoft</a> <a href="#">Gates</a>
*	<a href="#">Microsoft</a> <a href="#">Bill Veghte</a>
*	<a href="#">Microsoft</a> <a href="#">VP</a>
	<a href="#">Richard Stallman</a> <a href="#">founder</a> <a href="#">Free Software Foundation</a>

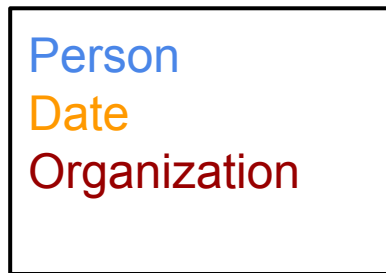
NAME	TITLE	ORGANIZATION
Bill Gates	CEO	Microsoft
Bill Veghte	VP	Microsoft
Richard Stallman	founder	Free Soft...

# Named Entity Recognition

# Named Entity Recognition (NER)

- A very important sub-task: find and classify names in text for example:

The decision by the independent MP **Andrew Wilkie** to withdraw his support for the minority **Labor** government sounded dramatic but it should not further threaten its stability. When, after the **2010** election, **Wilkie**, **Rob Oakeshott**, **Tony Windsor** and the **Greens** agreed to support **Labor**, they gave just two guarantees: confidence and supply.



# Goals of Information Extraction

- The uses:
  - Named entities can be indexed, linked off, etc.
  - A lot of IE relations are associations between named entities
  - For question answering, answers are often named entities.
  - Sentiment can be attributed to companies or products.
- Concretely:
  - Many web pages tag various entities, with links to bio or topic pages, etc.
    - Reuters' OpenCalais, AlchemyAPI, Yahoo's Term Extraction,...
  - Microsoft: smart recognizers for document content
    - E.g., recognize a name, can take actions, such as add to contacts and open contacts.



# The NER Task

- Task: Predict entities in a text

Foreign	ORG	
Ministry	ORG	
spokesman	O	
Shen	PER	} Standard evaluation is per entity, <i>not</i> per token
Guofang	PER	
told	O	
Reuters	ORG	
:	:	

# Precision/Recall/F1 for IE/NER

- Recall and precision are straightforward for tasks like text categorization.
- The measure is a bit different for IE/NER when there are *boundary errors* (which are common):
  - First Bank of Chicago announced earnings.....
- This counts both a false positive and a false negative
- Select **nothing** would have been better.
- Some other metrics (e.g., MUC scorer) give partial credit (according to complex rules)

# Sequence Problems

- In document, each sentence or phrase contains a sequence of words.
- We can label each item in a sequence for **name entity recognition**.
  - A sequence classifier or sequence labeler is a model whose job is to assign some label or class to each unit.

PERS	O	O	O	ORG	ORG
Murdoch	discusses	future	of	News	Corp.

**Named entity recognition**

VBG	NN	IN	DT	NN	IN	NN
Chasing	opportunity	in	an	age	of	upheaval

**POS tagging**

# The ML sequence model approach to NER

- **Training**

- Collect
- Label each token for its entity class or other (O)
- Design feature extractors appropriate to the text and classes
- Train a sequence classifier to predict the labels from the data

- **Testing**

- Receive a set of testing documents
- Run sequence model inference to label each token
- Appropriately output the recognized entities

# Encoding Classes for Sequence Labeling

	IO encoding	IOB encoding (short for Inside, Outside, Beginning)
Fred	PER	B-PER
showed	O	O
Sue	PER	B-PER
Mengqiu	PER	I-PER
Huang	PER	I-PER
's	O	O
new	O	O
painting	O	O

Which encoding method need more training data?

# Features for sequence labelling

- Words
  - Current word
  - Previous/next word (context)
- Other kinds of inferred linguistic classification
  - Part-of-speech tags
- Label context
  - Previous label

**Input (current word: London):** *Thousands of demonstrators have marched **through London to** protest the war in Iraq and demand the withdrawal of British troops from that country.*

**Label:** ....., ('through', 'O'), ('London', 'B-geo'), ('to', 'O'), .....

Features for the word **London**

```
{'bias': 1.0,  
 'word.lower()': 'london',  
 'word[-3:]': 'don',  
 'word[-2:]': 'on',  
 'word.isupper()': False,  
 'word.istitle()': True,  
 'word.isdigit()': False,  
 'postag': 'NNP',  
 'postag[:2]': 'NN',  
 '-1:word.lower()': 'through',  
 '-1:word.istitle()': False,  
 '-1:word.isupper()': False,  
 '-1:postag': 'IN',  
 '-1:postag[:2]': 'IN',  
 '+1:word.lower()': 'to',  
 '+1:word.istitle()': False,  
 '+1:word.isupper()': False,  
 '+1:postag': 'TO',  
 '+1:postag[:2]': 'TO'},
```

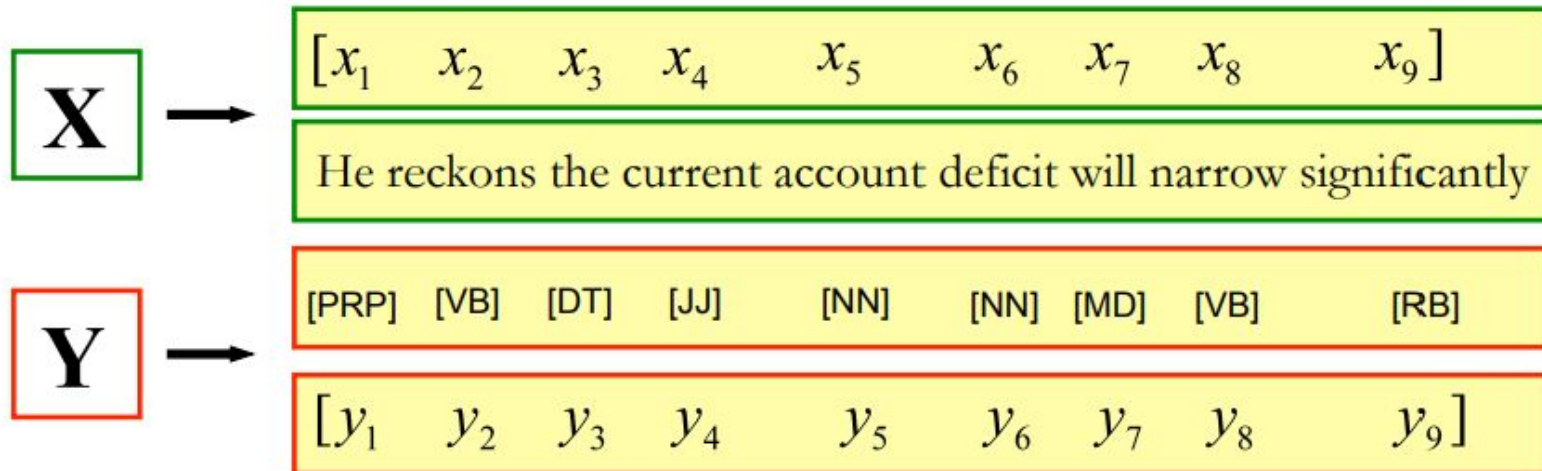
# Sequence Labelling

- A widely used algorithm for sequence labelling
- Finds the most probable label sequence  $\mathbf{y}$  given an observation sequence  $\mathbf{x}$

$$\mathbf{y} = \operatorname{argmax}_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$$

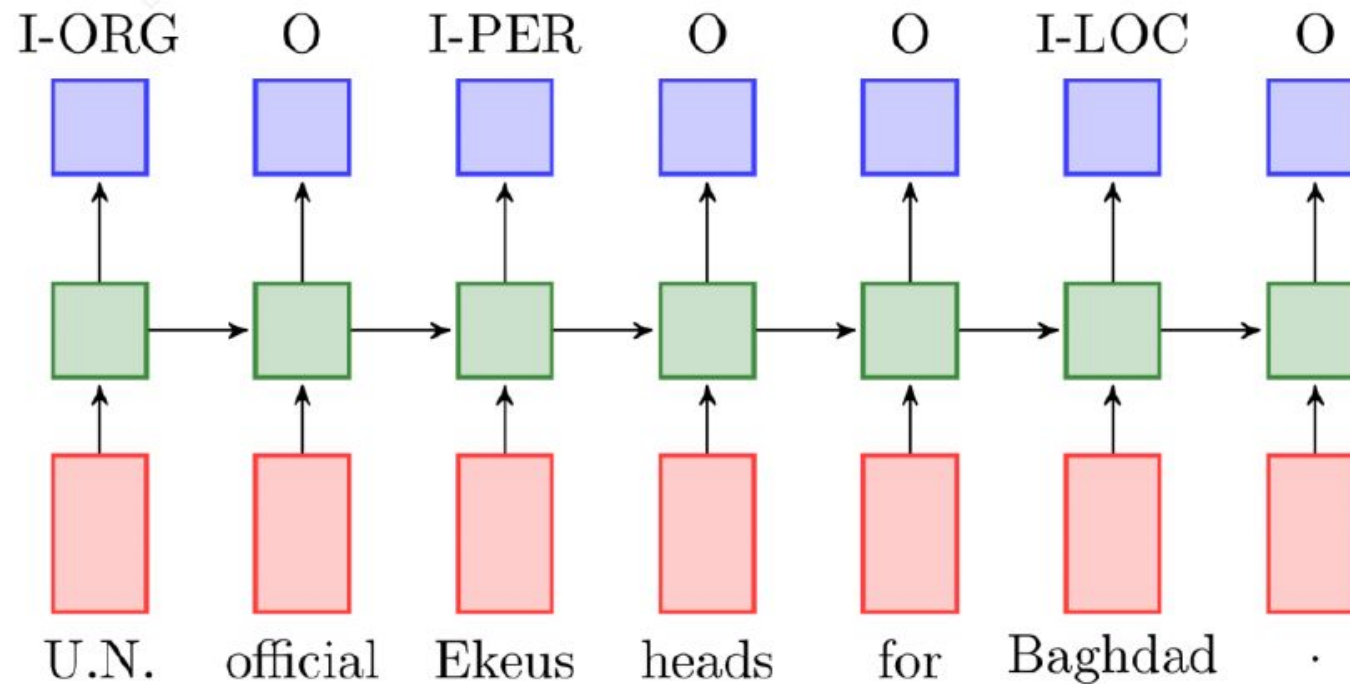
- Where  $\mathbf{x}$  consists of the sequence of tokens from input text.

Example: Part-of-Speech Tagging



# Sequential Model

- Hidden Markov Model
- Conditional Random Field
- RNN



From [Guide to Sequence Tagging with Neural Network in Python](#)