PATTERN RECOGNITION USING PYTHON

Text Data Mining

Wen-Yen Hsu
Dept Electrical Engineering
Chang Gung University, Taiwan

2019-Spring

Applying Machine Learning to Text Data Mining

- Building feature vectors from text documents
- Inferring topics from document collections for categorization

Knowledge Discovery in Databases (KDD) (Fayyad et al., 1996)

- Extracting useful information (knowledge) from the rapidly growing volumes of digital data

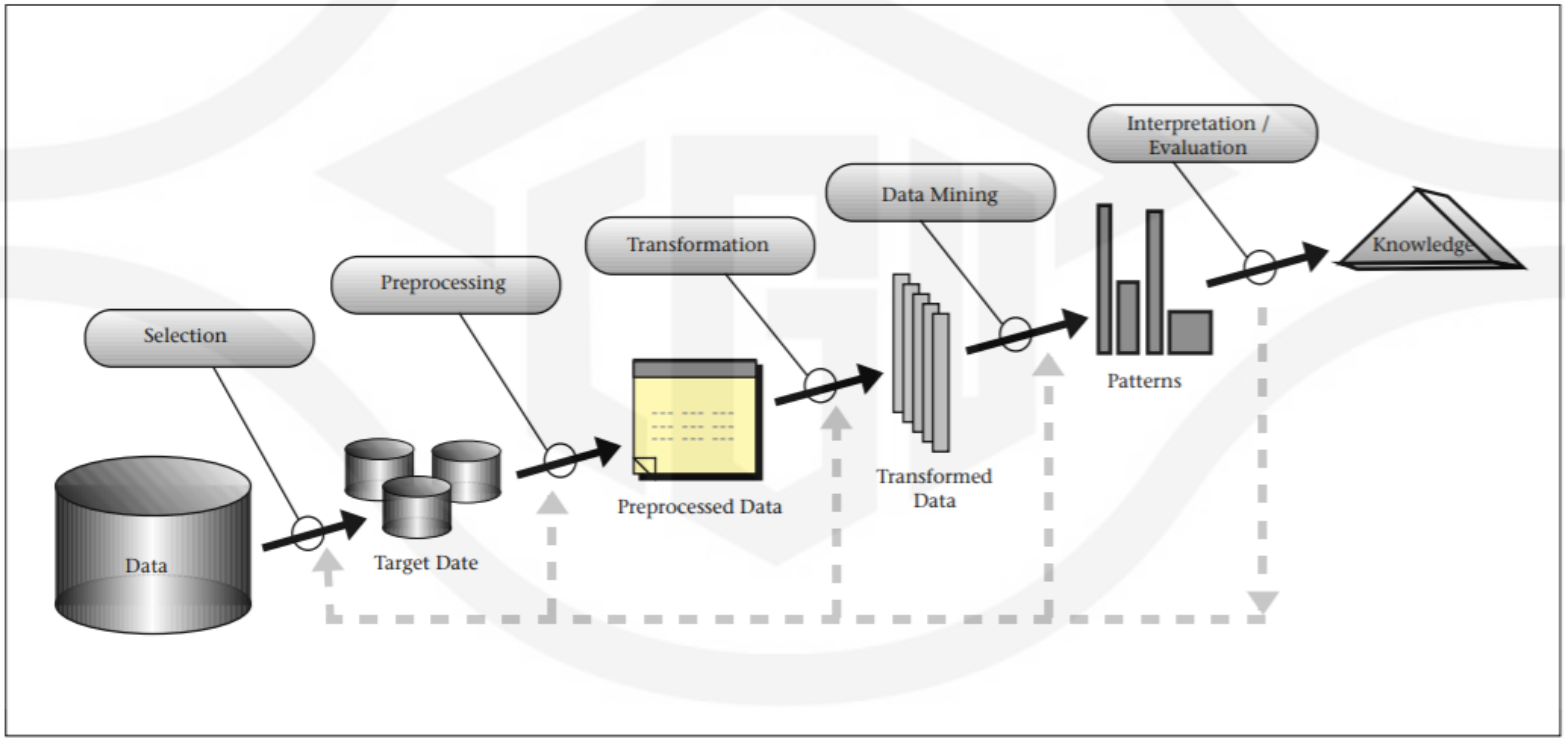


Figure 1. An Overview of the Steps That Compose the KDD Process.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996a). From data mining to knowledge discovery in databases. AI Magazine, 17(3), 37-54.

Data Mining (Fayyad et al., 1996)

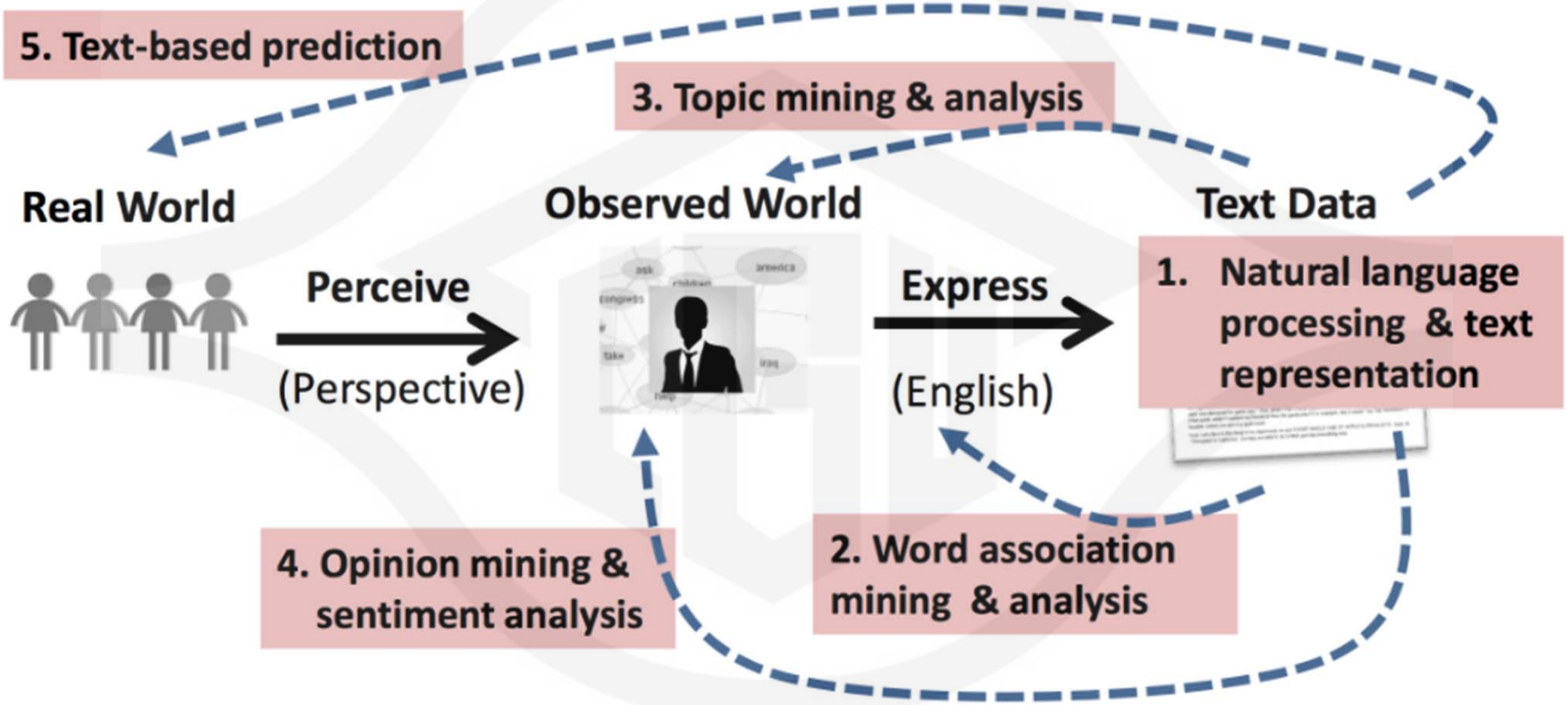
- Verification
  - Verifying the user's hypothesis
- Discovery
  - Prediction (Supervised)
    - **Classification**
    - Ranking
    - Regression
  - Description (Unsupervised)
    - **Clustering**
    - Association rules
    - Summarization

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996a). From data mining to knowledge discovery in databases. AI Magazine, 17(3), 37-54.

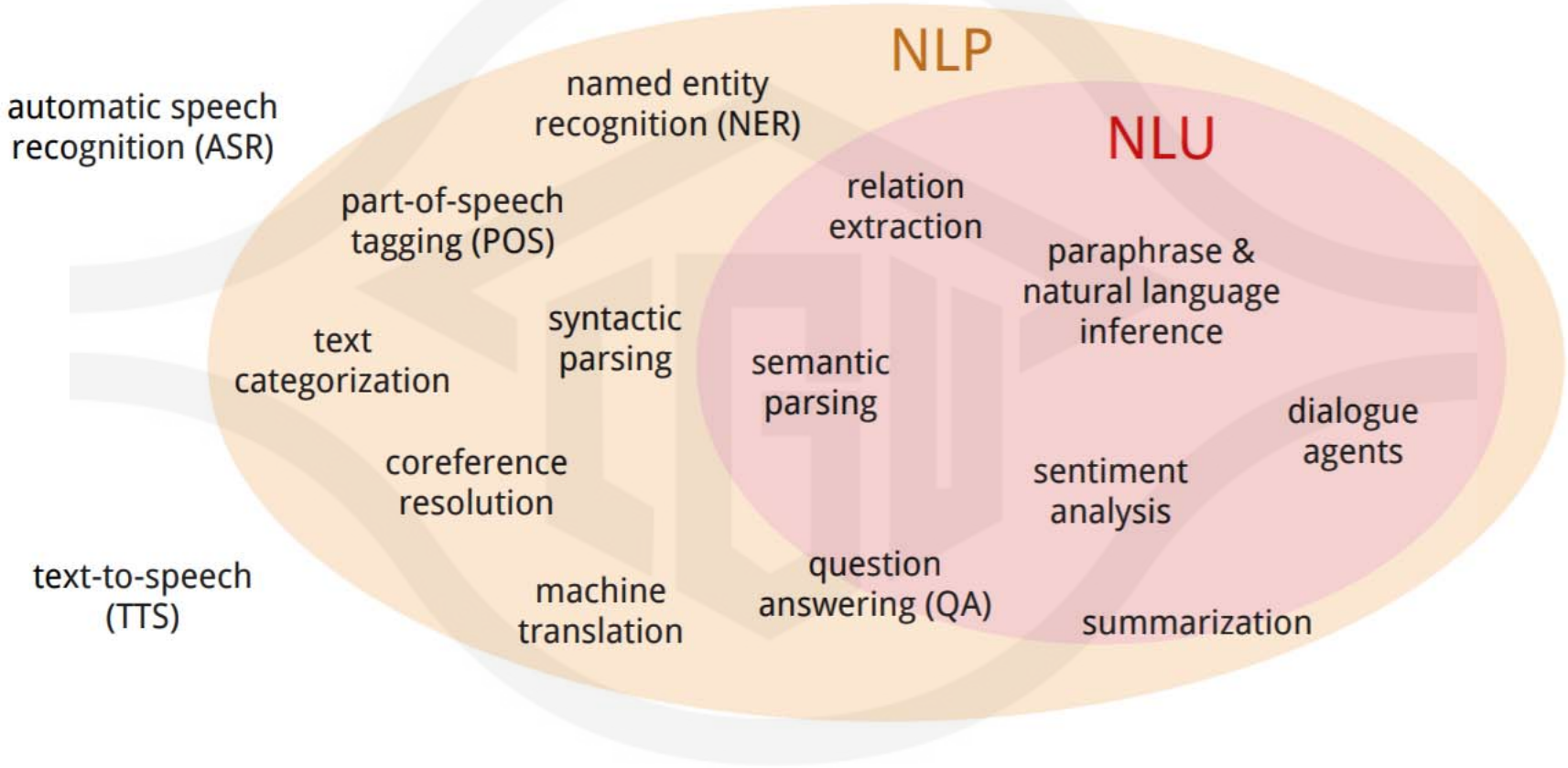
Text Data Mining

- Opinion mining (Sentiment Analysis)
  - Using natural language processing (NLP) to systematically identify, extract, quantify, and study affective states and subjective information
- Topic mining
  - Discovering the abstract "topics" that occur in a collection of documents

Text Data Mining Approach



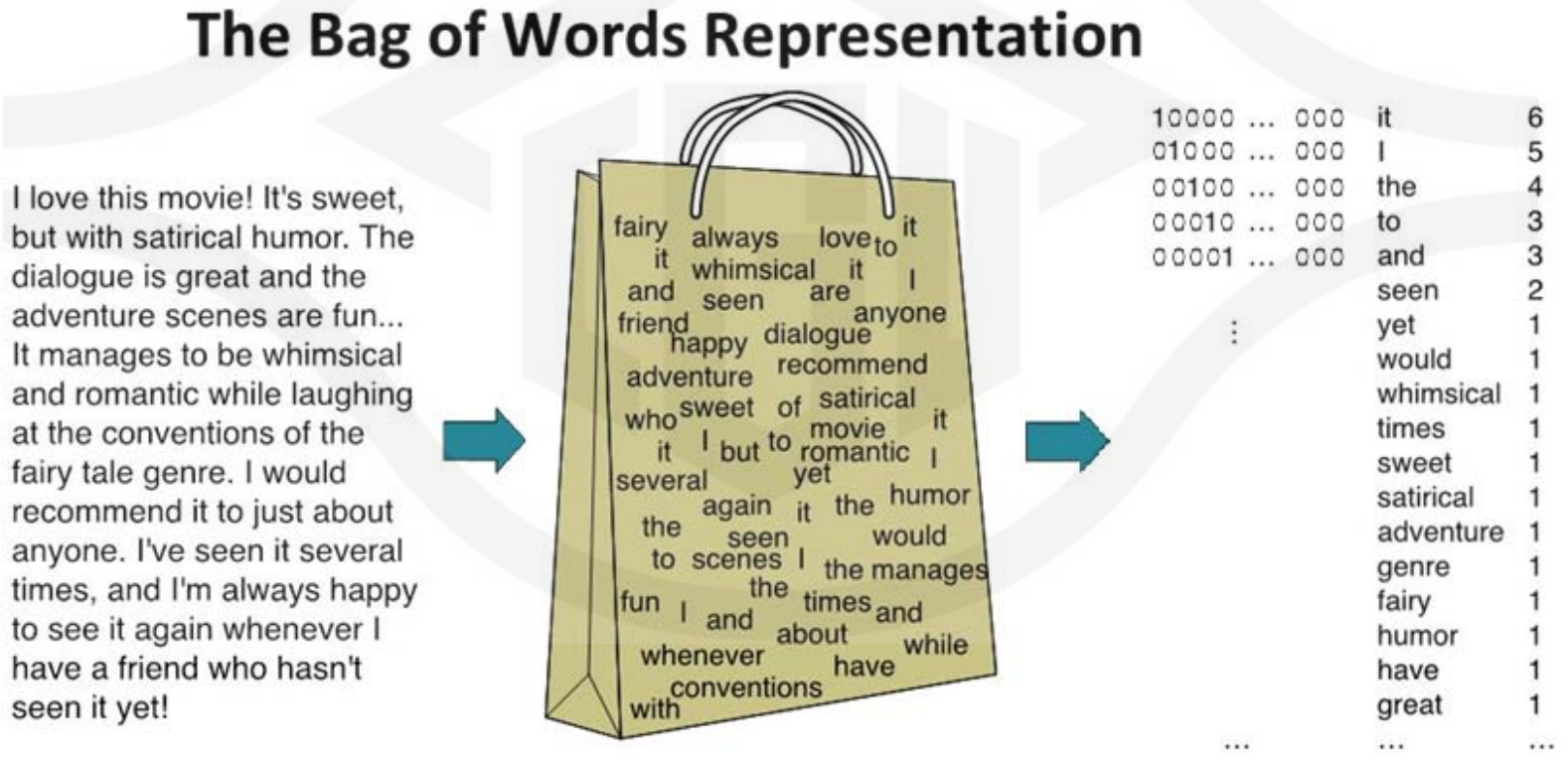
Natural Language Processing & Understanding



- https://www.youtube.com/watch?v=vcPd0V4VSNU
- http://web.stanford.edu/class/cs224u/#

Text representation (BOW + TF-IDF)

- Represent a document as a “bag” of important keywords, without ordering of the words



Bag of Words (Words Tokenization)

- Represent the sentence “code is written in Python” by **One Hot Encoding**

code	1	0	0	0	0
is	0	1	0	0	0
written	0	0	1	0	0
in	0	0	0	1	0
Python	0	0	0	0	1

- The word “written” is mapping to “00100”

Bag of Words (Doc-Words Vectorized)

- Example:
  - This is why I hate the Da Vinci Code, it is so boring
  - The code is written in Python
  - This is fucking horrible

	boring	code	da	fucking	hate	horrible	in	is	it	python	so	the	this	vinci	why	written
Document 1	1	1	1	0	1	0	0	2	1	0	1	1	1	1	1	0
Document 2	0	1	0	0	0	0	1	1	0	1	0	1	0	0	0	1
Document 3	0	0	0	1	0	1	0	1	0	0	0	0	1	0	0	0

raw term frequencies $\mathrm{tf}(t,d)$

Vectorized Text by sklearn

- Implement BOW

```
docs = np.array(['This is why I hate the Da Vinci Code, it is so boring', 'The code is written in Python', 'This is fucking horrible'])

## Vectorized
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer()
BOW = vectorizer.fit_transform(docs)
print(BOW.toarray())
print(vectorizer.vocabulary_)
```

Expected Overlap of Words in Context (TF-IDF)

- The importance of words can be estimated from the frequency of appearance of words in the text
  - Term Frequency : TF
- Frequently occurring words across multiple documents typically don't contain useful information
  - Document Frequency : DF

- TF-IDF : $\text{tf-idf}(t,d) = \text{tf}(t,d) \times (\text{idf}(t,d) + 1)$

$$\text{idf}(t,d) = log \frac{1 + n_d}{1 + \text{df}(d,t)}$$

Term Frequency and Document Frequency

- game TF:11 DF:2
- shoot TF:3 DF:2
- better TF:3 DF:1
- percent TF:2 DF:1
- minute TF:3 DF:1
- win TF:3 DF:1

Document 1

After shooting a combined 23 percent in Game 3, the three players were determined to play better than they had Sunday, and that meant squeezing in some extra repetitions. Some improvements were made -- ***Antetokounmpo*** finished with 25 points, 13 more than he had in Game 3 -- but it wasn't enough. The ***Milwaukee Bucks*** dropped Game 4 of the Eastern Conference finals to the Toronto Raptors 120-102, only the second time this season they have lost back-to-back games. "We just came out flat in the third quarter," ***Antetokounmpo*** said. "It's something we can get better at -- something we can fix." The ***Bucks***' system is built to withstand an individual player's shooting slump, but ***Bledsoe*** is frustrated with just how long he has struggled to find the basket. According to Second Spectrum tracking, ***Bledsoe*** is shooting just 27 percent on his jump shots this postseason, the worst among all players with at least 50 attempts. "I tell him just forget about it," ***Middleton*** told ***ESPN***. "That's the only way you can play better, is if you stop thinking about it so much."

Document 2

"Feel good," said ***Leonard***, who finished with 19 points on 6-for-13 shooting in 34 minutes. "Keep going, keep fighting. We have a chance to make history." Asked if the minutes from Game 3 caught up with him in Game 4, ***Leonard*** passed on answering. "There's no excuses," he said. "You're playing basketball. We got a win tonight." For so much of these playoffs, the ***Raptors*** have been getting wins because of ***Leonard***'s heroics. That was the case in both of the previous two games ***Toronto*** had played here at ***Scotiabank Arena*** -- in Game 7 against the ***Sixers***, in which he hit a classic game winner, and in Sunday's Game 3, when he played through those career-high 52 minutes. The privilege of having a transcendent superstar like ***Leonard*** isn't just the gift of the singular performance that wins a game, though ***Leonard*** has done plenty of that over the past six weeks of the postseason.

Words Relevancy by sklearn

- Implement TF-IDF

```
## TF-IDF
from sklearn.feature_extraction.text import
TfidfTransformer
np.set_printoptions(precision=2)
tfidf = TfidfTransformer(use_idf=True, norm='l2',
smooth_idf=True)
X = tfidf.fit_transform(BOW)
print(X.toarray())
```

Stop Words in Text

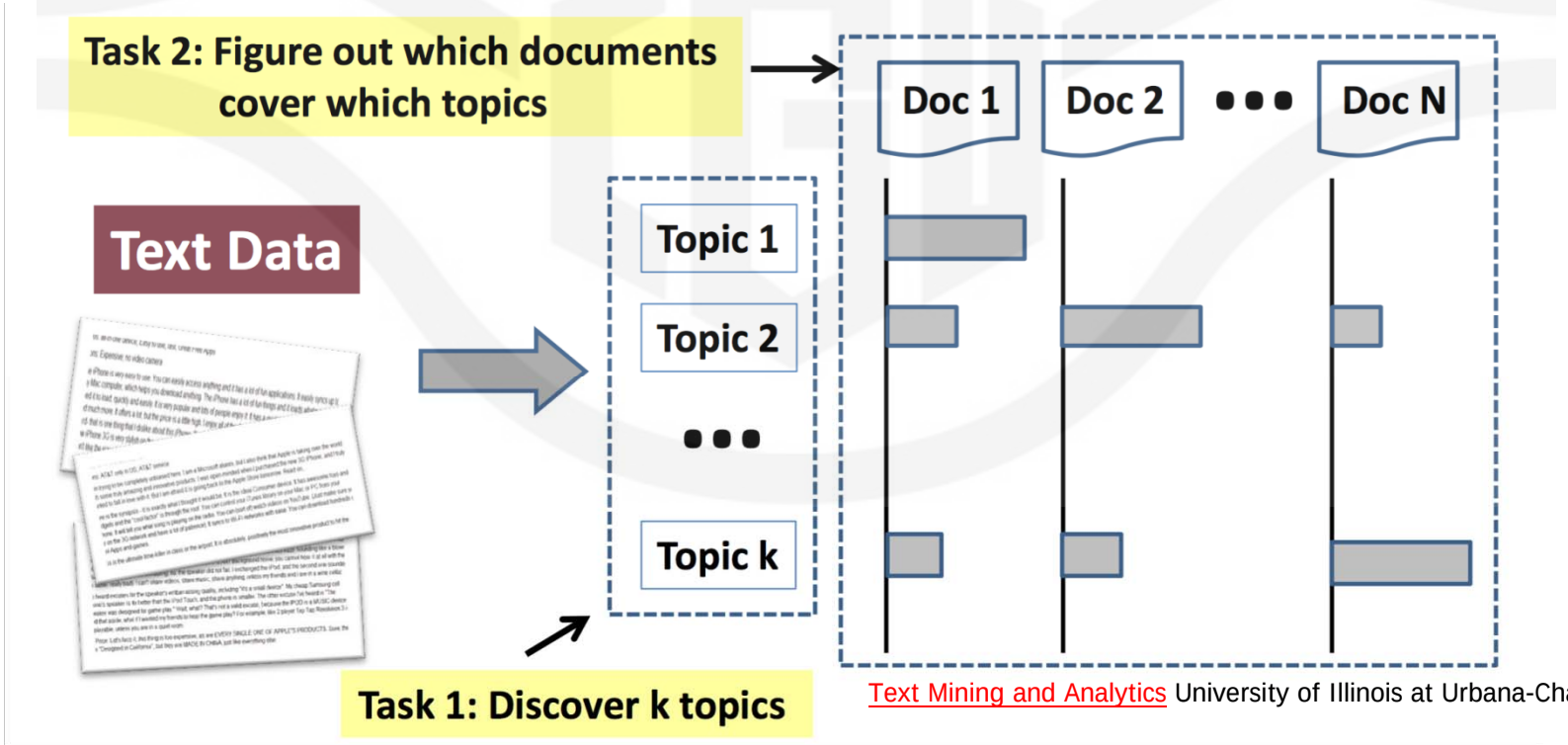
- In English: I, you, is, an, the, of, in...
- 中文：的, 啊, 了, 個...

- Implement by sklearn

```
## stop word
vectorizer_stopwords =
CountVectorizer(stop_words="english")
BOW_stopwords = vectorizer_stopwords.fit_transform(docs)
print(BOW_stopwords.toarray())
print(vectorizer_stopwords.vocabulary_)
```

Topic mining (Topic modeling)

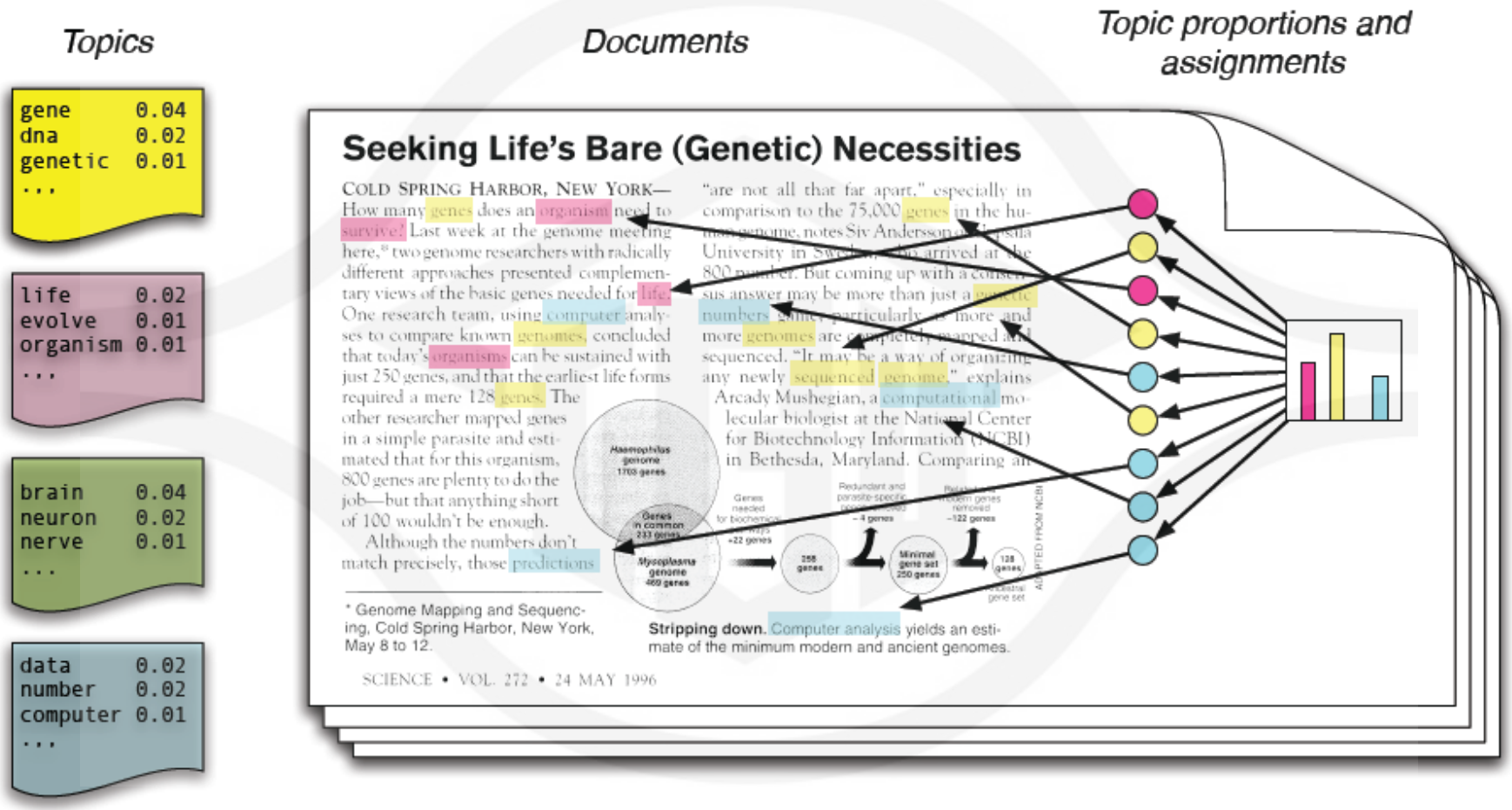
- Topic modeling describes the broad task of assigning topics to unlabelled text documents.
- A typical application would be the categorization of documents in a large text corpus of newspaper articles
- Consider topic modeling as a clustering task, an unsupervised learning



Topic mining (Cont.)

- Bag-of-words approach:
  - Mixture of unigram language model
  - Expectation-maximization algorithm
  - Probabilistic latent semantic analysis
  - **Latent Dirichlet allocation (LDA) model**
- Graph-based approach :
  - TextRank (Mihalcea and Tarau, 2004)
  - Reinforcement Approach (Xiaojun et al., 2007)
  - CollabRank (Xiaojun er al., 2008)

Latent Dirichlet Allocation



Topic Mining Example (Sport News)

- Utilize 3 sport news articles about NBA Eastern Conference Final Game 4 (2019.05.22)

- Two articles from ESPN
- One article from USA TODAY

Document 1 (ESPN)

Bucks reassure Bledsoe as shooting woes linger

Malika Andrews
ESPN

May 22, 2019

After shooting a combined 23 percent in Game 3, the three players were determined to play better than they had Sunday, and that meant squeezing in some extra repetitions. Some improvements were made -- ***Antetokounmpo*** finished with 25 points, 13 more than he had in Game 3 -- but it wasn't enough. The ***Milwaukee Bucks*** dropped Game 4 of the Eastern Conference finals to the Toronto Raptors 120-102, only the second time this season they have lost back-to-back games. "We just came out flat in the third quarter," ***Antetokounmpo*** said. "It's something we can get better at -- something we can fix." The ***Bucks'*** system is built to withstand an individual player's shooting slump, but ***Bledsoe*** is frustrated with just how long he has struggled to find the basket. According to Second Spectrum tracking, ***Bledsoe*** is shooting just 27 percent on his jump shots this postseason, the worst among all players with at least 50 attempts. "I tell him just forget about it," ***Middleton*** told ***ESPN***. "That's the only way you can play better, is if you stop thinking about it so much."

Document 2 (ESPN)

VanVleet's shot reborn after birth of 2nd child

Tim Bontemps
ESPN

2:41 PM CT

"Feel good," said ***Leonard***, who finished with 19 points on 6-for-13 shooting in 34 minutes. "Keep going, keep fighting. We have a chance to make history." Asked if the minutes from Game 3 caught up with him in Game 4, ***Leonard*** passed on answering. "There's no excuses," he said. "You're playing basketball. We got a win tonight." For so much of these playoffs, the ***Raptors*** have been getting wins because of ***Leonard***'s heroics. That was the case in both of the previous two games ***Toronto*** had played here at ***Scotiabank Arena*** -- in Game 7 against the ***Sixers***, in which he hit a classic game winner, and in Sunday's Game 3, when he played through those career-high 52 minutes. The privilege of having a transcendent superstar like ***Leonard*** isn't just the gift of the singular performance that wins a game, though ***Leonard*** has done plenty of that over the past six weeks of the postseason.

Document 3 (USA TODAY)

Raptors even Eastern Conference finals with blowout Game 4 win over Bucks

Jeff Zillgitt, USA TODAY Published 11:04 p.m. ET May 21, 2019 | Updated 2:51 a.m. ET May 22, 2019

Antetokounmpo finished with 25 points, 10 rebounds and five assists. ***Khris Middleton*** scored a game-high 30 points, but ***Nikola Mirotic*** was the only other ***Bucks*** player with double-digit points. ***Milwaukee*** shot just 31.4% on threes. "We're going to have to finish better at the 3-point line or make more threes," ***Budenholzer*** said. It is just the second time this season the ***Bucks*** lost two consecutive games, and even though they trailed ***Boston*** 1-0 in the conference semifinals, they now face their biggest test of the playoffs with a spot in the Finals distilled to a best-of-3. When the ***Raptors*** left ***Milwaukee***, they had questions to answer and found them at home. Leaving ***Toronto***, the ***Bucks*** have their own problems to solve. How to get more from starting point guard ***Eric Bledsoe*** and reserve guard ***Malcolm Brogdon***? How to take some offensive pressure off ***Antetokounmpo*** and ***Middleton***? And how to slow down ***Toronto***'s offense?

Observe Key Words and Guess

Bucks reassure Bledsoe as shooting woes linger

Malika Andrews
ESPN

May 22, 2019

VanVleet's shot reborn after birth of 2nd child

Tim Bontemps
ESPN

2:41 PM CT

Raptors even Eastern Conference finals with blowout Game 4 win over Bucks

Jeff Zillgitt, USA TODAY Published 11:04 p.m. ET May 21, 2019 | Updated 2:51 a.m. ET May 22, 2019

Using LDA Model (Topic = 2)

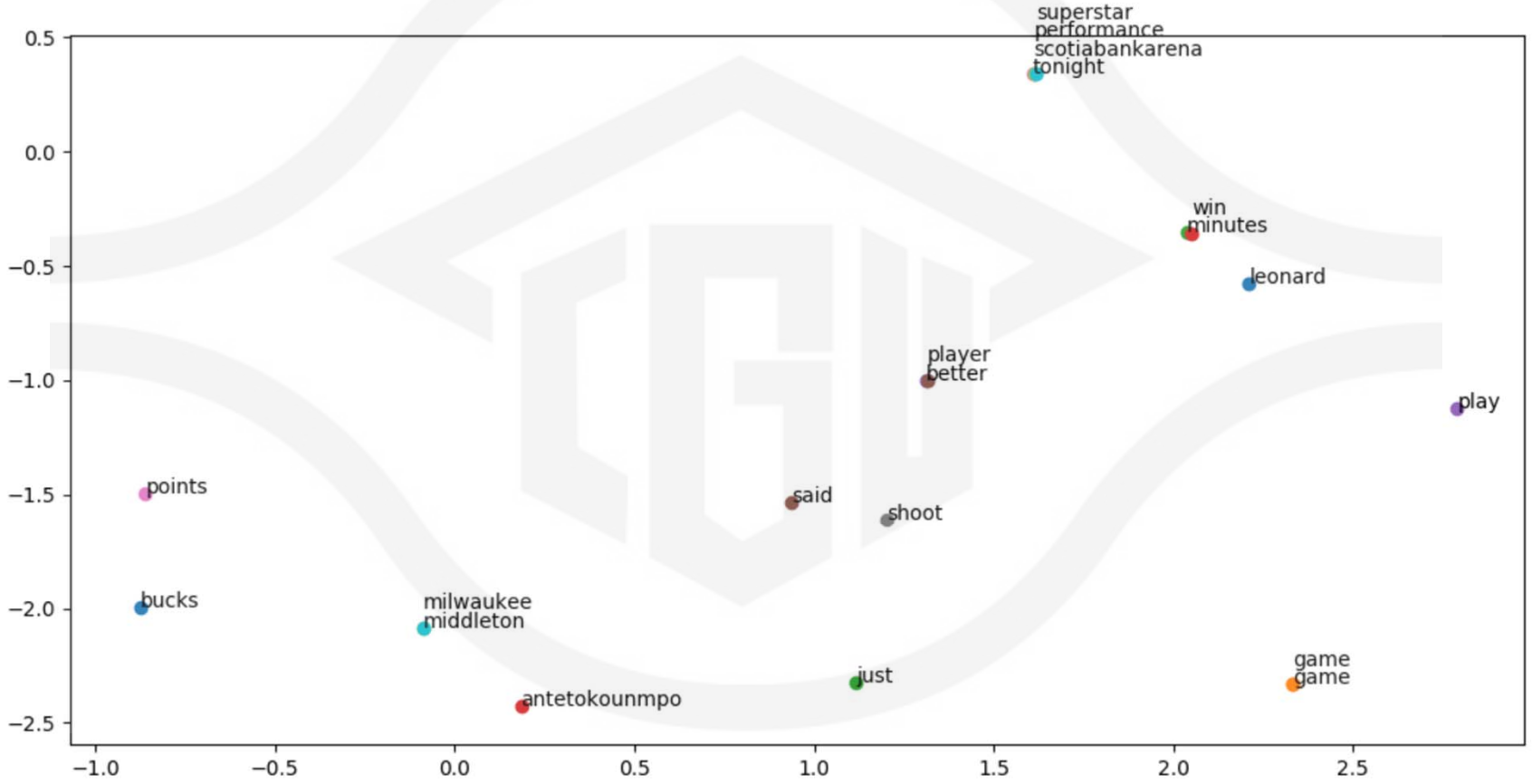
```
from sklearn.decomposition import LatentDirichletAllocation
lda = LatentDirichletAllocation(n_components=2,
random_state=0)
lad_result = lda.fit_transform(X)
lad_result = np.argmax(lad_result, axis=1)
print(lad_result)
n_top_words = 10
feature_names = vectorizer.get_feature_names()
print(feature_names[0])
for topic_idx, topic in enumerate(lda.components_):
    print("Topic %d:" % (topic_idx))
    print(" ".join([feature_names[i] for i in
topic.argsort() [:-n_top_words - 1:-1]]))
```

Allocation Results [101]

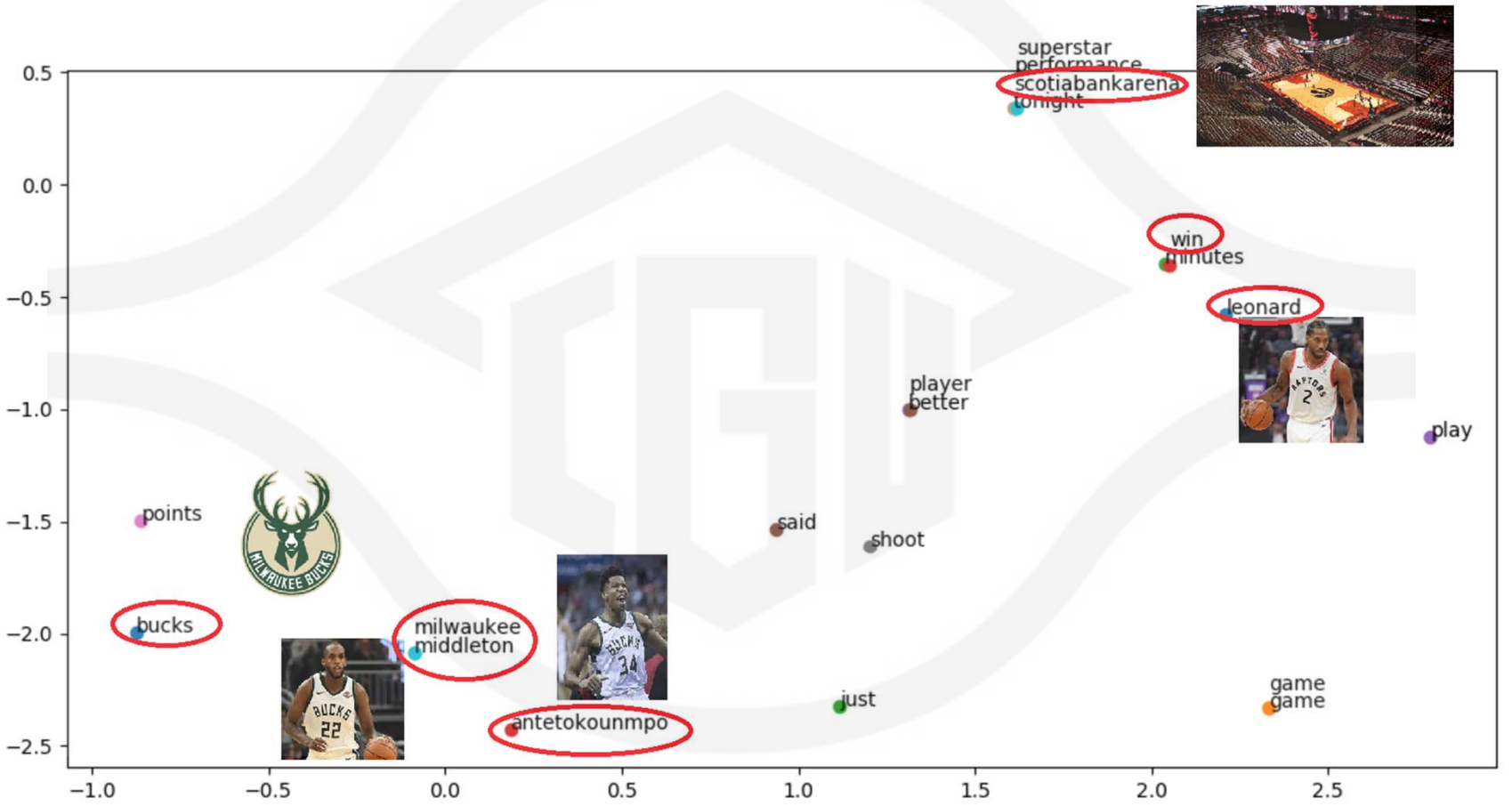
- Because there are only 3 articles, pick k = 2
- The first article is assigned the same topic as the third article.

```
{'shoot': 53, 'percent': 28, 'game': 9, 'player': 31, 'play': 30, 'better': 3, 'sunday': 64, 'meant':
 18, 'squeezing': 60, 'repetitions': 44, 'antetokounmpo': 2, 'finished': 8, '25': 1, 'points': 35, '1
3': 0, 'wasn': 75, 'milwaukee': 20, 'bucks': 5, 'conference': 6, 'finals': 7, 'toronto': 71, 'raptor
s': 42, 'second': 50, 'time': 68, 'season': 49, 'lost': 15, 'just': 13, 'quarter': 40, 'said': 46, 's
lump': 58, 'bledsoe': 4, 'struggled': 63, 'secondspectrum': 51, 'tracking': 72, 'shots': 54, 'postsea
son': 36, 'middleton': 19, 'told': 69, 'way': 76, 'stop': 62, 'thinking': 66, 'leonard': 14, 'minute
s': 21, 'going': 10, 'make': 16, 'passed': 26, 'win': 78, 'tonight': 70, 'playoffs': 32, 'scotiabanka
rena': 48, 'sixers': 56, 'winner': 79, 'high': 12, 'privilege': 38, 'transcendent': 74, 'superstar':
 65, 'singular': 55, 'performance': 29, 'plenty': 33, 'past': 27, 'weeks': 77, 'rebounds': 43, 'score
d': 47, 'nikola': 23, 'mirotic': 22, 'threes': 67, 'point': 34, 'trailed': 73, 'semifinals': 52, 'spo
t': 59, 'questions': 41, 'problems': 39, 'starting': 61, 'guard': 11, 'reserve': 45, 'malcolm': 17,
 'offensive': 25, 'pressure': 37, 'slow': 57, 'offense': 24}
[1 0 1]
13
Topic 0:
leonard game minutes win play said tonight scotiabankarena performance superstar
Topic 1:
bucks game just antetokounmpo better player points shoot middleton milwaukee
```

10 Top Words in Vector Space (Word Embedding)



Special Phenomenon (Nearest neighbors)



Why Word Embedding ?

- BOW loss the information about words order
- Example:
  - White blood cells destroying an infection
  - An infection destroying white blood cells

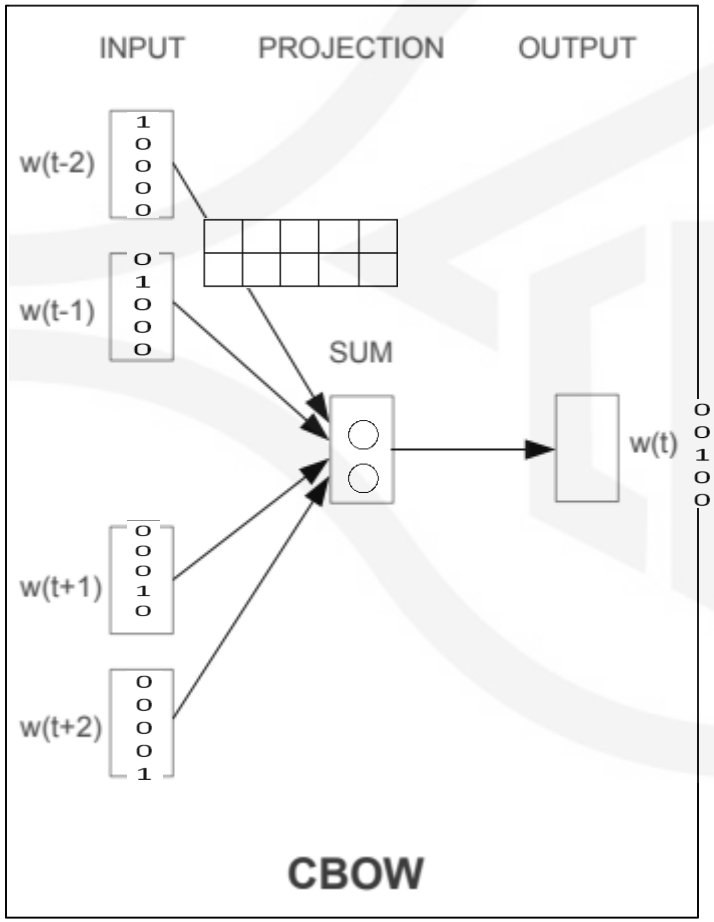
- Sparse matrix have the **curse of dimensionality** problem (One Hot Encoding)
- Using **finite-size**d vectors to represent an **infinite** number of real numbers → Word Embedding

Word Embedding Implementation

- Word embedding method
  - Word2Vec (Google, 2013)
  - Glove (Stanfrod, 2014)
  - FastText (Facebook, 2016)
  - ELMo(AI2, 2018)
- Self-supervise learning

Word2Vec (Counties BOW)

- code is □ □ □ □ in Python



hidden layer

network's weights

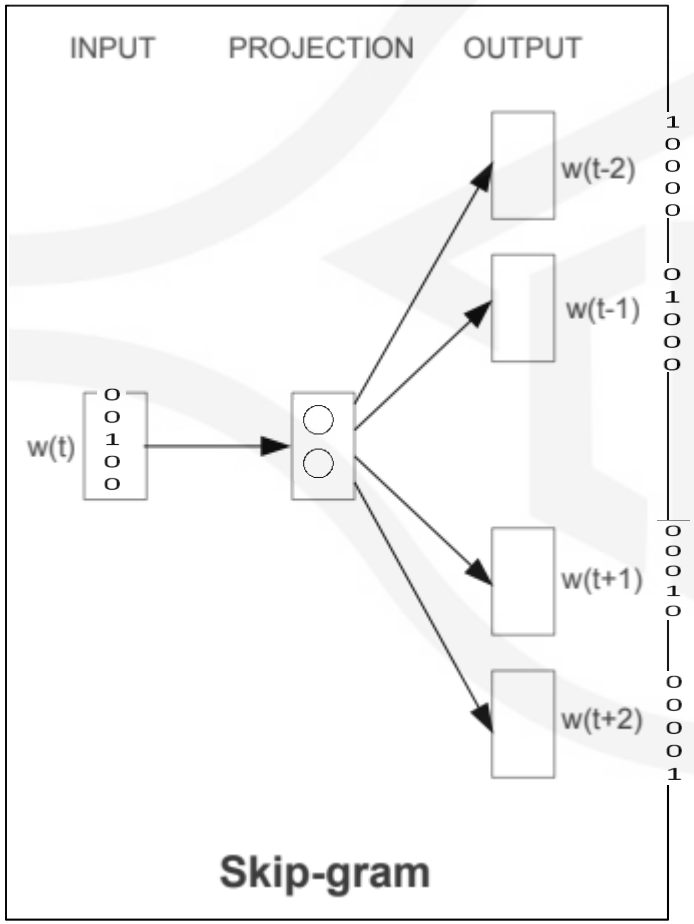
.28	.19	.2	.01	.7
.63	.5	.15	.81	.54

input vector

→(0.2, 0.15)
The embedding vector of "written"

Word2Vec (Skip-gram)

- □□□□ □□□□ written □□□□ □□□□



Glove ☹Global Vectors for Word Representation)

- Consider statistical information with word-word **co-occurrence matrix**

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

  - K related to ice but not related to steam, such as k = solid, the ratio will be large
  - K related to steam but not related to ice, such as k = gas, the ratio will be small
  - K related to both words (k=water) or not related to both words (k=fashion), this ratio will be close to 1

- Objective Function

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}} \longrightarrow J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

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

Word Embedding by PyTorch

- Implement by nn.Embedding()

```
docs = np.array(['This is why I hate the Da Vinci Code, it
is so boring', 'The code is written in Python', 'This is
fucking horrible'])
vectorizer = CountVectorizer()
BOW = vectorizer.fit_transform(docs)
print(vectorizer.vocabulary_)
word_to_ix = vectorizer.vocabulary_
# 16 words in vocab, 2 dimensional embeddings
embeds = nn.Embedding(16, 2)
lookup_tensor = t.tensor([word_to_ix["written"]],
dtype=t.long)
written_embed = embeds(lookup_tensor)
print(written_embed)
```

Reference

- Sebastian Raschka, Vahid Mirjalili. Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow. Second Edition. Packt Publishing, 2017.
- Vishnu Subramanian. Deep Learning with PyTorch: A practical approach to building neural network models using PyTorch. Packt Publishing, 2018.