Feature Extraction

BAF675 금융 빅데이터 분석

이재훈, Week 6

Feature Extraction, or Text Representation

- Texts need to be converted into vectors to be fed to machines
- The conversion of texts to vectors is called **Feature Extraction** or **Text Representation**

Text Representation

- There are three types of text representation methods
  - **Simple word counting**:
    - One Hot Encoding, Bag of Words (BoW), TF-IDF
  - **Context-free embedding**:
    - Word2vec (Google), fastText (Facebook), GloVe (Stanford Univ)
  - **Context-based embedding**:
    - ELMo, BERT (Google)

Simple Word Counting

- Okay, let's start with a simple example.
- Let's say we have following three sentences.
  - "Cat growls"
  - "Dog barks"
  - "Dog and cat play together"

Simple Word Counting

- The simplest way to convert is simply counting words
  - Vocabulary: ["cat", "dog", "growls", "barks", "play", "and", "together"]
  - S1: "Cat growls" => [1, 0, 1, 0, 0, 0, 0]
  - S2: "Dog barks" => [0, 1, 0, 1, 0, 0, 0]
  - S3: "Dog and cat play together" => [1, 1, 0, 0, 1, 1, 1]
- That's it! This is how **Bag-of-Words (BoW)** work

Simple Word Counting

- **Bag of Words (BoW)**
  - Simply count the frequency of each word
- **One-Hot Encoding**
  - Frequency is ignored. Either 0 or 1
- **TF-IDF**
  - Assign less weight to common words and higher weight to rare words
  - (e.g.) little weight to "I", "you", "this", "they"
  - (e.g.) high weight to "crash", "war", "BTS"

Simple Word Counting

- **TF-IDF**
  (term frequency
  - inverse document frequency)

$$tf\,idf\,(t, d, D) = tf\,(t, d)\ .\ idf\,(t, D)$$

$$tf\,(t, d) = log\,(1 + freq\,(t, d))$$

$$idf\,(t, D) = log\,\left(\frac{N}{count\,(d \in D : t \in d)}\right)$$

Simple Word Counting

- There are a few problems though
- #1. "dog bites human" and "human bites dog" are represented by the same vector
  - This case may sound weird, but in fact it is not a big problem.
  - For example, let's assume we want to do sentiment classification (positive or negative). Both cases would be called negative.

Simple Word Counting

- #2. "cat" and "dog" are equally as far as "cat" and "altogether"
  - This can be a serious problem, especially when data sample are not sufficient

Embedding

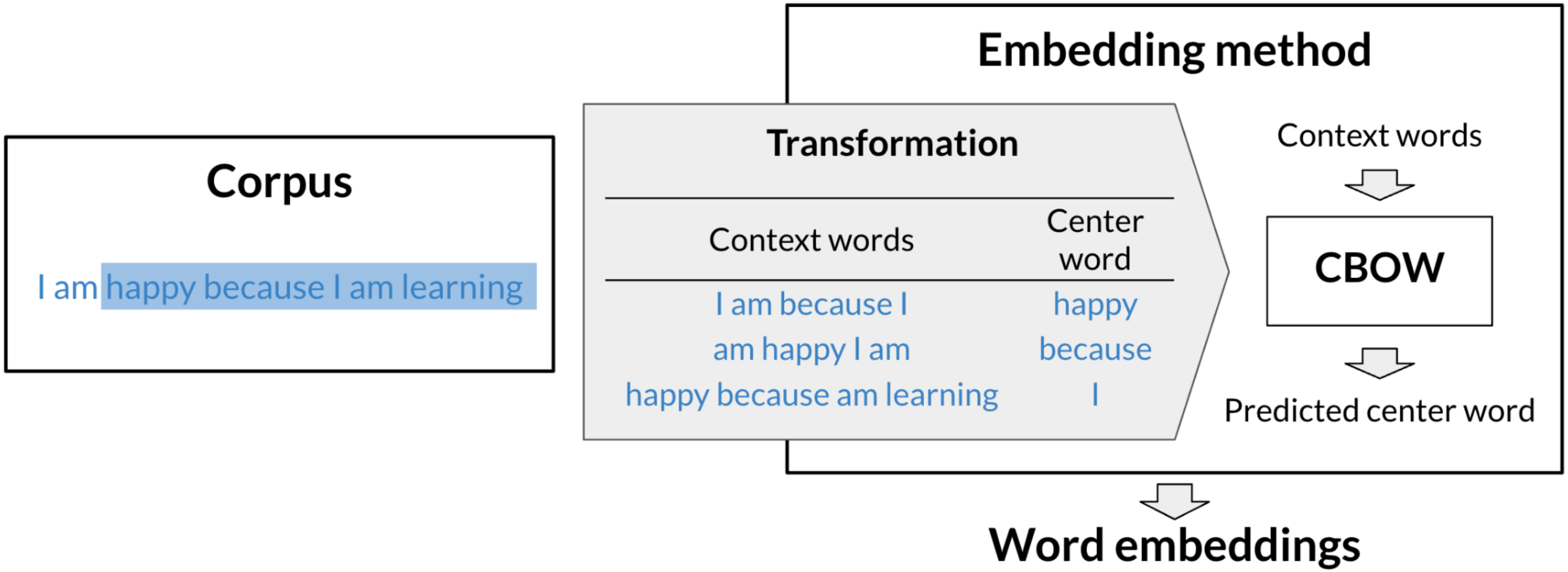
- That's why **Embedding** methodology was developed!
- Embedding maps a word to a vector based on distributional representation.
  - Earlier, each vocabulary is mapped to an independent dimension
    (# vocabularies = # dimensions)
  - Now, each vocabulary is mapped to a point in vector space
    (# vocabularies > # dimensions)

Embedding

- Continuous Bag of Words Model (CBOW) example
  - S1: I serve my lord.
  - S2: I serve my wife.
  - => CBOW would learn "lord" is similar to "wife".
- Thus, called self-supervised learning.

Continuous Bag of Words Model

- Self-supervised learning



Continuous Bag of Words Model

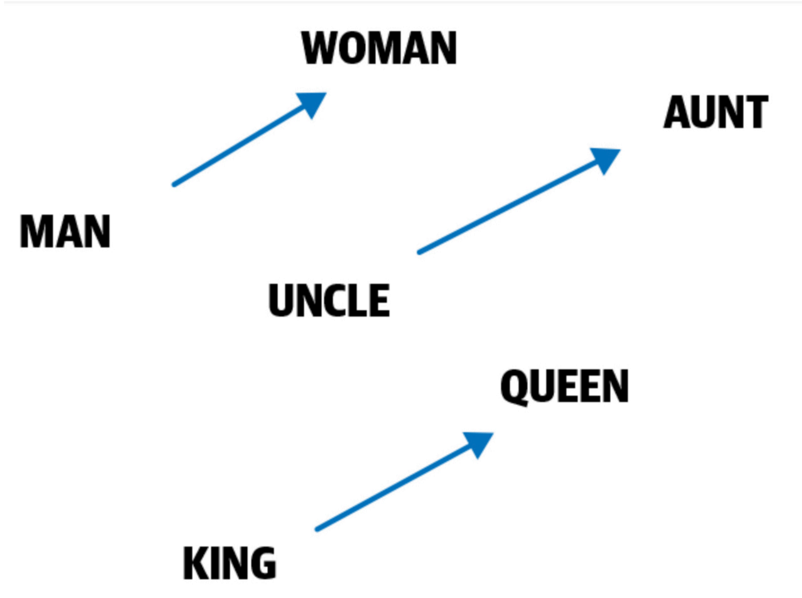
- Self-supervised learning

$$
\left[
\begin{array}{rccccccc}
 & \text{I} & & \text{am} & & \text{because} & & \text{I} \\
\begin{array}{r}\text{am}\\ \text{because}\\ \text{happy}\\ \text{I}\\ \text{learning}\end{array} &
\begin{pmatrix}0\\0\\0\\1\\0\end{pmatrix} & + &
\begin{pmatrix}1\\0\\0\\0\\0\end{pmatrix} & + &
\begin{pmatrix}0\\1\\0\\0\\0\end{pmatrix} & + &
\begin{pmatrix}0\\0\\0\\1\\0\end{pmatrix}
\end{array}
\right] / 4 \;=\;
\begin{array}{c}
\text{I am because I} \\
\begin{pmatrix}0.25\\0.25\\0\\0.5\\0\end{pmatrix}
\end{array}
$$

Context words	Context words vector	*Center word*	Center word vector
I am because I	[0.25; 0.25; 0; 0.5; 0]	*happy*	[0; 0; 1; 0; 0]

Embedding

- What is possible with embeddings
  - "cat" and "dog" are closer than "cat" and "altogether"
  - Such relation can be induced:
    - King - Man + Woman ≃ Queen
- This is what **Word2Vec** can do!!



Embedding

- Continuous bag-of-words (CBOW)
- Continuous skip-gram / Skip-gram with negative sampling (SGNS)
- word2vec (Google, 2013)
- Global Vectors (GloVe) (Stanford, 2014)
- fastText (Facebook, 2016)

Embedding

- However, there still is a problem.
- Word2Vec maps each vocabulary to a vector.
- However, the word "bank" has different meanings in the following two sentences.
  - "Open a bank account"
  - "Sit on river bank"
- How to solve the problem??

Context-based embedding

- We, humans, know the two "bank"s have different meanings
  - How? Because of its contexts!
- Thus, machines can also distinguish the difference by taking into account the context in the embedding process
- **Word2vec vs. BERT**
  - **Word2vec** maps a word to a vector **unconditionally**
  - **BERT** maps a word to a vector **conditional on surrounding words**

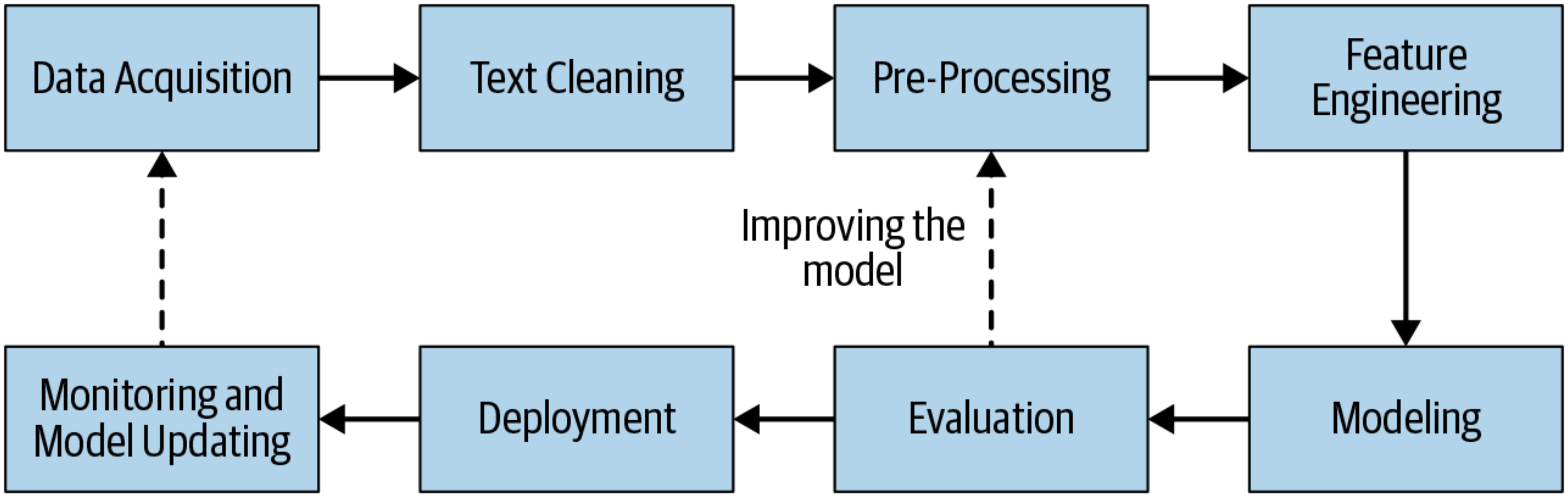
Text Representation: Summary

- **Simple word counting**:
  - Bag of Words (BoW), One Hot Encoding, TF-IDF
- **Context-free embedding**:
  - Word2vec (Google), fastText (Facebook), GloVe (Stanford Univ)
- **Context-based embedding:**
  - ELMo, BERT (Google)

Text Representation: Example

- **Example 1**
  - Banana, Apple, Porsche => **Word2Vec** would be useful
- **Example 2**
  - Banana, Apple, Microsoft => **BERT** would be needed

Machine Learning Workflow



Machine Learning Workflow

- "Feature Engineering" includes
  - Bag-of-Words, Word2vec, BERT
- "Modeling" includes
  - Naive Bayes (conditional probability)
  - Logistic Regression
  - Support Vector Machine (SVM)
  - Deep Learning (DL)

Tokenization

- **Feature Extraction (or, text representation)**
  - Convert text data into vectors
- **Tokenization**
  - Breaking down documents to sentences and words.
- **Part-of-Speech (PoS) Tagging**
  - Parts of speech (e.g., verbs, nouns, prepositions, adjectives) indicate how a word is functioning within the context of a sentence.