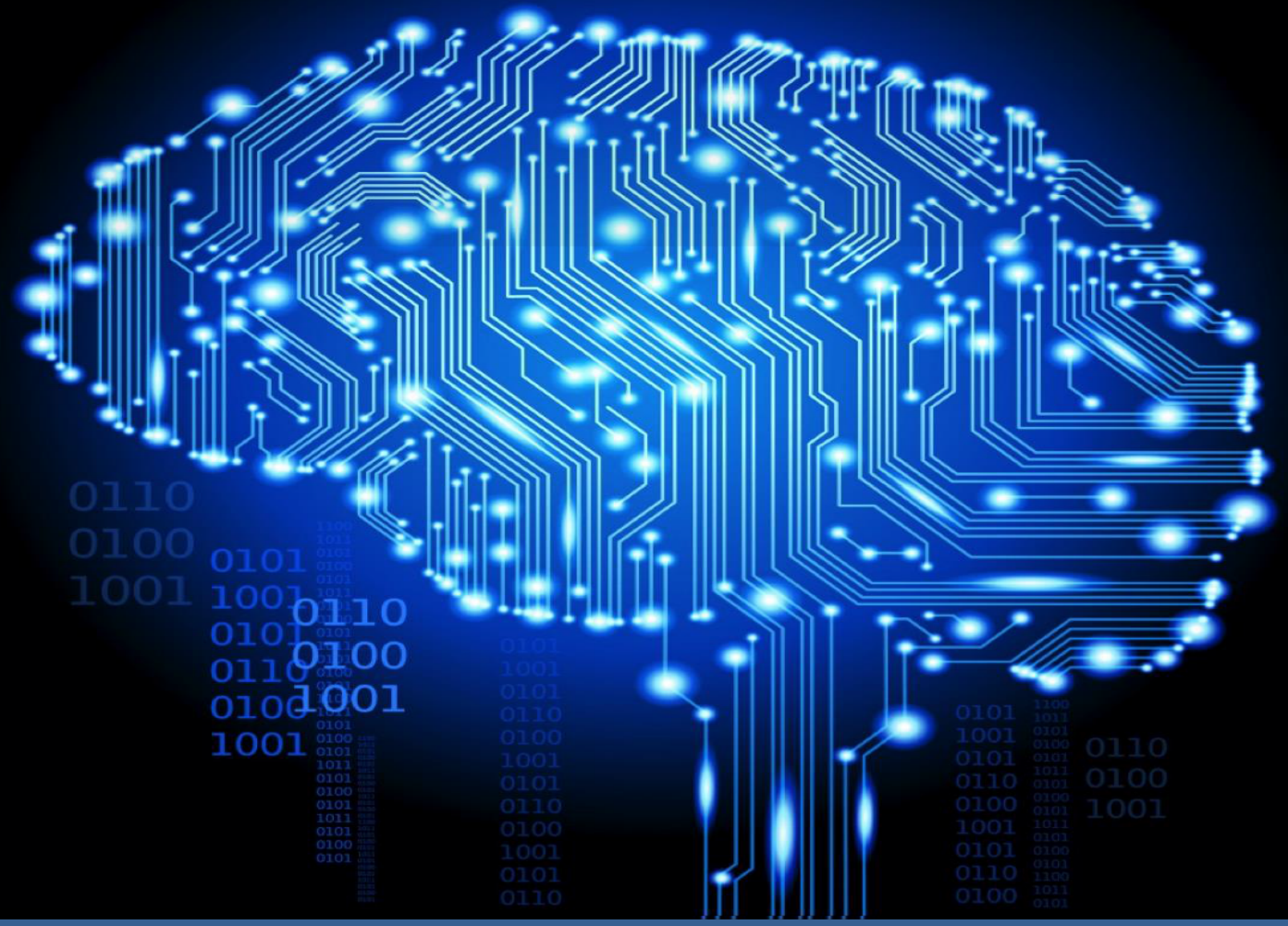
Statistical Learning and Computational Finance Lab.
Department of Industrial Engineering
http://slcf.snu.ac.kr



Day 13: NLP Basics

Agenda

- Introduction to NLP
- Text Preprocessing
- Word Embedding
- Exercises
  - NLP with SEC report
  - Sentimental Analysis
  - Keyword Extraction

Introduction to NLP

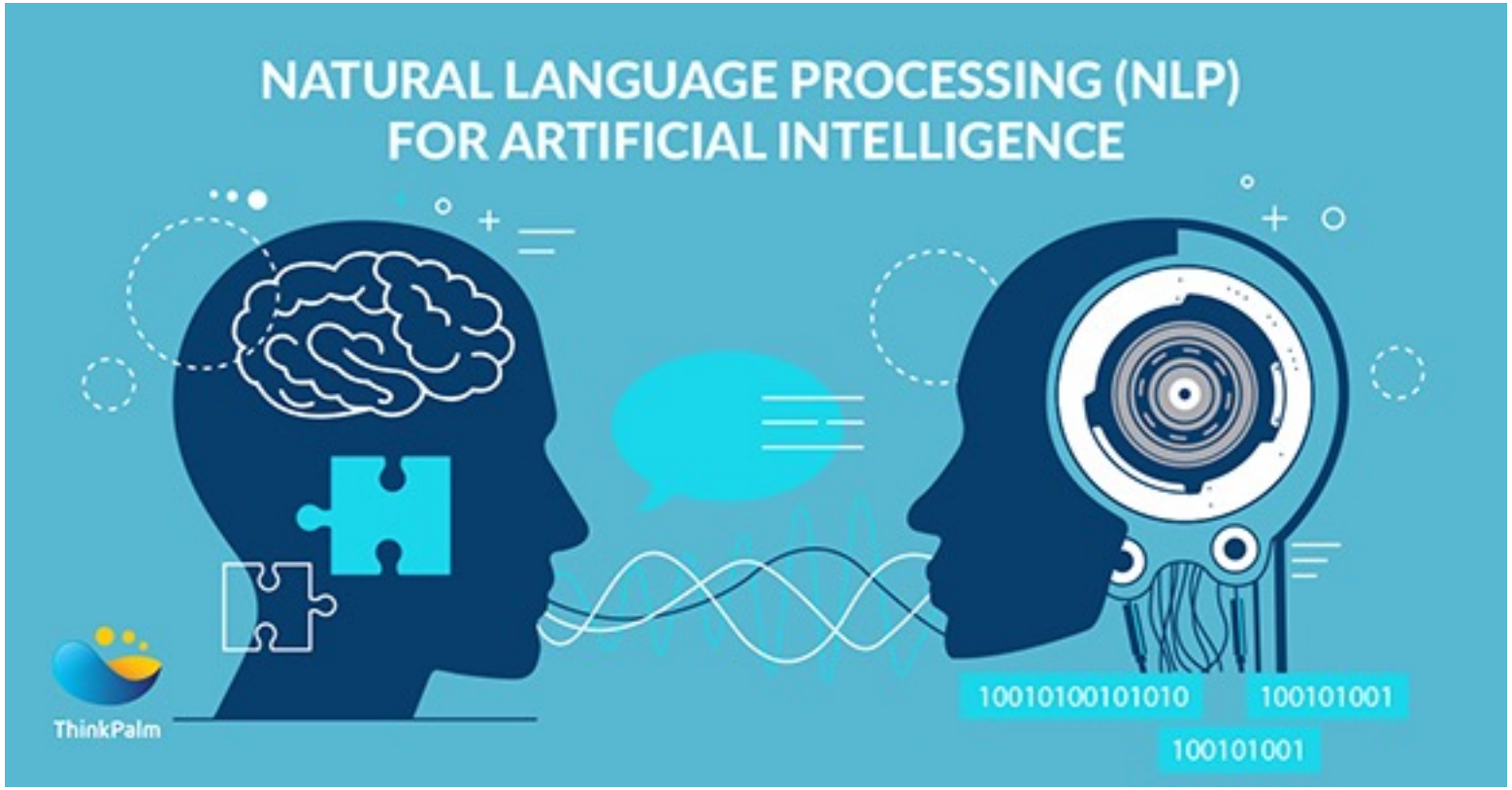


Image Source: https://thinkpalm.com/blogs/natural-language-processing-nlp-artificial-intelligence/

NLP (Natural Language Processing)

Definition

- Natural Language Processing (NLP) is a field of Artificial Intelligence (AI) that makes human language intelligible to machines
  - NLP combines the power of linguistics and computer science to study the rules and structure of language and create intelligent systems (run on machine learning and NLP algorithms) capable of understanding, analyzing, and extracting meaning from text and speech.

Benefits of NLP

- Perform large-scale analysis
- Automate processes in real-time
- Tailor NLP tools to your industry
- And so much more

Refenrence: https://monkeylearn.com/natural-language-processing/

NLP Examples

- Common Examples of NLP
  - Email filters
  - Virtual assistants, voice assistants, or smart speakers
  - Online search engines
  - Predictive text and autocorrect
  - Sentiment Analysis
  - Automating processes in customer support
  - Chatbots
  - Automatic summarization
  - Machine translation
  - Natural language generation

Refenrence: https://monkeylearn.com/natural-language-processing/

NLP in Finance

- 5 Use Cases of NLP in the Finance Sector
  - Unstructured Data Utilization
    - Utilize finance data stored in varying formats like pdf, XML, HTML, web, feeds
  - Efficient Text Analytics With NLP
    - Ex) Determine underlying sentiment and extract key financial entities
  - Financial Document Analyzer
    - Able to read and comprehend large volumes of financial documents automatically
  - Content Enrichment
    - Compose better investment management and improve risk management and compliance
  - Reporting and Omnichannel Customer Engagement
    - Ex) Forecast and detect customer pain points

Reference: https://apiway.ai/community/articles/821-top-5-use-cases-of-nlp-in-finance

Why is Korean Language difficult

- Korean is one of the most difficult language for a computer. Here are some reasons
  - Word spacing
    - 아버지가방에들어가신다
  - Homonyms
    - 3연패 (win championship 3 times in a row, losing streak of 3 games)
    - 사과 (apple, apology)
  - Flexibility
    - 나는 밥을 먹으러 간다 == 밥을 먹으러 나는 간다.
- Additional Resource
  - Please check this GitHub repo: https://github.com/datanada/Awesome-Korean-NLP

Python libraries for NLP

- NLTK (Natural Language ToolKit)
- KoNLPy
  - Open Source '형태소 분석기'
- Gensim
  - Word2Vec Library
- Scikit-Learn
  - Preprocessing
- Huggingface
  - Attention-based models

Text Preprocessing

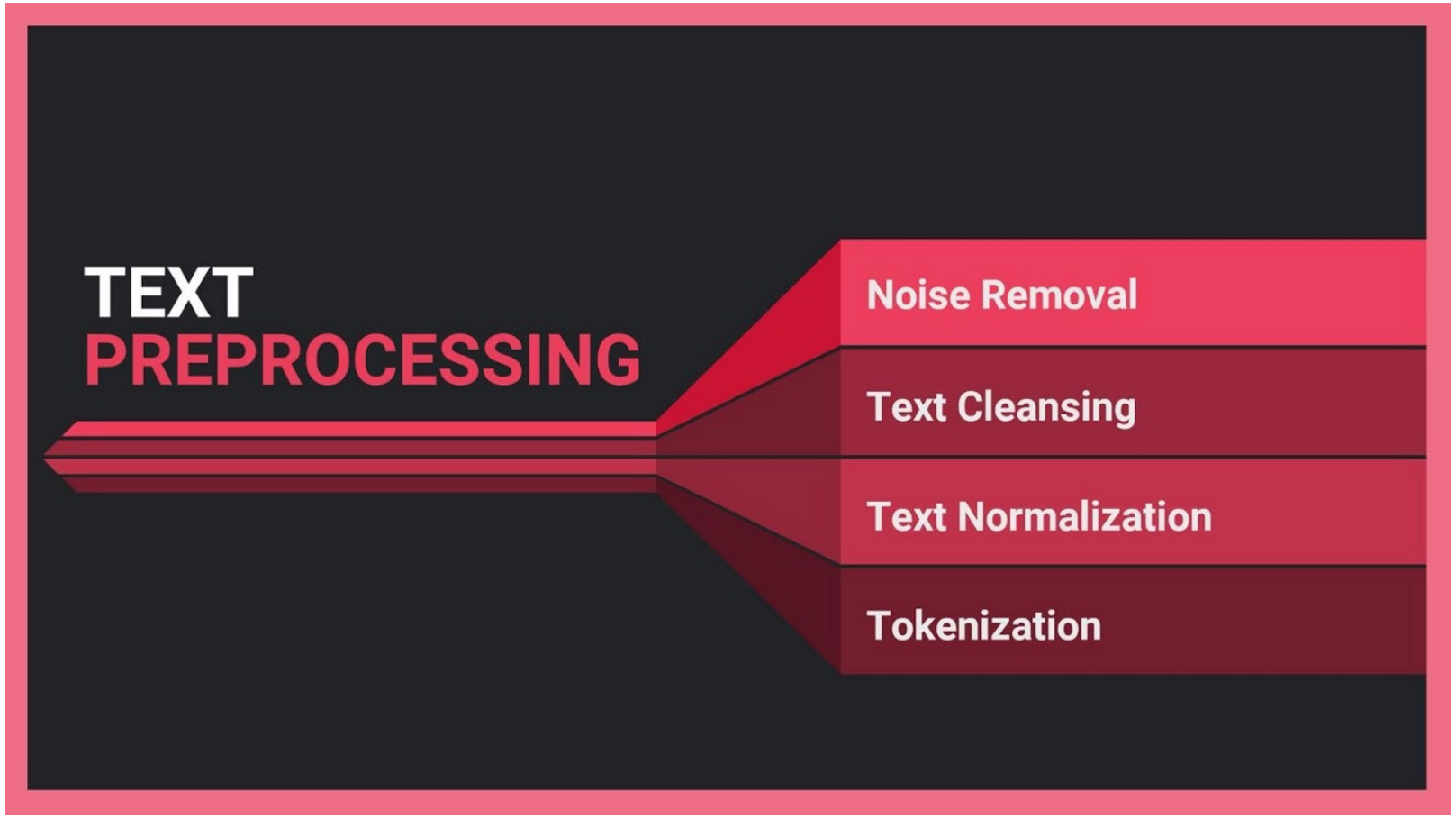


Image Source: https://medium.com/@ashok_/text-pre-processing-and-normalization-1054b043cff2

Keywords

- Corpus
- Token
- Tokenization
- Stop Words
- Stemming
- Lemmatizing
- Part-of-speech Tagging

Corpus

- Definition
  - Language resource consisting of a large and structured set of texts
  - In NLP, corpus becomes a dataset to train our model
    - This implies that we should extract samples from metadata with certain purpose and standards

Token & Tokenization

- Definition (Source: Stanford NLP Group)
  - An instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing
- Tokenization
  - Tokenization is a key (and mandatory) aspect of working with text data
  - We can't work with text data if we don't perform tokenization -> Inevitable task!
  - Example
    - "I want to go home" -> "I", "want", "to", "go", "home"

Reference: https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/

Stop Words

- Definition
  - Stop words are a set of commonly used words in any language.
    - This implies that they don't affect our analysis
- Why it's important?
  - If we remove the words that are very commonly used in a given language, we can focus on the important words instead
  - Example
    - Original: "I want to go home"
    - Removed: "I", "want", "go", "home"
  - Famous stop word lists
    - Snowball stop word list, Terrier stop word list

Source: https://kavita-ganesan.com/what-are-stop-words/#.YnN6ki8Rr0o

Stemming

- Definition (Source: Wikipedia)
  - Process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form.
- Why it's important
  - Not the format but the MEANING of the word is important
- Example
  - 'cat' for cats, catlike, and catty
  - 'fish' for fishing, fished, and fisher.
  - The stem need not be a word!
    - Ex) the Porter algorithm reduces, argue, argued, argues, arguing, and argus to the stem argu.

Lemmatizing

- Definition (Source: Wikipedia)
  - Process of grouping together the inflected forms of a word so they can be analyzed as a single item, identified by the word's lemma, or dictionary form
- Difference with Stemming
  - Depends on correctly identifying the intended part of speech and meaning of a word in a sentence, as well as within the larger context surrounding that sentence
- Example
  - The word "better" has "good" as its lemma. This link is missed by stemming
  - The word "meeting" can be either the base form of a noun or a form of a verb ("to meet") depending on the context; e.g., "in our last meeting" or "We are meeting again tomorrow".
    - Unlike stemming, lemmatization attempts to select the correct lemma depending on the context.

Part-of-speech Tagging (POS Tagging)

- Definition
  - Categorizing words in a text (corpus) in correspondence with a particular part of speech, depending on the definition of the word and its context.

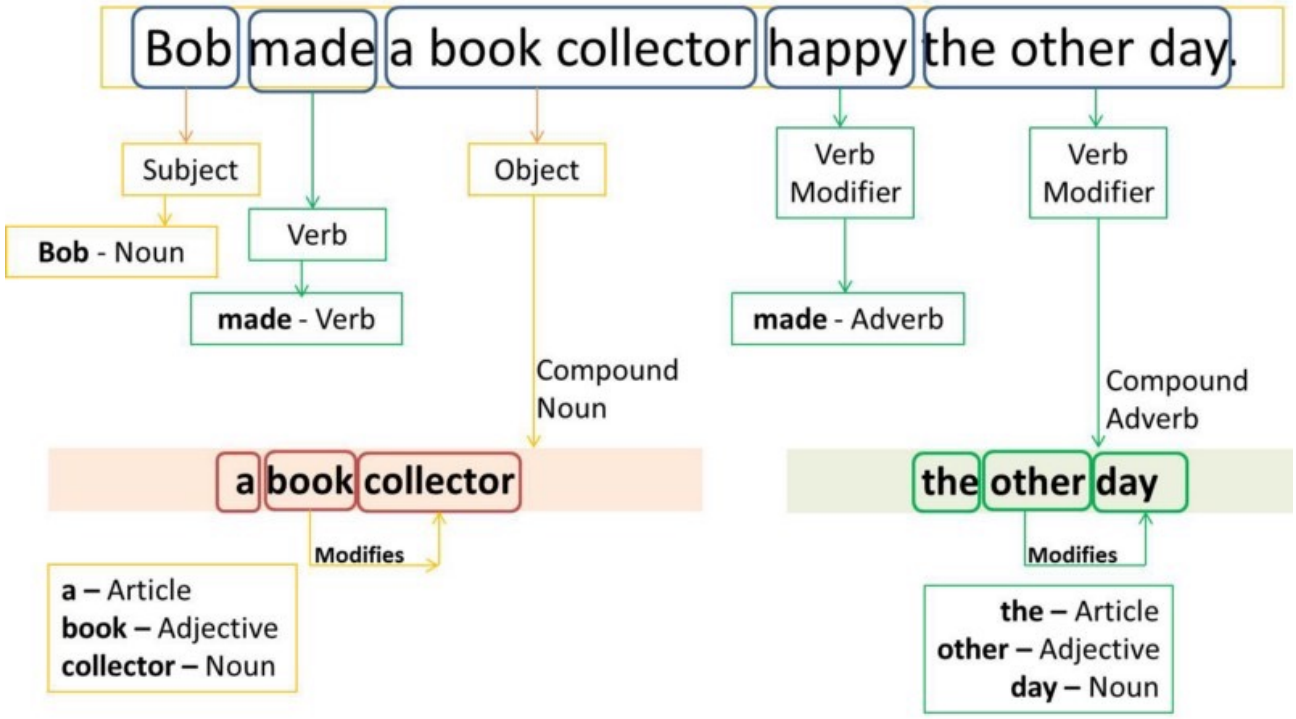
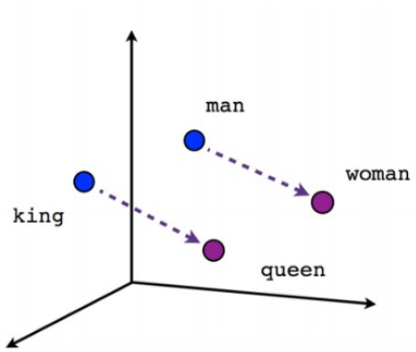


Image Source: https://www.freecodecamp.org/news/an-introduction-to-part-of-speech-tagging-and-the-hidden-markov-model-953d45338f24/

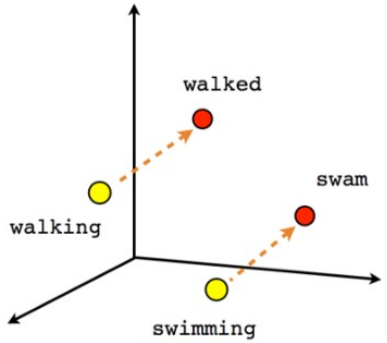
What else?

- Noise Removal
  - Removing some characters, punctuations, digits, and some texts that may disturb the analysis
    - This may be the first step of preprocessing
    - 'regex' is often used for this process
- Text Normalization
  - Process of transforming text into a single canonical form that it might not have had before. (Source: Wikipedia)
    - Normalizing text before storing or processing it allows for separation of concerns, since input is guaranteed to be consistent before operations are performed on it.
  - Example
    - Goooooooood!!! -> Good

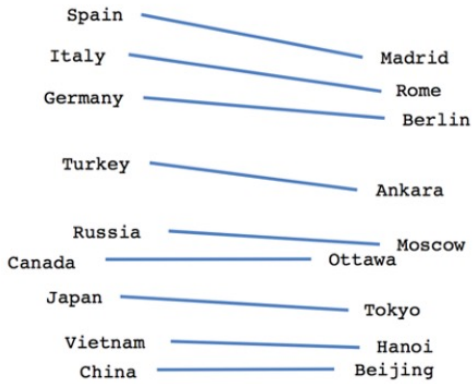
Word Embedding



Male-Female



Verb tense



Country-Capital

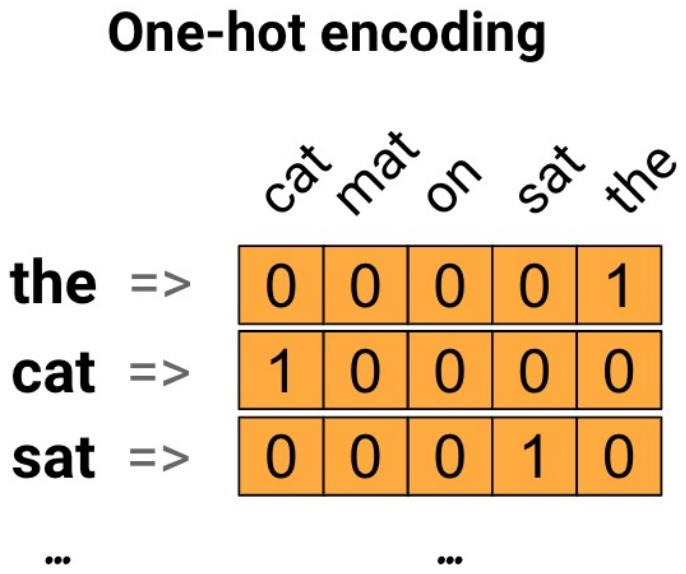
Image Source: https://towardsdatascience.com/creating-word-embeddings-coding-the-word2vec-algorithm-in-python-using-deep-learning-b337d0ba17a8

Embedding

- Definition (Source: Wikipedia)
  - In natural language processing (NLP), word embedding is a term used for the representation of words for text analysis, typically in the form of a real-valued vector that encodes the meaning of the word such that,
  - the words that are closer in the vector space are expected to be similar in meaning.
- Types
  - Sparse representation based
  - Frequency based
  - Prediction based
  - Frequency + Prediction based
  - Transformer Attention based (Next lecture)

Sparse Representation based Embedding

- One-Hot Encoding
  - Convert N vocabs into N dimensional vector
    - To represent each word, you will create a zero vector with length equal to the vocabulary, then place a one in the index that corresponds to the word



Reference: https://www.tensorflow.org/text/guide/word_embeddings

Problems with One-Hot Encoding

- Relationship between words
  - Inner product between two word vectors would always become 0
  - This implies that they are orthogonal, in other words independent
  - Thus, with one-hot encoding we cannot discover the relationship between words
- Curse of dimensionality
  - What if we use 1M words to create a sparse word matrix?

Frequency based Embedding

- Definition
  - Vectorize the text depending on the frequency of occurrence of the words in the text/ document
- How it works?
  - The words in the texts are vectorized using frequency based vectorizer
  - Then the documents are trained based on the vectors obtained by frequency vectorization
- Famous Vectorizers
  - Count Vectorizer (Bag of Words)
  - TF-IDF

Count Vectorizer

- How it works
  - Creates a matrix with documents and token counts (bag of terms/tokens)
  - This is often called 'Bag of words'
- Pros and Cons
  - Pros: Simple, Intuitive, Easy to implement
  - Cons: Ignores word orders
    - Ex) No employees have left the company == The company has no employees left
- In Python
  - Simply use CountVectorizer from sklearn.feature_extraction.text

TF-IDF

- what does TF-IDF stand for?
  - TF: Term Frequency
  - IDF: Inverse Document Frequency
- Idea
  - Keep the idea of bag of words but we should re-weight to reduce the importance of frequent words
  - We can shape the importance of repeated words within a single document, while reducing the impact of common words among a collection of documents

TF-IDF

- TF (Term Frequency)
  - Frequency of the word(w) in a single document(d)

$$tf(w, d) = f_{w,d}$$

  - We often apply a logarithm to the equation (to reduce the impact of repeated words)
  - Sometimes we normize the frequency by dividing with average word frequency

TF-IDF

- IDF (Inverse Document Frequency)
  - We need to reduce the impact of common words that appear in multiple documents
  - idf of word w is the inverse of the fraction of the documents containing that word

$$idf(w) = \frac{N_d}{df_w}$$

  - It's useful to add a logarithm to reduce the impact of very large idf
  - We also add one to the idf of equation so that we avoid zero idf values

TF-IDF

- TF-IDF is defined as product of term frequency and inversed document frequency

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

You can choose other options to normalize those values. Calibrate them to find the best option for your data!

Image Source: https://ted-mei.medium.com/demystify-tf-idf-in-indexing-and-ranking-5c3ae88c3fa0

TF-IDF

- Possible Usages
  - Search Engine based on keyword search
  - Keyword analysis
  - Search Engine Optimization
- In Python
  - Simply import TfidfVectorizer from sklearn.feature_extraction.text

Prediction based

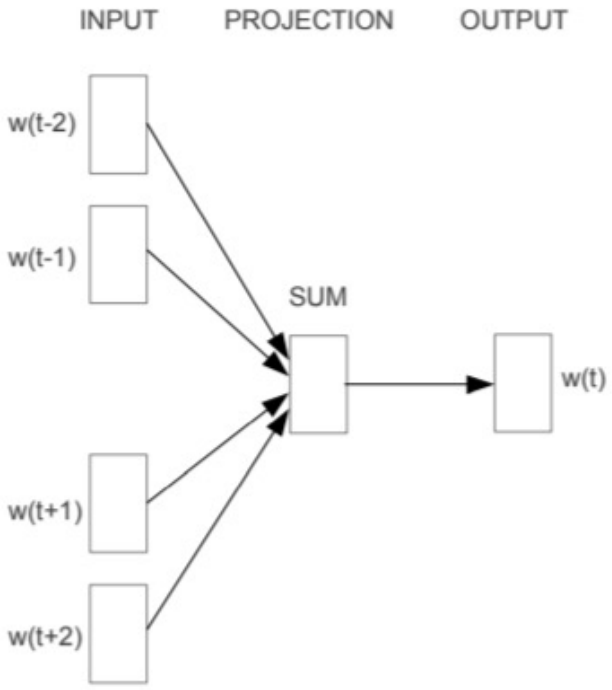
- Definition
  - Use neural network to predict which word will appear in certain contexts
- Word2Vec (2013) (Source: Wikipedia)
  - uses a neural network model to learn word associations from a large corpus of text.
  - represents each distinct word with a particular list of numbers called a vector
    - Vectors are chosen carefully such that a simple mathematical function (the cosine similarity between the vectors) indicates the level of semantic similarity between the words represented by those vectors.
  - Word2Vec is not a singular algorithm, rather, it is a family of model architectures and optimizations that can be used to learn word embeddings from large datasets!

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

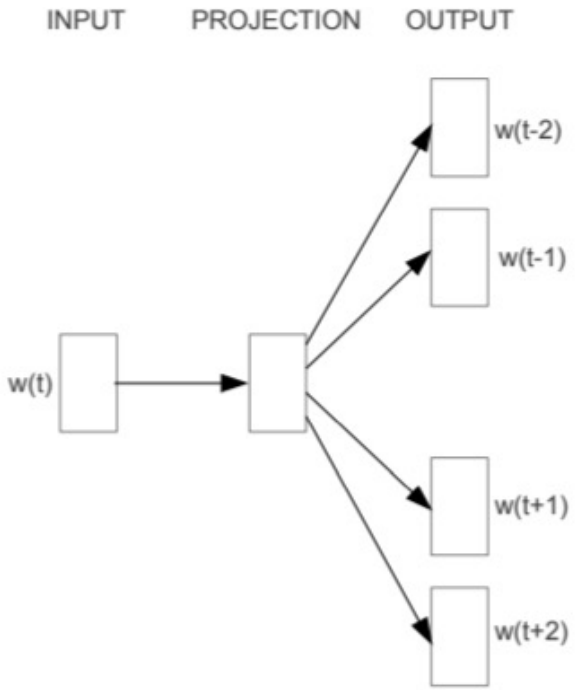
How to learn representation of words?

- CBOW
  - Continuous Bag of Words
  - Goal: predict the middle word based on surrounding context words
  - The context consists of a few words before and after the current (middle) word.
  - Example
    - Original Sentence: Calm cat slept of the sofa
    - Task: Predict 'slept'
- Skip-gram
  - Goal: predicts words within a certain range before and after the current word in the same sentence
  - Example
    - With 'slept' predict surrounding words

CBOW and Skip-gram



CBOW



Skip-gram

Image Source: https://arxiv.org/pdf/1309.4168v1.pdf

Prediction + Frequency based Embedding

- GloVe
  - Global Vectors for Word Representation
  - Word embedding with the information of global co-occurrence statistics
    - skip-gram + statistical methods
  - Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.
  - We can't say that GloVe is better than Word2Vec! It's circumstantial
  - You can download global vector for word representations from the link below
    - https://nlp.stanford.edu/projects/glove/

Exercises

Image Source: https://community.atlassian.com/t5/Jira-Service-Management/Way-to-get-a-WORDCLOUD-from-all-JIRA-tickets-for-Top-topics/qaq-p/1281903

What we are going to do

- SEC filing analysis
  - For those of you who are not familiar with SEC filings please check the links below
    - https://www.sec.gov/edgar/search-and-access
    - https://www.investopedia.com/articles/fundamental-analysis/08/sec-forms.asp
- IMDB review data sentimental analysis
  - We will use RNN for this task
- Naver Financial News Keyword Extraction
  - This is a very simple example so I hope you can put to use what you have learned in this session