# TEXT AUGMENTATION

## OUTLINE

1. Idea
2. Some techniques
   - Thesaurus
   - Word embeddings
   - Back translation
   - Contextualize word embeddings
   - Text generation
   - Random deletion
3. Some wellknown researches
   - In English
   - In Vietnamese
4. References

# IDEA

- More data we have, better model, performance we can achieve.
- Training data is **NOT** unlimited.
- We need a techniques to generate more data.

# SOME TECHNIQUES

## Thesaurus

**Definition (Thesaurus)**

Replace non-stop words or phrase with their synonyms.

# SOME TECHNIQUES

## Thesaurus

**Definition (Thesaurus)**

Replace non-stop words or phrase with their synonyms.

**Definition (Stop word)**

Words which are filtered out before processing of natural language data.

# SOME TECHNIQUES

## Thesaurus

**Definition (Thesaurus)**

Replace non-stop words or phrase with their synonyms.

**Definition (Stop word)**

Words which are filtered out before processing of natural language data.

```
1         import nltk
2         from nltk.corpus import stopwords
3         nltk.download('stopwords') # run only the first time
4         print(set(stopwords.words('english')))
5         # {'the', 'they', 'of', 'you', 'each', 'some', 'be', 'down
      ', 's', 're', 'between', 'we', "mustn't", 'so', ... }
6         print(len(set(stopwords.words('english'))))
7         # 179
8
```

Listing 3: Stop words in English

# SOME TECHNIQUES

**Thesaurus** - Automatic thesaurus generation

There are two main approaches:

# SOME TECHNIQUES

**Thesaurus** - Automatic thesaurus generation

There are two main approaches:

- Exploit word co-occurrence

# SOME TECHNIQUES

## **Thesaurus** - Automatic thesaurus generation

There are two main approaches:

- Exploit word co-occurrence
- Use a shallow grammatical analysis

# SOME TECHNIQUES

## **Thesaurus** - Automatic thesaurus generation

There are two main approaches:

- Exploit word co-occurrence
- Use a shallow grammatical analysis

| Word | Nearest neighbors |
|---|---|
| absolutely | absurd, whatsoever, totally, exactly, nothing |
| bottomed | dip, copper, drops, topped, slide, trimmed |
| captivating | shimmer, stunningly, superbly, plucky, witty |
| doghouse | dog, porch, crawling, beside, downstairs |
| makeup | repellent, lotion, glossy, sunscreen, skin, gel |
| mediating | reconciliation, negotiate, case, conciliation |
| keeping | hoping, bring, wiping, could, some, would |
| lithographs | drawings, Picasso, Dali, sculptures, Gauguin |
| pathogens | toxins, bacteria, organisms, bacterial, parasite |
| senses | grasp, psyche, truly, clumsy, naive, innate |

▶ **Figure 9.4** An example of an automatically generated thesaurus. This example is based on the work in Schütze (1998), which employs latent semantic indexing (see Chapter 18 ).

# SOME TECHNIQUES

## Word embeddings

**Definition (Word embeddings)**

Represent a word by a high-dimensional vector where each dimension represents a value of a property

# SOME TECHNIQUES

## Word embeddings

**Definition (Word embeddings)**

Represent a word by a high-dimensional vector where each dimension represents a value of a property

Example :

$$\text{Guitar} = \begin{pmatrix} 0.12 \\ 0 \\ 0 \\ 0.74 \\ -0.89 \end{pmatrix} \quad \text{Piano} = \begin{pmatrix} 0.67 \\ 0.8 \\ 0 \\ 0.22 \\ -0.80 \end{pmatrix}$$

# SOME TECHNIQUES

## Word embeddings

### Definition (Word embeddings)

Represent a word by a high-dimensional vector where each dimension represents a value of a property

Example :

$$\text{Guitar} = \begin{pmatrix} 0.12 \\ 0 \\ 0 \\ 0.74 \\ -0.89 \end{pmatrix} \text{Piano} = \begin{pmatrix} 0.67 \\ 0.8 \\ 0 \\ 0.22 \\ -0.80 \end{pmatrix} \Longrightarrow \text{cos\_sim} = 0.6157$$

# SOME TECHNIQUES

## Word embeddings

**Definition (Word embeddings)**

Represent a word by a high-dimensional vector where each dimension represents a value of a property

Example :

$$\text{Guitar} = \begin{pmatrix} 0.12 \\ 0 \\ 0 \\ 0.74 \\ -0.89 \end{pmatrix} \quad \text{Piano} = \begin{pmatrix} 0.67 \\ 0.8 \\ 0 \\ 0.22 \\ -0.80 \end{pmatrix} \quad \Longrightarrow \text{cos\_sim} = 0.6157$$

Some useful models for word embeddings: word2vec, GloVe, fastText

| word2vec | GloVe | fasttext |
|---|---|---|
| foxes | nbc | henhouse |
| squirrel | abc | foxes |
| rabbit | cbs | hare |
| squirrels | turner | Fox |
| coyote | disney | fennec |

Figure: Most similar words of *fox* among classical word embeddings models

# SOME TECHNIQUES

## **Word embeddings** - Gensim Word2Vec Model

- Install `gensim, beautifulsoup4` library
  `pip install gensim beautifulsoup4`

```
1         #import libraries
2     from gensim.models import Word2Vec
3     import bs4 as bs
4
```

Listing 4: Library importation

# SOME TECHNIQUES

## **Word embeddings** - Gensim Word2Vec Model

- Install `gensim, beautifulsoup4` library
  `pip install gensim beautifulsoup4`

```
1          #import libraries
2      from gensim.models import Word2Vec
3      import bs4 as bs
4
```

Listing 5: Library importation

- Other utility : `nltk.stem.PorterStemmer`, `nltk.stem.WordNetLemmatizer`
  Example : (programs, programming, programed, ...) = program

# SOME TECHNIQUES

## Back translation

**Definition (Back translation)**

Translate target language to source language and mixing both original source sentence and back-translated sentence to train a model

# SOME TECHNIQUES

## Back translation

**Definition (Back translation)**

Translate target language to source language and mixing both original source sentence and back-translated sentence to train a model

Example :
English : I play soccer
Vietnamese : $\Rightarrow$ Tôi chi bóng đá
English : $\Rightarrow$ I play football

# SOME TECHNIQUES

## Text generation

**Definition (Text generation)**

Generate text with the goal of appearing indistinguishable to human-written text.

# SOME TECHNIQUES

## Text generation

**Definition (Text generation)**

Generate text with the goal of appearing indistinguishable to human-written text.

Some useful libraries/techniques :

- LSTM Recurrent Neural Networks in Python with Keras
- N-gram, RNNs, GRUs, LSTMs, seq2seq(Conditional Language Model)

# SOME TECHNIQUES

## Text generation

**Definition (Text generation)**

Generate text with the goal of appearing indistinguishable to human-written text.

Some useful libraries/techniques :

- LSTM Recurrent Neural Networks in Python with Keras
- N-gram, RNNs, GRUs, LSTMs, seq2seq(Conditional Language Model)

Example :
When we type : as soon as

# SOME TECHNIQUES

## Text generation

**Definition (Text generation)**

Generate text with the goal of appearing indistinguishable to human-written text.

Some useful libraries/techniques :

- LSTM Recurrent Neural Networks in Python with Keras
- N-gram, RNNs, GRUs, LSTMs, seq2seq(Conditional Language Model)

Example :
When we type : as soon as
Google suggest : as soon as|*possible*

# SOME TECHNIQUES

## Random deletion

**Definition**

Randomly remove one (or many) word(s) from a sentence to create a now sentence.

# SOME TECHNIQUES

## Random deletion

**Definition**

Randomly remove one (or many) word(s) from a sentence to create a now sentence.

Example :
Original: The quick brown fox jumps over the lazy dog
Augmented Text: The fox jumps over the lazy dog

# SOME WELLKNOWN RESEARCHES

- In English
  - Gmail suggestion while composing.
  - Google BERT
  - Facebook RoBERTa.
  - Amazon sagemaker.

# SOME WELLKNOWN RESEARCHES

- In English
  - Gmail suggestion while composing.
  - Google BERT
  - Facebook RoBERTa.
  - Amazon sagemaker.
- In Vietnamese
  - Vietnam Language and Speech Processing (https://vlsp.hpda.vn/)

# REFERENCES


1. https://towardsdatascience.com/data-augmentation-library-for-text-9661736b13ff
2. https://towardsdatascience.com/data-augmentation-in-nlp-2801a34dfc28
3. https://www.kilgarriff.co.uk/Publications/2003-K-Beijing-thes4NLP.pdf
4. https://vlsp.hpda.vn/demo
5. https://nlp.stanford.edu/IR-book/html/htmledition/automatic-thesaurus-generation-1.html
6. https://stackabuse.com/implementing-word2vec-with-gensim-library-in-python/
7. https://towardsdatascience.com/exploring-wild-west-of-natural-language-generation-from-n-gram-and-rnns-to-seq2seq-2e816edd89c6
8. https://docs.aws.amazon.com/sagemaker/latest/dg/whatis.html

# LINEAR REGRESSION

## Idea

- Linear regression is a linear approach to modeling the relationship between a scalar response and one or more explanatory variables.
- Type of linear regression :
  - Simple linear regression : $\hat{y} = xw + b$
    where $\hat{y}, x, w, b$ is a scalar variable.
  - Multivariate linear regression : $\hat{y} = xw + b = \bar{x}w$
    where $w, x$ are vectors, $\hat{y}, b$ is a scalar number.