Natural Language Processing

What are we going to cover?

1. What is NLP?
2. Brief History of NLP
3. NLP Use Cases
4. NLP in Healthcare
5. Word Embeddings
6. Q & A

What is NLP?

What is NLP?

- We are constantly shaping our environment to our human needs and this comes in many forms, one of them is making machines understand the most complex and special feature that set us apart from mere giant apes' “speech”.
- Speech is the most effective way of communication and it’s only normal that we want our inventions to understand it.
- Natural Language Processing is the technology used to aid computers to understand the human’s natural language.

Brief History of NLP

NLP Timeline

1950

Alan Turing published an article titled “Computing Machinery and Intelligence” which is now called Turing Test is a test of a machine’s ability to exhibit intelligent behavior.

1980

There were complex handwritten rules to accomplish NLP tasks and Machine Learning came into picture. Linguistics, Grammar parsing etc.

2010

Deep learning(DL) took over and deep neural network-style ML methods became widespread in natural language processing.

2020

With the power of DL models like RNNs, Transformers , Encoders etc.., we can process large text and with a much greater accuracy than ever before.

NLP Use Cases

Common Applications

- Sentiment Analysis
- NER
- Text Classification
- Auto-Correct
- Speech Recognition
- Spell Check
- Machine Translation
- Chatbots

NLP in Healthcare

Information Extraction from Medical Documents

Vast amounts of medical information are still recorded as unstructured text. The knowledge contained in this textual data has a great potential to improve clinical routine care, to support clinical research, and to advance personalization of medicine.

. In Information extraction, we are trying to understand the text and extract important entities present in it. The Training obviously require Labelled data. There are broadly 3 steps in IE.

1. NER -> Named Entity Recognition , we are trying to label each word in the text.
2. Coref-Resolution -> Same entities are grouped together.
3. Relation Extraction -> Predicting relationships between the entities.

Information Extraction from Medical Documents

- Let’s take an example:

This report was received on 25-AUG-2015. A physician reported that a 40-year-old female patient experienced non-serious aggravated renal function after initiating Amoxicillin. On 26-NOV-2014 the patient started on amoxicillin 22.5 mg/day for ADPKD. On 05-FEB-2015 the patient experienced aggravated renal function. On the same day the dose of drug was reduced to 15 mg/day. The outcome of the event was reported as "not resolved" at the time of this report.

First, NER runs on the text, and different entities as highlighted are extracted. Each entity is classified with some pre-defined label.

The predefined labels for this example are:

Receipt Date Reporter Patient Age Patient Gender AE seriousness
AE name Drug Name Drug date Dosage Indication AE date AE outcome

Information Extraction from Medical Documents

- This report was received on 25-AUG-2015. A physician reported that a 40-year-old female patient experienced non-serious aggravated renal function after initiating Amoxicillin. On 26-NOV-2014 the patient started on amoxicillin 22.5 mg/day for ADPKD. On 05-FEB-2015 the patient experienced aggravated renal function. On the same day the dose of drug was reduced to 15 mg/day. The outcome of the event was reported as "not resolved" at the time of this report.

Coref Resolution is applied on extracted Entities to group them and is then used for finding relationship between them. For instance, here adverse event 'aggravated renal function' is related with '05-FEB-2015' .
Also, the entity 'same day' and '05-FEB-2015' are same, we can say that dosage '15 mg/day' is related to '05-FEB-2015'. Similarly, event 'aggravated renal function' and event outcome 'not resolved' are related.

Information Extraction from Medical Documents

- Finally, the unstructured data could be converted into Structured format.

Receipt Date	Reporter
25-AUG-2015	Physician

Patient Age	Patient Gender
40 year	Female

AE Name	AE Onset Date	AE Seriousness	AE outcome
aggravated renal function	05-FEB-2015	Non-serious	Not-resolved

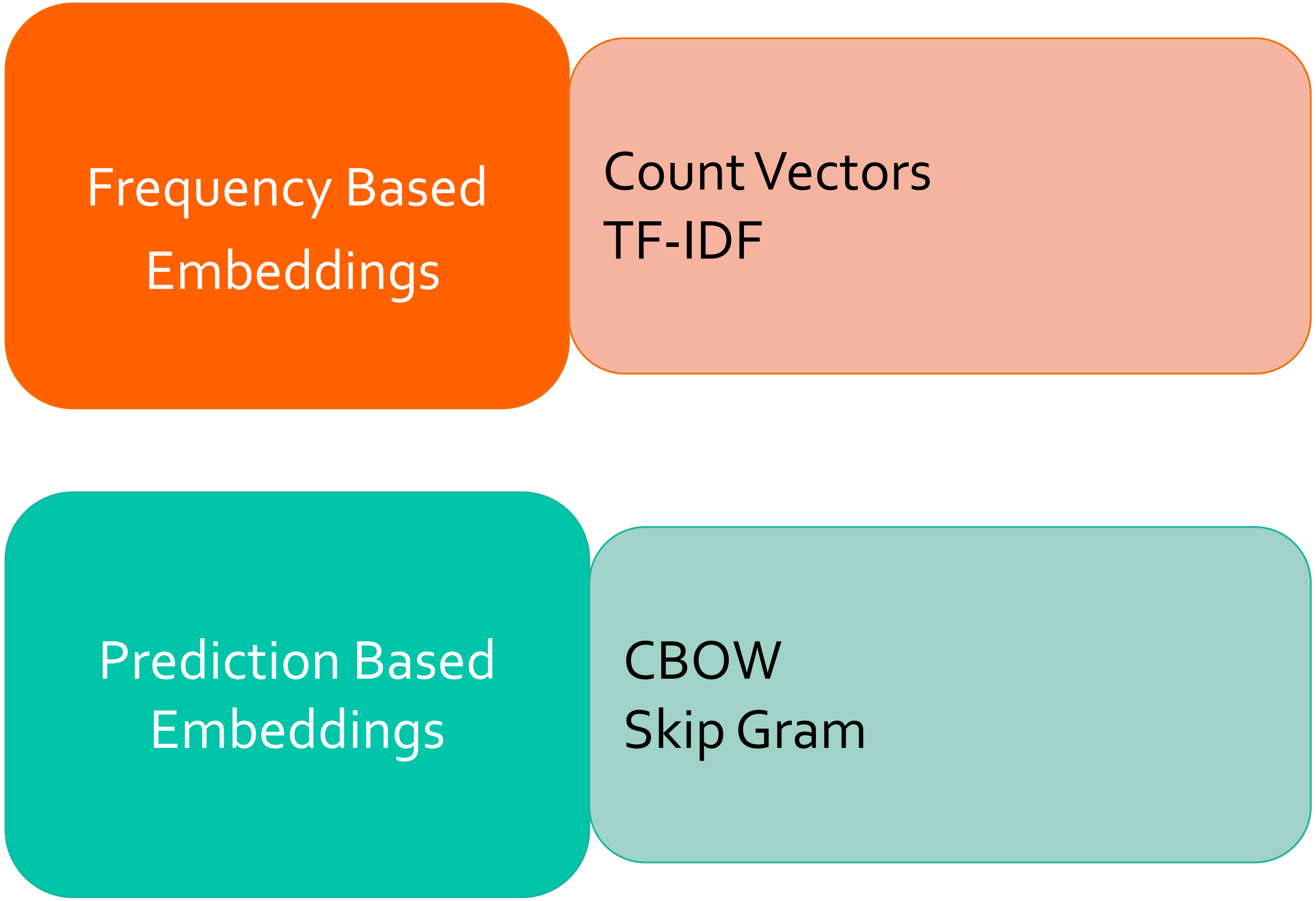
Drug	Amoxicillin
Indication	ADPKD
Dosage	**Dose Start Date**
22.5mg/day	26-NOV-2014
15mg/day	05-FEB-2015

Word Embeddings

Why word embeddings?

- NLP deals with text , which is itself composed of smaller units like words and characters. Because of so many languages present in world, approx. 65000,there is a need to produce a standard approach that all languages could be mapped to.
- Since our computers, scripts and Machine Learning models can’t read and understand text in any human sense. But could only work on vectors / numeric values. So, each word is converted into a vector for further processing. Those vectors are known as word embeddings.
- There are various ways in which a word could be converted into its vector form. Let's go over some of them.

Techniques to represent word as vectors



Frequency Based Embeddings

Count Vectors
TF-IDF

Prediction Based Embeddings

CBOW
Skip Gram

Count Vectors

- Count vector model learns a vocabulary from all the documents, then models each document by counting the number of times each word appears
- For example, consider we have **D** documents and **T** is the number of different words in our vocabulary then the size of count vector matrix will be given by D*T .
- Let's understand this through code:
- https://colab.research.google.com/drive/1FJ3V6ZNeAKsyJdeUtYQ14-JMYLSZ8mj4?usp=sharing

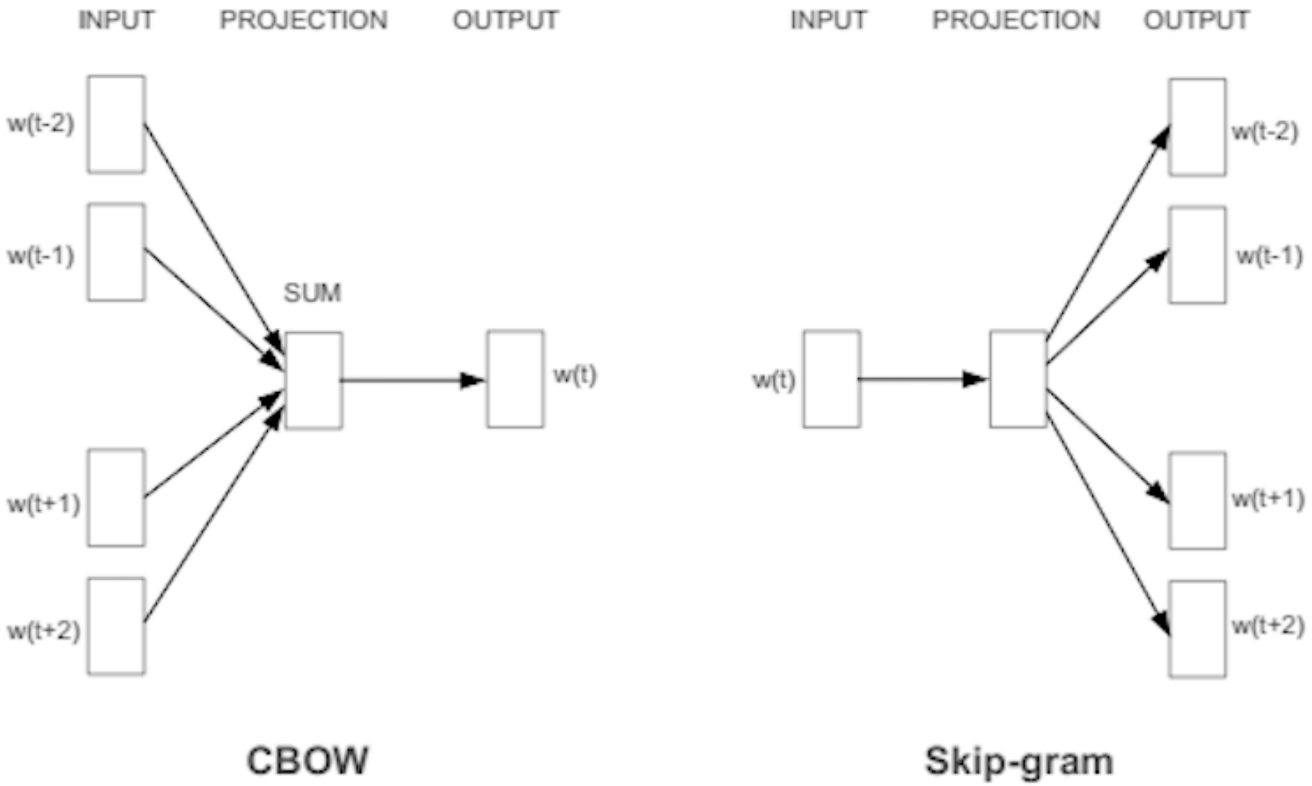
TF-IDF Vectors

- In TF-IDF, along with Term Frequency(TF), IDF (inverse document frequency) is also calculated. What TF-IDF does is it balances out the term frequency (how often the word appears in the document) with its inverse document frequency (how often the term appears across all documents in the data set). This means that words like “a” and “the” will have very low scores as they’ll appear in all documents in your set.
- TF = (Number of times term t appears in a document)/(Number of terms in the document)
- IDF = log(N/n), where N is the total number of documents and n is the number of documents a term t has appeared in.
- TF-IDF(t, document) = TF(t, document) * IDF(t)

Prediction Based Embeddings

- Frequency based embeddings are usually very sparse whereas the prediction-based embeddings are dense and doesn’t increase with the increase in vocabulary.
- The distributed representation is learned based on the usage of words. This allows words that are used in similar ways to result in having similar representations, naturally capturing their meaning.
- There is deeper linguistic theory behind the approach, namely the “distributional hypothesis” by Zellig Harris that could be summarized as: words that have similar context will have similar meanings.
- You shall know a word by the company it keeps!

Prediction Based Embeddings



CBOW Model Implementation

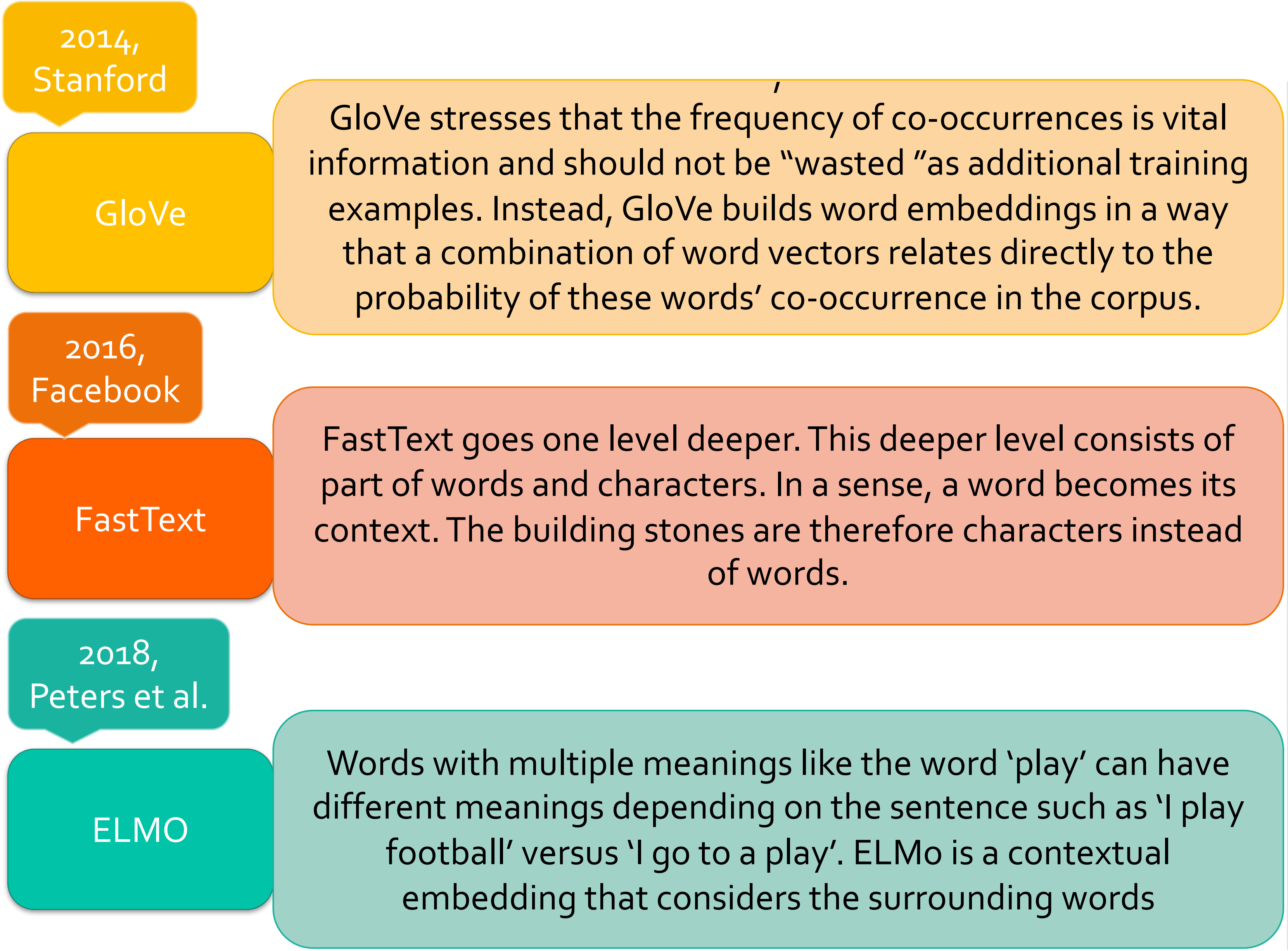
- https://colab.research.google.com/drive/1cwXyhQMOInlq9Xgh-Tat-Eulo3B6oScS?usp=sharing

Skip –Gram Model Implementation

- https://colab.research.google.com/drive/17sVgSHPtlzX6erexHg5KaGRTeq9fgvS3?usp=sharing

Some more State-of-the-Art Word2Vec Models

Word2Vec Models



2014, Stanford

GloVe

GloVe stresses that the frequency of co-occurrences is vital information and should not be “wasted ”as additional training examples. Instead, GloVe builds word embeddings in a way that a combination of word vectors relates directly to the probability of these words’ co-occurrence in the corpus.

2016, Facebook

FastText

FastText goes one level deeper. This deeper level consists of part of words and characters. In a sense, a word becomes its context. The building stones are therefore characters instead of words.

2018, Peters et al.

ELMO

Words with multiple meanings like the word ‘play’ can have different meanings depending on the sentence such as ‘I play football’ versus ‘I go to a play’. ELMo is a contextual embedding that considers the surrounding words

Q & A