
NLP
Overview
Barry Trim
25th Nov 2021
Stockholm Public Library

“When I use a word,’ Humpty Dumpty said in rather a scornful tone, ‘it means just what I choose it to mean — neither more nor less.’

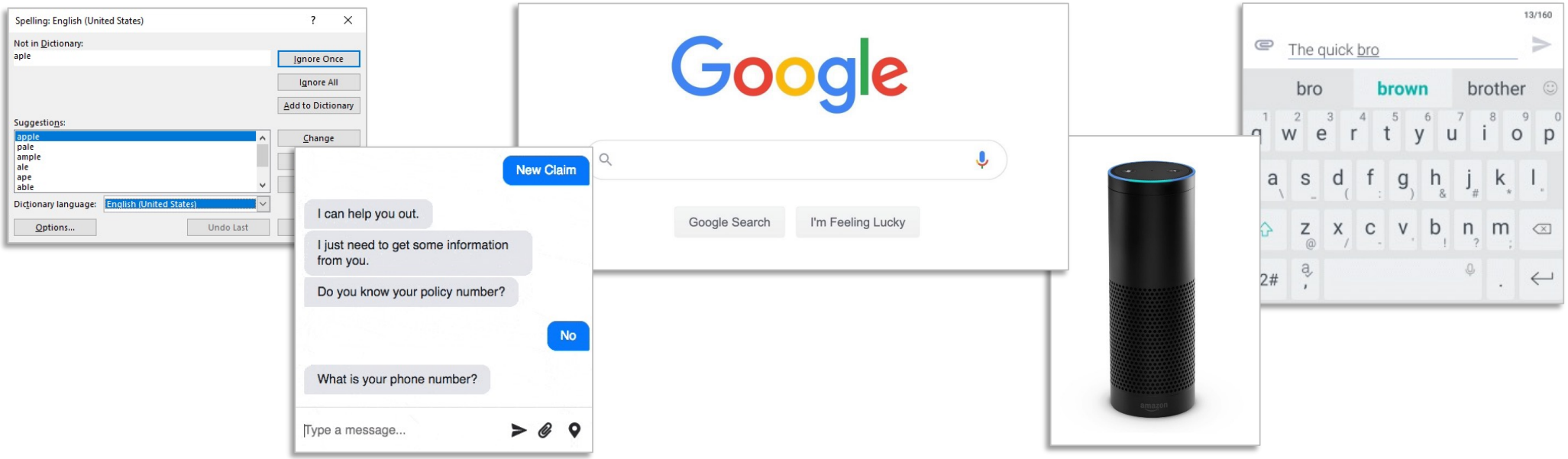
Humpty Dumpty speaking to Alice
Through the Looking Glass, Lewis Carroll

http://mural.uv.es/masanar/10.html

NLP OVERVIEW 25.11.21

WHAT IS NLP?

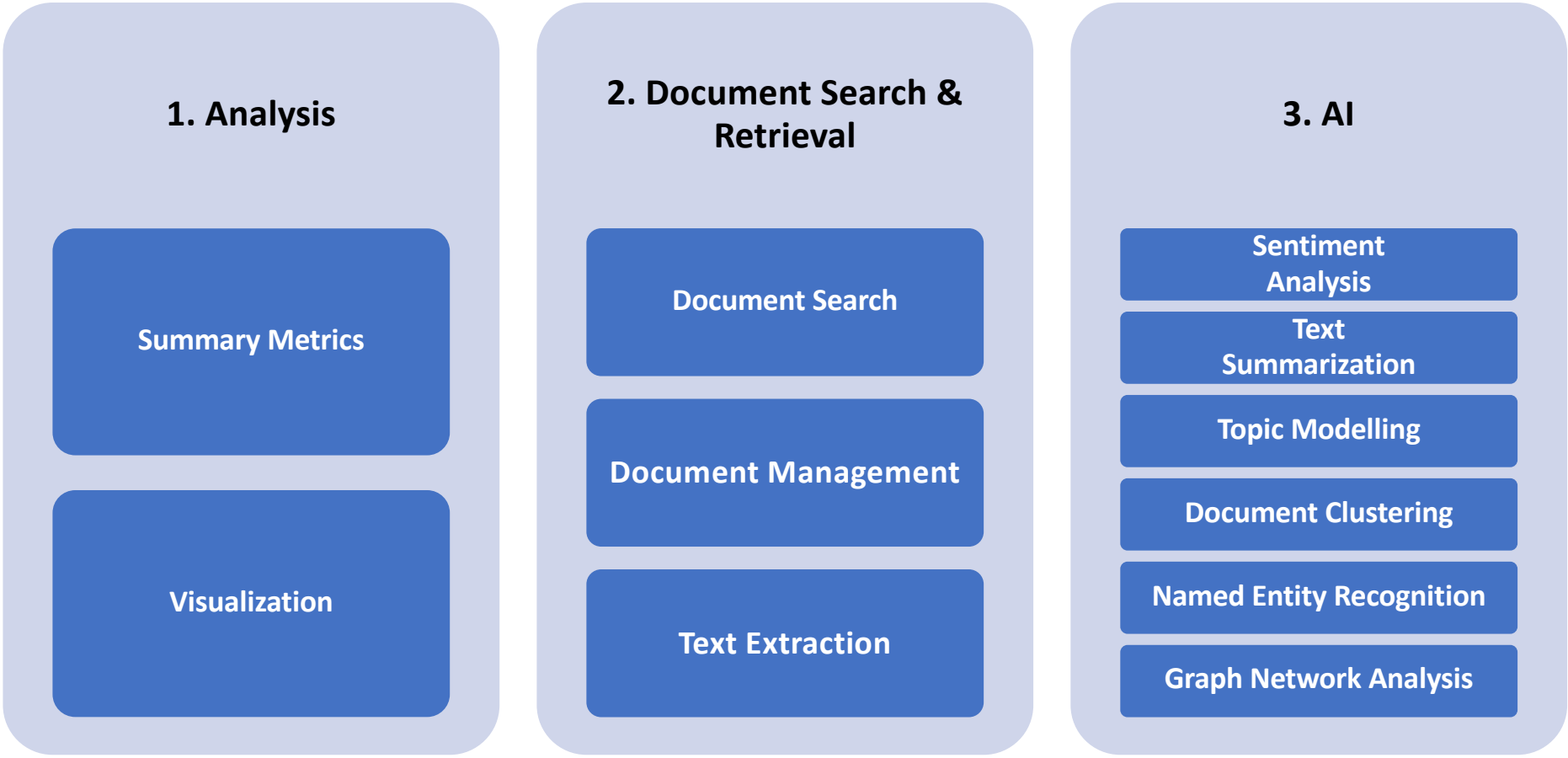
Natural Language Processing is a branch of Artificial Intelligence (AI) that enables computers to understand and process natural human language in a meaningful way



NLP

- Natural language processing (NLP) is a branch of artificial intelligence that helps computers understand, interpret and manipulate human language. NLP draws from many disciplines, including computer science and computational linguistics, in its pursuit to fill the gap between human communication and computer understanding
- Python NLP Library – NLTK (Natural Language Tool Kit)
  - Take a look at the NLTK book @ https://www.nltk.org/book/
- NLP Book)
  - Speech and Language Processing by Daniel Jurafsky
  - and James H. Martin
  - https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf

WHAT IS NLP?


1. Analysis
Summary Metrics
Visualization
2. Document Search & Retrieval
Document Search
Document Management
Text Extraction
3. AI
Sentiment Analysis
Text Summarization
Topic Modelling
Document Clustering
Named Entity Recognition
Graph Network Analysis

WHAT IS NLP?

Method	Overview	Description	Benefits
1. Analysis	**Summary Metrics**	• Term Frequency: Word\phrase count • Document Scan: High-level detail of documents contained within a repository	• Fast high-level understanding of large document repositories • Useful for data migration, cleansing and storage activities • Input for advanced AI models
	Visualization	• Graphical plots and chart of document and text	
2. Document Search & Retrieval	**Document Search**	• Searching of massive document repositories, ranging from full text search to ccomplex Boolean searches, fuzzy searches and relevance searches (searching for concept the particular words)	• Ability to find specific documents in a massive document corpus
	Document Management	• Document organization, de-duplication	• Improves document search and reduces document storage needs
	Text Extraction	• Extracting document text for analysis or alternative uses	• Provides ability to capture tables from design specifications to use a data source for calculations
3. AI	**Sentiment Analysis**	• Detection of emotion, positive and negative sentiment in within text	• Provides high-level perception and opinion or particular topics and initiatives, e.g. employee satisfaction, public consultation etc
	Text Summarization	• Technique for generating a concise and precise summary of voluminous texts	• Reduces time to review and process documents
	Topic Modelling	• Finding latent clusters of topics within a text corpus	• Provides insight on topics currently being discussed • Enables sifting through large text datasets to identify the most important topics in a scalable and accurate way
	Document Clustering Modelling	• Finding latent clusters of documents within a text corpus	• Enable faster processing and sorting of documents
	Named Entity Recognition	• Identifying things (nouns), e.g. people, places, objects within a document	• Helps identify the key elements in a text (people, places, companies etc.) and detect important information
	Graph Network Analysis	• Nodal graph analysis of entities contained within text or between documents	• Show relationships between entities, useful for understand interactions between people, departments and entities

NLP PROCESS STEPS

Domain Expertise

Leverage domain expertise and knowledge from Client and Arup Business Teams

1. Data Collection

Locate & ingest relevant text data sources

2. Data Preparation

Prepare the data for modelling

3. Modelling

Create AI NLP models

4. Analyse

Analyse and present model output

PREPROCESSING

Prepare data for modelling

#	Technique	Overview
1	**Extract Metadata**	• Extract metadata from the text
2	**Delimit**	Delimit document boundaries for to enhance analysis & modelling, the following approaches may be used • **Keep Existing**: Keep existing document boundaries • **Group**: Group documents based on specific criteria (e.g. data\time, user) • **Split**: Split documents based on specific criteria (e.g. sentences, paragraphs, sections, pages, chapters etc)
3	**Tokenize**	• Split sentences \ documents down to it's atomic elements for modeling (un-grams, bi-grams, n-grams, noun chunks)
4	**Normalise**	• Set text to lowercase, may impact sentiment analysis
5	**Spelling Check**	• Identify and address misspelt words
6	**Stop Characters**	• Remove unnecessary characters, e.g. text formatting (\n, \c, bullet points etc), non-alphanumeric characters
7	**Stop Words**	Filter unnecessary words, i.e. • Words which occur less than n-times • Words which occur frequently but carry little meaning, e.g. 'and, if, the, there, this' etc • TF-IDF (Term Frequency-Inverse Document Frequency) - Works by increasing proportionally to the number of times a word appears in a document, but is offset by the number of documents that contain the word. So, words that are common in every document, such as this, what, and if, rank low even though they may appear many times, since they don't mean much to that document in particular.
8	**POS Filtering**	• Filter in key POS attributes (Nouns, Adjectives, Verbs, Adverbs, Numbers)
9	**Lemmatization**	• Considers the context (how the word is used in context) and converts the word to its meaningful base form (called Lemma)
10	**Stemming**	• Removes or stems the last few characters of a word, often leading to incorrect meanings and spelling

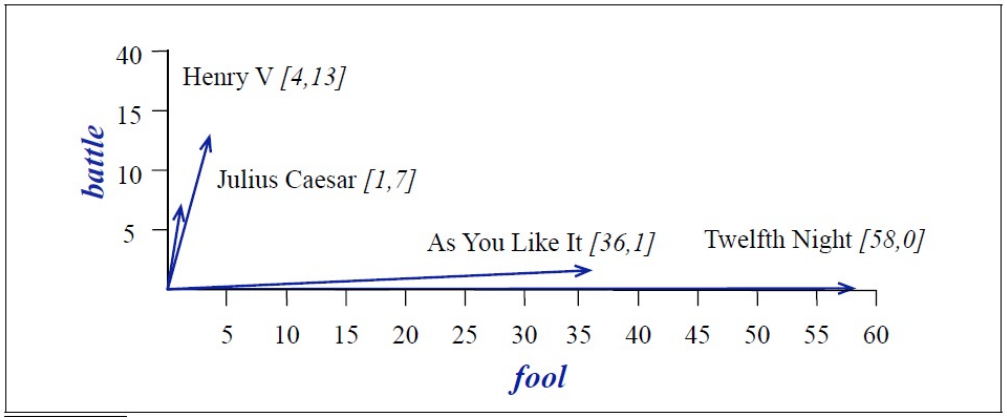
EMBEDDING

- Embedding is converting ‘Text’ to numerical vectors which represent the words
- Text Embedding causes models to function better as embedding models (such as BERT) can take into account content indicate similarity of meaning between two different words (e.g. ‘car’ and ‘automobile’ meaning the same thing

Simple Embedding Example

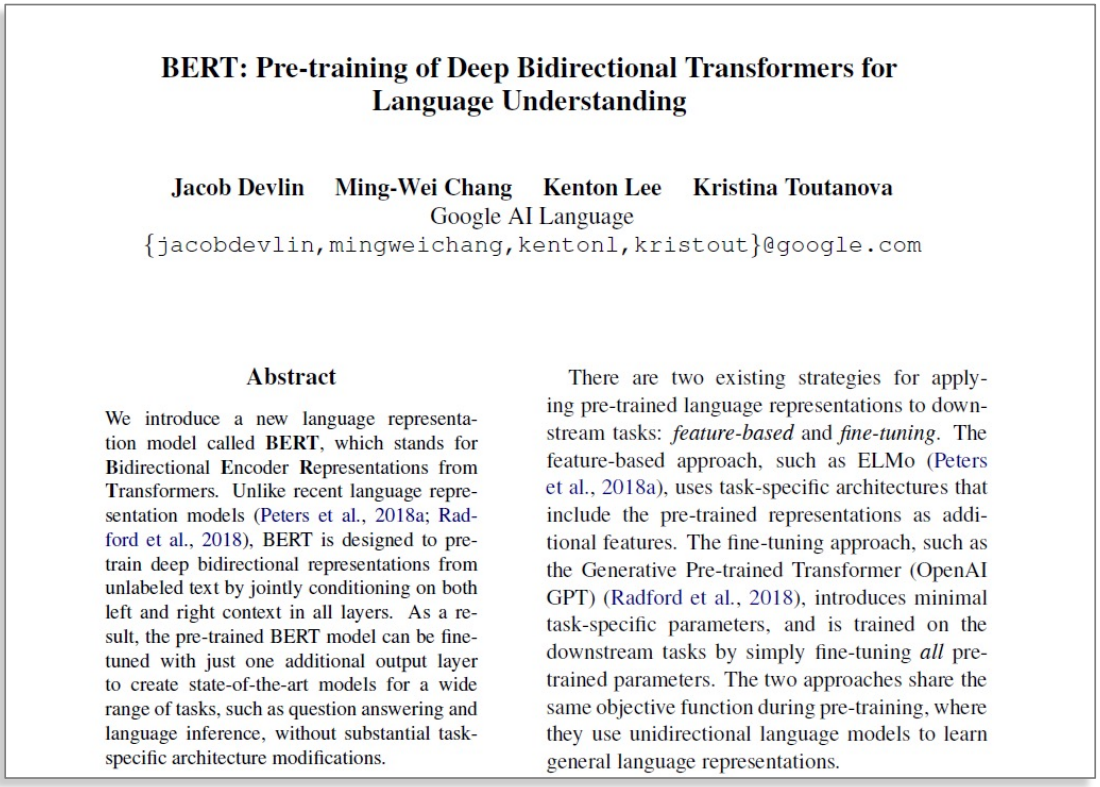
	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Term Document Embedding



Vector Similarity

BERT - BIDIRECTIONAL ENCODER REPRESENTATIONS



BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

https://arxiv.org/pdf/1810.04805.pdf

- Pre Trained NLP Transformer Model
- Trained on Wikipedia(that's 2,500 million words!) and Book Corpus (800 million words)
- Used to calculate word embeddings for text corpus
- Vector length 768
- 2 versions Base and Large Models
- A number of Python\Spark Libraries can run the BERT model for predicting
- Transfer Learning – models learned on one task and applied to another task

BERT - BIDIRECTIONAL ENCODER REPRESENTATIONS

- Words or text strings are feed into the BERT and an embedding (numerical vector is produced)
- The embeddings are the Neural Network weights