Using Text Analysis And Data Modeling To Understand Big Data

Maryam Jahanshahi Ph.D.
Research Scientist

tapRecruit.co

Content matters in job descriptions

Same title, Different job

Finance Manager Kraft Foods	Finance Manager Roche
Junior (3 Years)	Senior (6-8 Years)
No Managerial Experience	Division Level Controller
	Strategic Finance Role
	MBA / CPA

- ✓ **Same Title**
- ✗ Required Experience
- ✗ Required Responsibility
- ✗ Preferred Skill
- ✗ Required Education

Different title, Same job

Performance Marketing Manager PocketGems	Senior Analyst, Customer Strategy The Gap	
Mid-Level	Mid-Level	✓ **Required Experience**
Quantitative Focus	Quantitative Focus	✓ **Required Skills**
iBanking Expertise	Finance Expertise	✓ **Required Experience**
Data Analysis Tools (SQL)	Relational Database Experience	✓ **Required Skills**
Consulting Experience Preferred	External Consulting Experience Preferred	✓ **Preferred Experience**
MBA Preferred	BA in Accounting, Finance, MBA Preferred	✓ **Preferred Education**

tapRecruit.co

Today:

1. Designing Text Preprocessing Pipelines
2. Transforming Text into Feature Representations
3. Approaches to Classify Documents

Typical Questions when Analyzing Text

Data Resolution:
Are these two pieces of data the same?

"Software Engineer"
"Software Enginere"

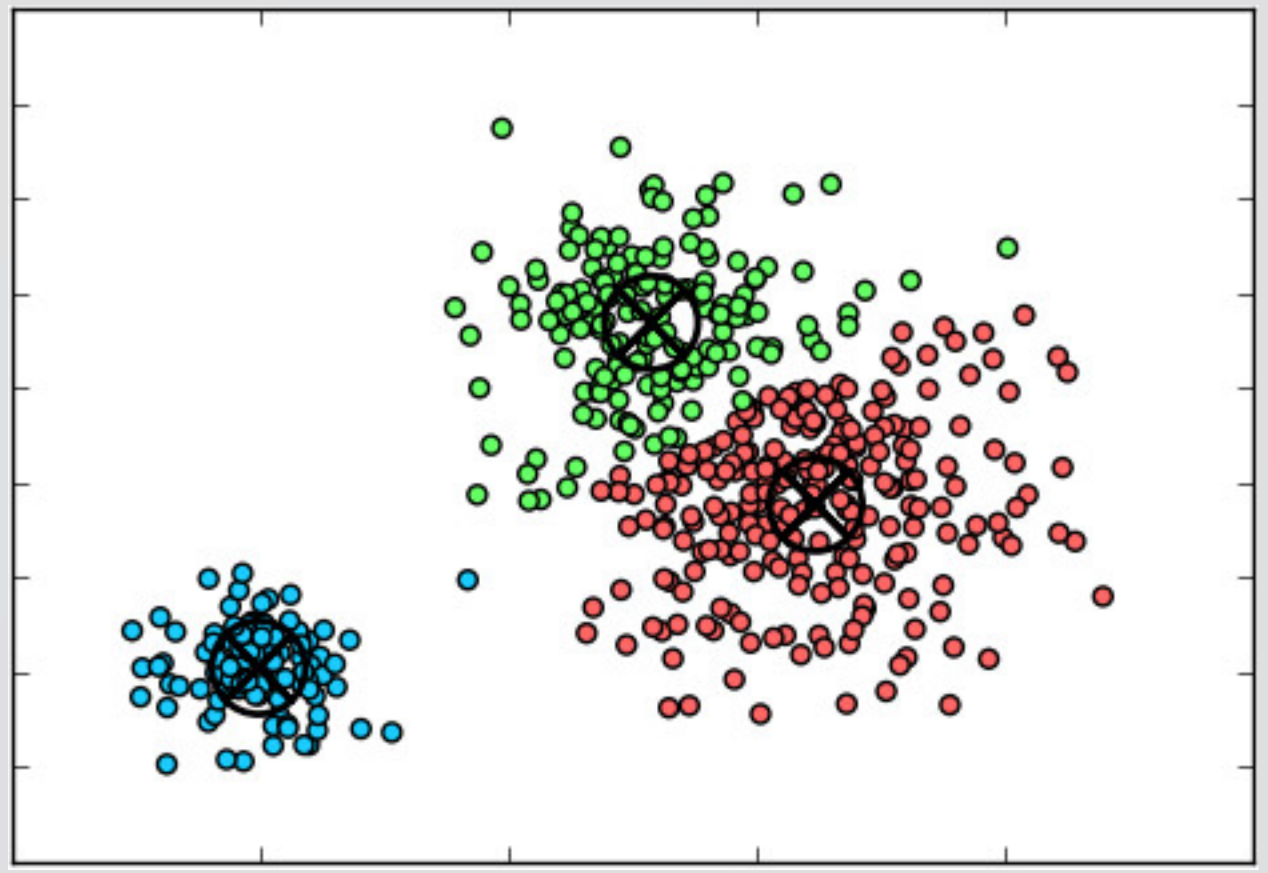
"Software Engineer"
"Sr Software Engineer"

"Software Engineer"
"Computer Programmer"

Document Classification:
What type of news does this article contain?

Predict Unknown Values

Document Clustering:
What structure does this collection of documents show?

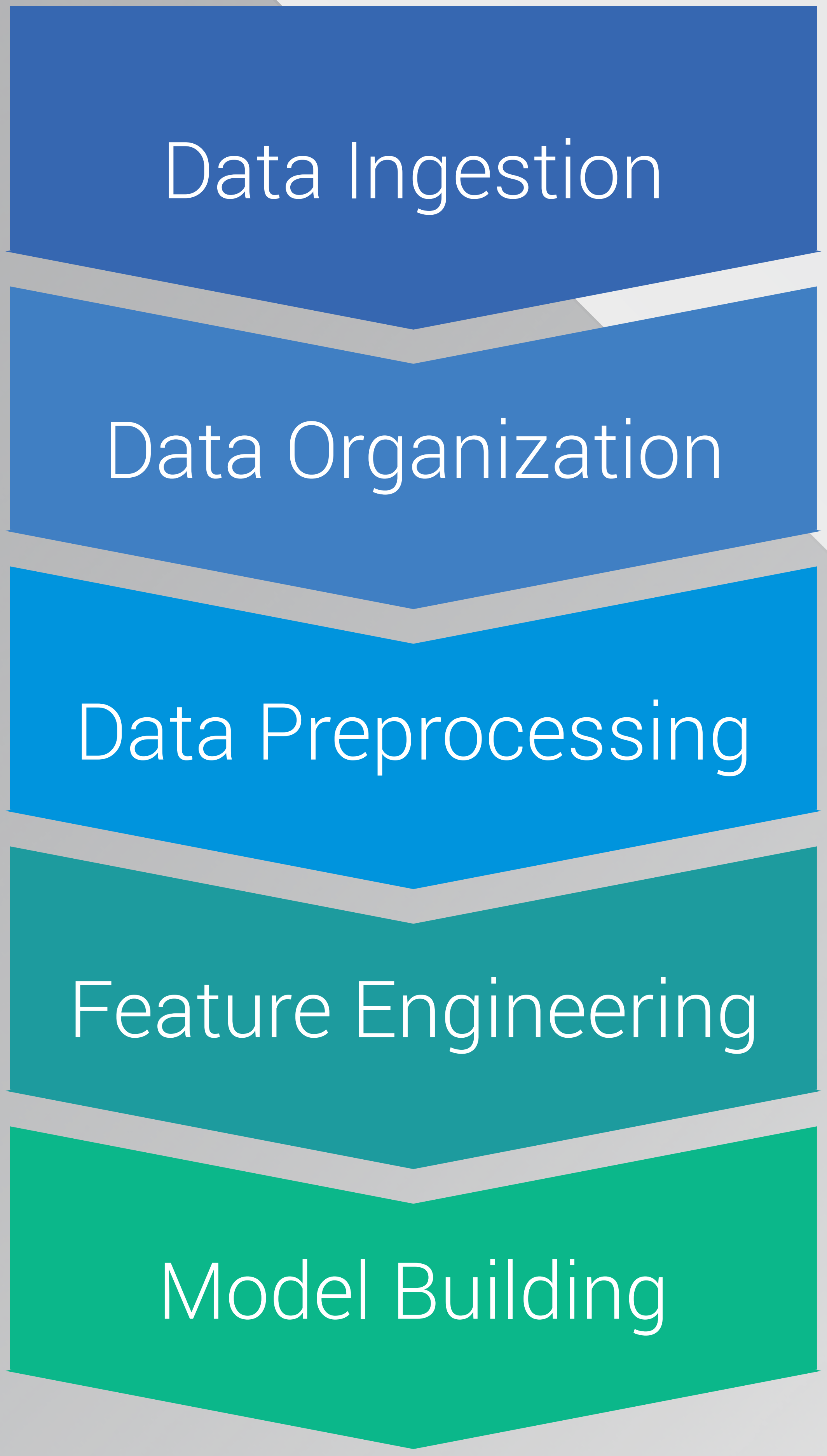


Find Patterns to Describe Data

Approaches to Analyze Text

Text Mining	Natural Language Processing
Goal: Extract Useful Information from Text Data (Overview)	Goal: Understand what the Text is Conveying (Granular)
Approaches: Pattern Matching or Matching Structure of Text Data	Approaches: Extracting Semantic Meaning from Text Data Models used typically include: Lexical Analysis (Part of Speech tagging) Named Entity Recognition, Relationship Extraction and Semantic Analysis (WordNet, DBpedia)

Resources for Text Analysis Projects



Applied Text Analysis with Python
by Benjamin Bengfort, Rebecca Bilbro & Tony Ojeda

Speech and Language Processing
by Dan Jurafsky & James Martin

Foundations of Statistical NLP
by Chris Manning & Hinrich Schutze

Natural Language Processing with Python
by Steven Bird, Ewan Klein & Edward Loper

Text Mining with R
by Julia Silge & David Robinson

tapRecruit.co

Data Preprocessing is Critical

- Data Ingestion
- Data Organization
- Data Preprocessing
- Feature Engineering
- Model Building

Alex Gude
@alex_gude

Here is a real use case from work for model improvement and the steps taken to get there:

- Baseline: 53%
- Logistic: 58%
- Deep learning: 61%
- **Fixing your data: 77%**

Some good ol' fashion "understanding your data" is worth it's weight in hyperparameter tuning!

3:48 PM · Apr 24, 2019 · Twitter Web Client

The Lifecycle of a Text Analysis Project

- Data Ingestion
- Data Organization
- Data Preprocessing
- Feature Engineering
- Model Building

Hackathons are awesome'

('hackathon', NN)

Corpus:
A collection of documents

Document:
Unprocessed string, typically associated with structured data

Segment:
Processed string (i.e. sentence, paragraph etc.)

Token:
Single processed data point

Today:

1. Designing Text Preprocessing Pipelines

2. Transforming Text into Feature Representations

3. Approaches to Classify Documents

tapRecruit.co

Best Practices in Data Organization

Separating Raw Documents from Processed Intermediaries

```
corpus
|_README.md
|_raw
   |_01.txt
   |_02.txt
   |_03.txt
   |_metadata.json
|_processed
   |_processed.json
   |_metadata.json
|_scripts
```

Processed documents:
Save down processed documents either as JSON objects or in a document database (NoSQL)

Metadata:
Define what has been processed and when in metadata:

- Files
- Words
- Unique Tokens
- Date of latest preprocessing

Further Reading: Applied Text Analysis with Python by Bengfort, Bilbro & Ojeda

Advanced Best Practices in Data Organization

Creating a Corpus Reading Module

- Define which files should be loaded and how those files should be loaded.
  - Store these as parameters in README.
  - Regex for file names / formats [**\w\.txt+**]
  - Can include a filter list for restricting files

```
import json

def project_reader(self):
    return json.load(self.open("README"))
```

```
corpus
|_README.md
|_raw
  |_01.txt
  |_02.txt
  |_03.txt
  |_metadata.json
|_processed
  |_processed.json
  |_metadata.json
|_scripts
```

Further Reading: Applied Text Analysis with Python by Bengfort, Bilbro & Ojeda

Designing Data Preprocessors

Clean up

Goal: Remove inconsistency between otherwise similar data points

Segmentation

Goal: Split text chunks into data points (i.e. the unit of analysis or evaluation)

Normalization

Goal: Put data points on an equal footing

Designing Data Preprocessors

Clean up

Segmentation

Normalization

General Considerations

- What is the unit or data structure of analysis? (Tokens vs sentences vs paragraphs vs docs)
- Can the cleanup aid segmentation?

Specific Considerations

- What is the role of punctuation?
- What role do hyphenated words play?
- Will parsing emojis or emoticons be helpful?

Typical Clean Up Pipeline

Step	Example	Tools
HTML/XML	** **We didn\u0027t start the fire!** **	Python: **`Beautiful Soup`** R: **`xml2?`**
Unicode	We didn**\u0027**t start the fire!	Python: **`regex`** R: **`utf8`**
Contractions	We did**n't** start the fire!	Python: **`spaCy`** R: **`textclean`**
Punctuation	We did not start the fire**!**	Python: **`spaCy`** R: **`textclean`**
Tokenization	We | did | not | start | the | fire	Python: **`spaCy`** R: **`tokenizers`**

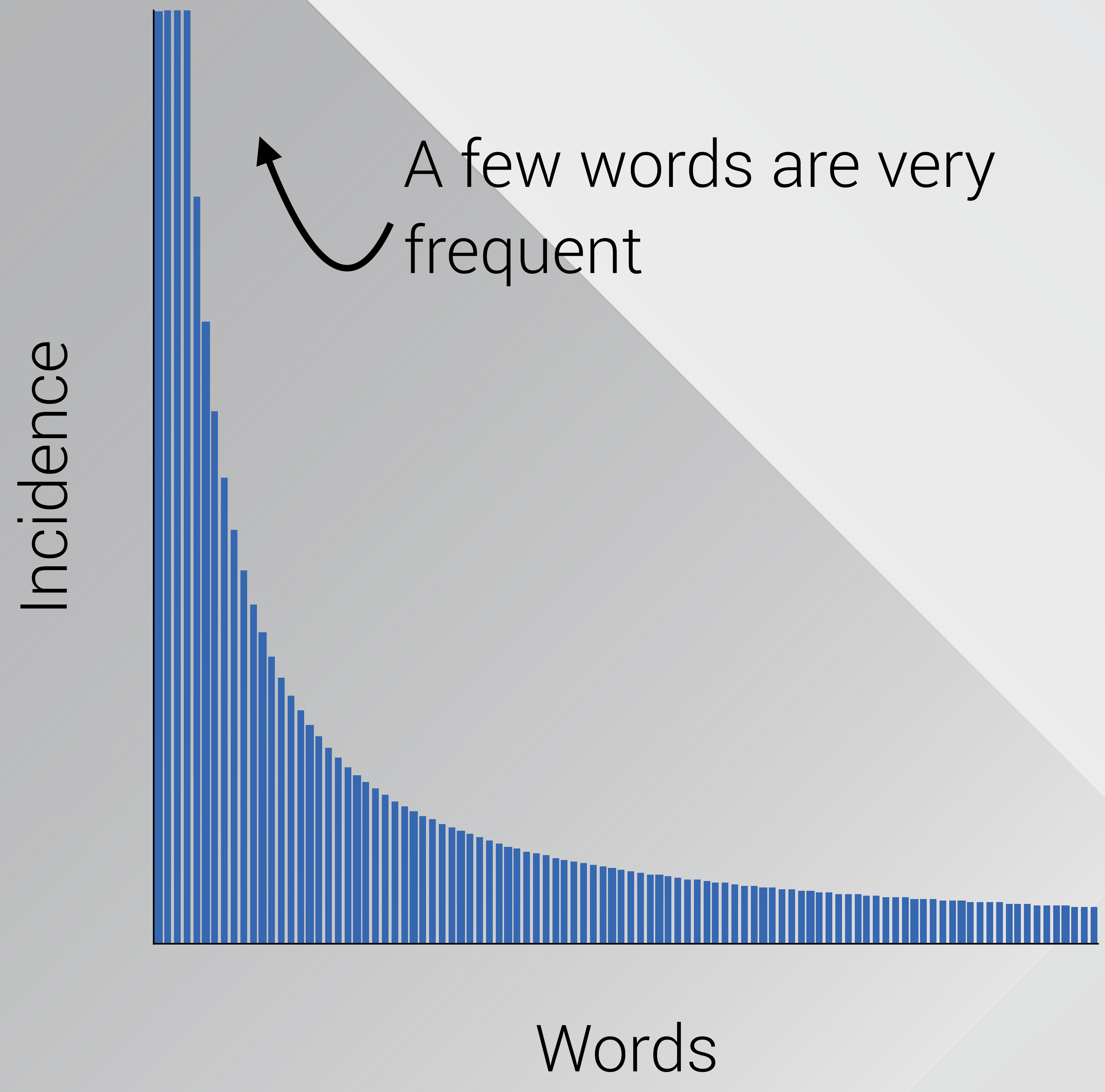
tapRecruit.co

The Order of Operations Matters!

HTML/XML	** **We didn\u0027t start the fire!** **
Contractions	We didn****u0027t start the fire!
Punctuation	We didnu0027t start the fire
Unicode	We didnu0027t start the fire
Tokenization	We | didnu0027t | start | the | fire

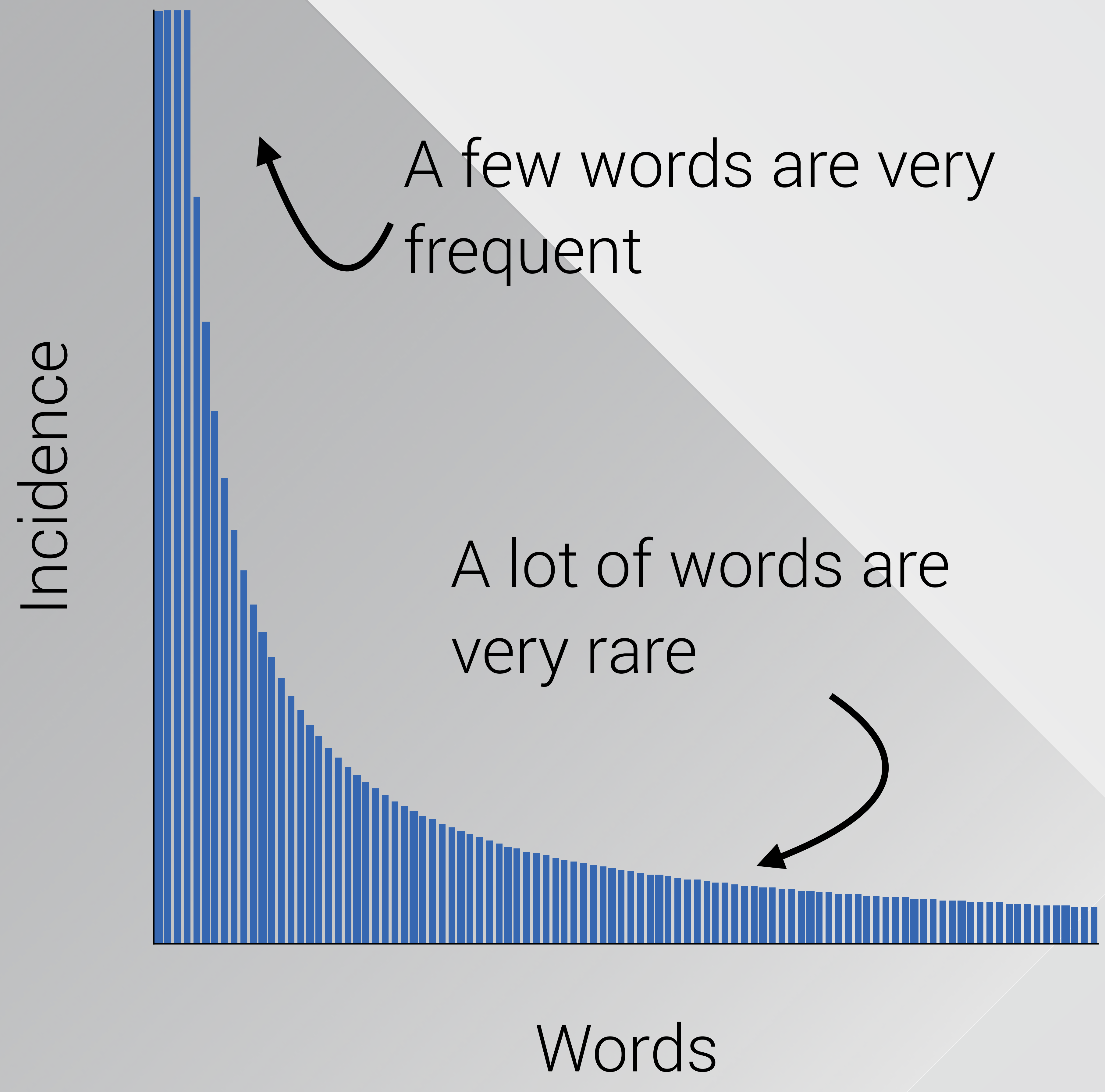
tapRecruit.co

Word Incidence is Rarely Distributed Normally



- **Stop words**: Removing most frequent words.
  - Standard list with most NLP libraries
  - Make your own artisanal list

Word Incidence is Rarely Distributed Normally



- **Stop words**: Removing most frequent words.
  - Standard list with most NLP libraries
  - Make your own artisanal list
- **Changing cases**:
  - Standard is to convert to lower case
  - Casing may matter for you (e.g. IT vs it)
- **Process numbers**:
  - Standard is to remove all numbers
  - Convert into words via **`inflect`** library in Python and **`textclean`** package in R
- **Correct spelling**:
- **Stem or lemmatize words**: Lemmatization is standard

tapRecruit.co

Case Study 1

Add information from a new source to your CRM database, joining on job titles.
The new information comes from a scraped website.

How should we preprocess this data to ensure we correctly resolve duplicate records?

Today:

1. Designing Text Preprocessing Pipelines

2. Transforming Text into Feature Representations

3. Approaches to Classify Documents

Feature Representations

Bag of Words representation vectorizes through word counts

	baby	did	doo	fire	not	shark	start	the	we
"We didn't start the fire"	0	1	0	1	1	0	1	1	1
"Baby shark, doo doo doo doo doo doo"	1	0	6	0	0	1	0	0	0

Feature Representations

Normalized Bag of Words can be instructive

“We didn’t start the fire”

	baby	did	doo	fire	not	shark	start	the	we
Bag of Words	0	1	0	1	1	0	1	1	1
- stop words	0	0	0	1	0	0	1	0	0
+ normalization	0	0	0	0.5	0	0	0.5	0	0

Feature Representation

One Hot Encoding and TFIDF normalize token frequencies

"Baby shark, doo doo doo doo doo doo"

	baby	did	doo	fire	not	shark	start	the	we
Bag of Words	1	0	6	0	0	1	0	0	0
One Hot Encoding	1	0	1	0	0	1	0	0	0
TFIDF	0.05	0	0.4	0	0	0.2	0	0	0

Overview of Feature Representations Methods

One Hot Encoding	Bag of Words	TF-IDF
Stop word removal?	Stop word removal	No need for stop word removal
None	Document-level normalization	Corpus and document-level normalization

Word Embeddings capture Semantic Similarities

Statistical modeling through software (e.g. SPSS) or programming language (e.g. **Python**)

Context **Word**

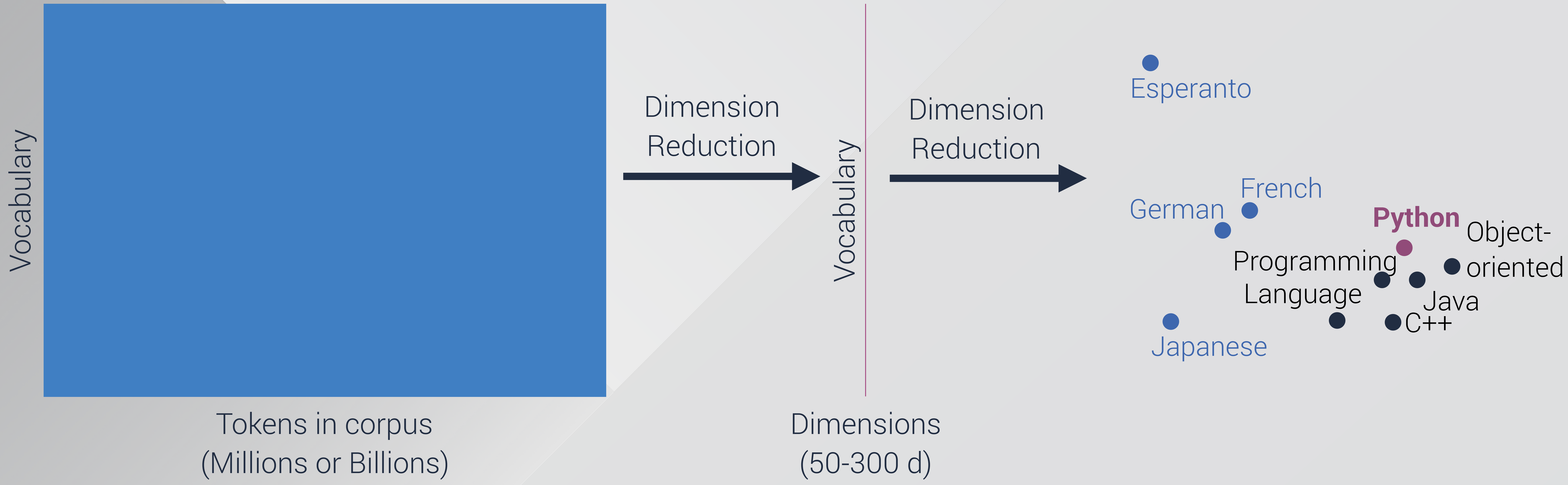
Experience in **Python**, Java or other object-oriented programming languages

Context **Word** **Context**

Proficiency programming in **Python**, Java or C++.

Context **Word** **Context**

A Simplified Representation of Word Vectors



tapRecruit.co

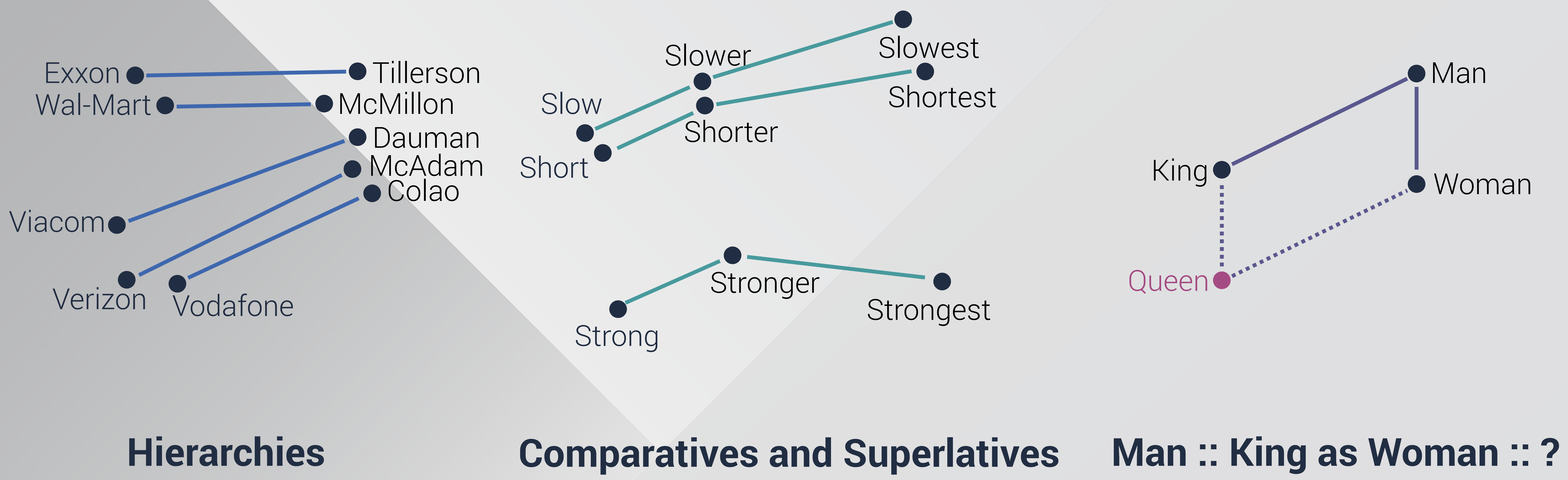
A Simplified Representation of Word Vectors

	Proficiency	programming	Python	Java	C++	...	Experience	Python	Java	object-oriented	programming	languages
proficiency	1	0	0	0	0		0	0	0	0	0	0
programming	0	1	0	0	0		0	0	0	0	1	0
python	0	0	1	0	0		0	1	0	0	0	0
java	0	0	0	1	0		0	0	1	0	0	0
c++	0	0	0	0	1		0	0	0	0	0	0
experience	0	0	0	0	0		1	0	0	0	0	0
object-oriented	0	0	0	0	0		0	0	0	1	0	0
languages	0	0	0	0	0		0	0	0	0	0	1

Dimension Reduction

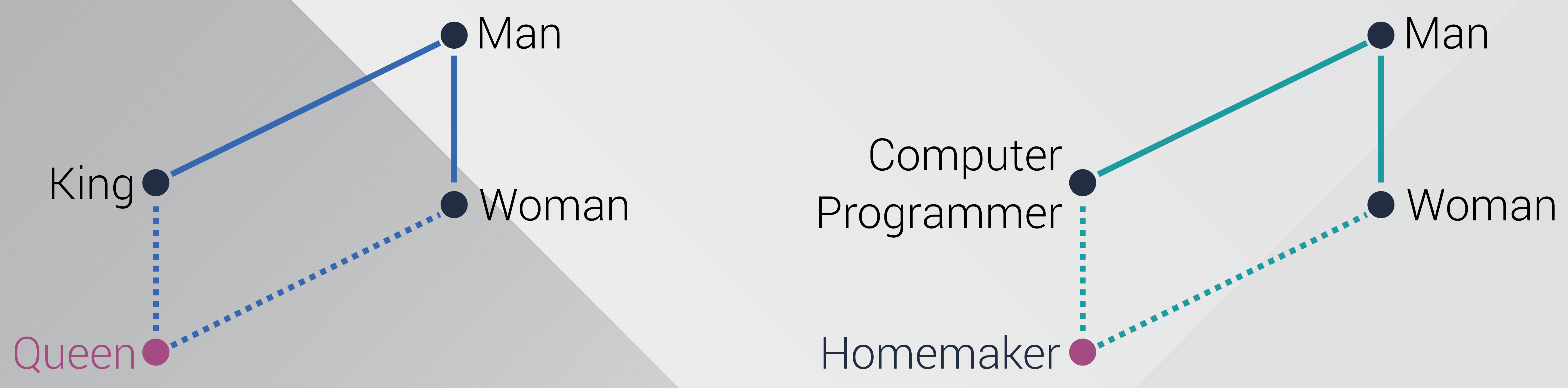
	qualification	**data science**	**translation**	...
proficiency	**0.8**	0.1	0.3	...
programming	0.2	**0.7**	0.1	...
python	0.1	**0.9**	-0.2	...
java	-0.1	**0.6**	0.1	...
c++	-0.2	0.4	-0.1	...
experience	**1.0**	0.3	0.4	...
object-oriented	-0.3	0.3	-0.2	...
languages	0.4	0.1	**0.9**	...

Word Embeddings capture Entity Relationships



Adapted from Stanford NLP GLoVE Project

Embeddings reflect Cultural Bias in Corpora



Man :: King as Woman :: ? **Man :: Programmer as Woman :: ?**

Adapted from Bolukbasi et al., arXiv: 1607:06520.

Overview of Feature Representations Methods

Non-distributed			Distributed
One Hot Encoding	Bag of Words	TF-IDF	Embeddings
Stop word removal?	Stop word removal	No need for stop word removal	Stop word removal?
None	Document-level normalization	Corpus and document-level normalization	Context and corpus-level normalization

tapRecruit.co

Overview of Feature Representations Methods

Non-distributed			Distributed
One Hot Encoding	Bag of Words	TF-IDF	Embeddings
Stop word removal?	Stop word removal	No need for stop word removal	Stop word removal?
None	Document-level normalization	Corpus and document-level normalization	Context and corpus-level normalization
All tokens are equidistant			Distance $\propto$ token similarity
High dimensionality, extremely sparse			Lower dimensionality

tapRecruit.co

Case Study 2

Our marketing department has been running ad campaigns on different social media platforms.
They want to understand which characteristics are shared by the ads that are high performers.

What would we advise them?

Today:

1. Designing Text Preprocessing Pipelines
2. Transforming Text into Feature Representations
3. **Approaches to Classify Documents**

tapRecruit.co

Tasks in Document Analysis

Descriptive Approaches	Predictive Approaches
Find patterns to describe data	Predict unknown values from a model using known baselines
Approach: Unsupervised Learning Class labels of data is unknown. Given a set of measurements, goal is to identify some clusters in the data.	Approach: Supervised Learning Generate a model using training data which has a set of labels which indicate class of observations. Model classifies new data.
Examples: Clustering, Summarization	Examples: Classification, Ranking, Regression etc.

Overview of Document Categorization

Extracting semantic structure from numeric features

Topic Modeling	Document Classification
What are the topics that occur in a collection of documents?	Which class does document X belong to?
Every document is a mixture of topics	Every document belongs to a single class
Every topic is a mixture of words	The presence or absence of a subset of words impacts the classification
Unsupervised Dimension Reduction (LDA / LSA)	Supervised ML Algorithms Regex (Standard or Artisanal)

Case Study 3

Our colleagues have conducted a survey, including a lot of questions that had free-form answers. They need help extracting some insights from these answers.

How would you advise them to analyze the data?

Takeaways:

1. Text Preprocessing

Goal: To put data points on an equal(ish) footing

Cleanup / Segmentation / Normalization

Stopwords / Cases / Spelling / Lemmatization

2. Feature Representation of Text

Goal: To transform text into vectorized representations for downstream

Bag of Words / TFIDF / One-Hot Encoding / Embeddings

3. Document Classification

Goal: To determine whether a document is similar or different to others

Topic Modeling / Document Classification

Thank you ERP Hackathon!

Link to deck: http://bit.ly/erp-hackathon-2019

Maryam Jahanshahi Ph.D.
Research Scientist

@mjahanshahi
maryam-j

tapRecruit.co