Lecture 8
Text Analytics

Dr. Lydia Wahid

Agenda

- Introduction
- Text Analytics Steps
- Text Analytics Example
- Text Analytics Example Process

Introduction

Introduction

- *Text analysis*, sometimes called *text analytics*, refers to the **representation**, **processing**, and **modeling** of textual data **to derive useful insights**.
- An important component of text analysis is *text mining*, **the process of discovering relationships and interesting patterns in large text collections**.
- The **high dimensionality** of text is an important issue, and it has a direct impact on the complexities of many text analysis tasks.
- Another major challenge with text analysis is that most of the time the **text is not structured**.

Introduction

➢The following table shows some example data sources and data formats that text analytics may have to deal with:

Data Source	Data Format	Data Structure Type
News articles	TXT, HTML, or Scanned PDF	Unstructured
Literature	TXT, DOC, HTML, or PDF	Unstructured
E-mail	TXT, MSG, or EML	Unstructured
Web pages	HTML	Semi-structured
Server logs	LOG or TXT	Semi-structured
Social network API firehoses	XML, JSON, or RSS	Semi-structured
Call center transcripts	TXT	Unstructured

Text Analytics Steps

Text Analytics Steps

- A text analytics problem usually consists of three important steps:
  - Parsing
  - Search and Retrieval
  - Text Mining

- A text analytics problem may also consist of other subtasks (such as discourse and segmentation)

Text Analytics Steps: Parsing

- ***Parsing*** is the process that takes unstructured text and imposes a structure for further analysis.

- The unstructured text could be a plain text file, a weblog, an Extensible Markup Language (XML) file, a HyperText Markup Language (HTML) file, or a Word document.

- Parsing deconstructs the provided text and renders it in a more structured way for the subsequent steps.

Text Analytics Steps: Search and retrieval

- ***Search and retrieval*** is the identification of the documents in a corpus that contain search items such as specific words, phrases, topics, or entities like people or organizations.

- These search items are generally called ***key terms.***

- Search and retrieval originated from the field of library science and is now used extensively by web search engines.

Text Analytics Steps: Text mining

- ***Text mining*** uses the terms and indexes produced by the prior two steps to discover meaningful insights pertaining to domains or problems of interest.
- With the proper representation of the text, many of the techniques mentioned in the previously, such as clustering and classification, can be adapted to text mining.
- For example, the *k*-means can be modified to cluster text documents into groups, where each group represents a collection of documents with a similar topic. The distance of a document to a centroid represents how closely the document talks about that topic.

Text Analytics Steps: Text mining

- Classification tasks such as sentiment analysis and spam filtering are prominent use cases for the naïve Bayes classifier.

- Text mining may utilize methods and techniques from various fields of study, such as statistical analysis, information retrieval, data mining, and natural language processing.

Text Analytics Steps

- In reality, all three steps do not have to be present in a text analytics project.

- If the goal is to construct a corpus or provide a catalog service, for example, the focus would be the parsing task using one or more text processing techniques, such as part-of-speech (POS) tagging, named entity recognition (NER), lemmatization, or stemming.

- Furthermore, the three tasks do not have to be sequential.

Text Analytics Steps

- **Part-of-Speech (POS) Tagging:**
  - The goal of POS tagging is to build a model whose **input** is a sentence, such as:
    - He saw a fox
  - and whose **output** is a tag sequence. Each tag marks the POS for the corresponding word, such as:
    - PRP VBD DT NN
  - Therefore, the four words are mapped to pronoun (personal), verb (past tense), determiner, and noun (singular), respectively.

Text Analytics Steps

➢ **Named Entity Recognition:**

- It is a subtask of information extraction that seeks to locate and classify named entities in unstructured text into pre-defined categories such as **person names**, **organizations**, **locations**, **time expressions**, monetary values.
- The input is an unannotated block of text, such as:
  - Jim bought 300 shares of Acme Corp. in 2006.
- And the output is an annotated block of text that highlights the names of entities:
  - [Jim]Person bought 300 shares of [Acme Corp.]Organization in [2006]Time.

Text Analytics Steps

➢ Lemmatization:

- It is the algorithmic process of determining the **lemma** (i.e. base) of a word based on its intended meaning.
- It reduces inflections or variant forms to the base form.
- For example, in English, the verb 'to walk' may appear as 'walk', 'walked', 'walks' or 'walking'. The base form, 'walk', in a dictionary, is called the *lemma* for the word.
- The association of the base form with a part of speech is often called a **lexeme** of the word.
- **For example:** Fire <u>causes</u> <u>problems</u> → Fire <u>cause</u> <u>problem</u>

Text Analytics Steps

- **Stemming:**
  - Different from lemmatization, ***stemming*** does not need a dictionary, and it usually refers to a basic process of removing affixes based on a set of heuristics with the hope of correctly achieving the goal to reduce inflections or variant forms.
  - After the process, words are become **stems**.
  - A stem is not necessarily an actual word defined in the natural language, but it is sufficient to differentiate itself from the stems of other words.
  - A well-known rule-based stemming algorithm is ***Porter's stemming algorithm***. It defines a set of production rules to iteratively transform words into their stems.
  - For the previous sentence: Fire <u>causes</u> <u>problems</u> → Fire <u>caus</u> <u>problem</u>

Text Analytics Example

Text Analytics Example

- To further describe the three text analysis steps, consider the fictitious company ACME, maker of two products: *bPhone* and *bEbook*.

- ACME is in strong competition with other companies that manufacture and sell similar products.

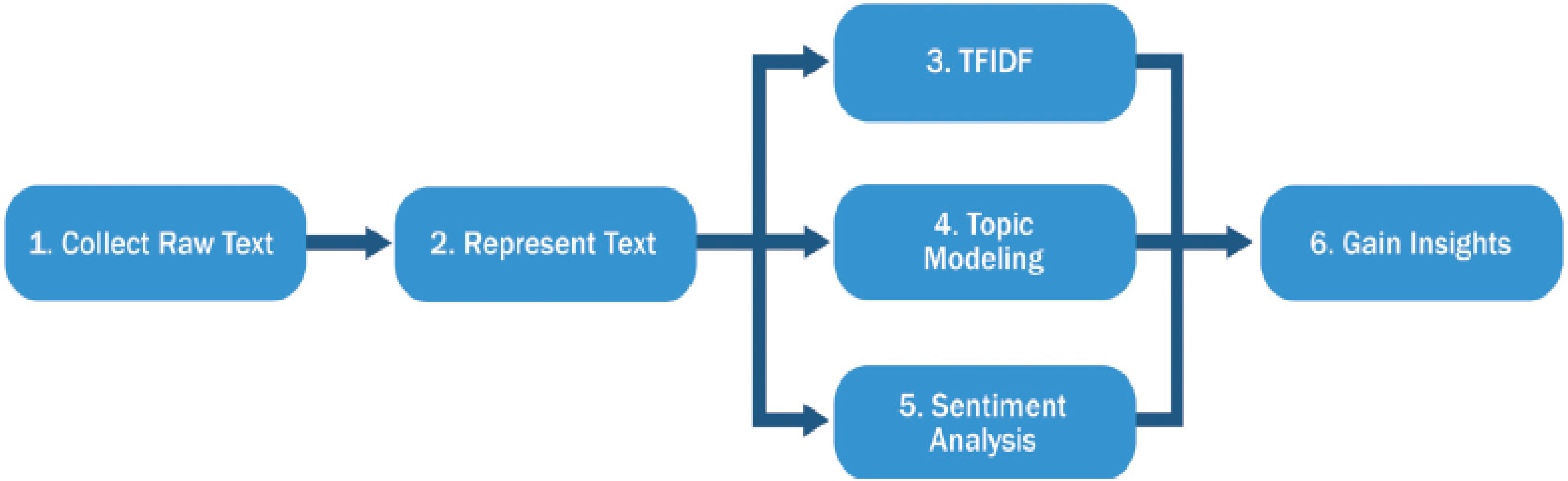
- To succeed, ACME needs to produce excellent phones and eBook readers and increase sales.

Text Analytics Example

- One of the ways the company does this is to monitor what is being said about ACME products in social media.
- It wants to answer questions such as these.
  - Are people mentioning its products?
  - What is being said? Are the products seen as good or bad?
  - If people think an ACME product is bad, why? For example, are they complaining about the battery life of the *bPhone*, or the response time in their *bEbook*?

Text Analytics Example

- ACME can monitor the social media buzz using a process based on the three steps outlined. This process includes the following modules:



Text Analytics Example

1. **Collect raw text.** In this step, the Data Science team at ACME monitors websites for references to specific products. The websites may include social media and review sites
2. **Represent text.** Convert each review into a suitable document representation with proper indices, and build a corpus based on these indexed reviews.
3. **TFIDF.** Compute the usefulness of each word in the reviews using methods such as TFIDF.
4. **Topic Modeling.** Categorize documents by topics. This can be achieved through topic models (such as latent Dirichlet allocation).

Text Analytics Example

5. **Sentiment Analysis.** Determine sentiments of the reviews. Identify whether the reviews are positive or negative.

- Many product review sites provide ratings of a product with each review. If such information is not available, techniques like sentiment analysis can be used on the textual data to infer the underlying sentiments.
- People can express many emotions. ACME considers three sentiments: positive, neutral, or negative

Text Analytics Example

6. **Gain Insights.** Review the results and gain greater insights.

- Results are gathered from the previous steps.
- Find out what exactly makes people love or hate a product.
- Use one or more visualization techniques to report the findings.
- Test the soundness of the conclusions and put the findings into operation if applicable.

Text Analytics Example Process

Text Analytics Example Process: 1. Collect raw text

- For text analytics, data must be **collected** before anything can happen.
- The Data Science team starts by actively **monitoring** various websites for user-generated contents.
- The user-generated contents being collected could be related **articles** from news portals and blogs, **comments** on ACME's products from online shops or reviews sites, or social media **posts** that contain keywords *bPhone* or *bEbook*.

Text Analytics Example Process: 1. Collect raw text

- Many websites and services offer **public APIs** for third-party developers to access their data.

- If APIs are not provided, the team may have to write **web scrapers** to parse web pages and automatically extract the interesting data from those HTML files.

- A ***web scraper*** is a software program (bot) that systematically browses the World Wide Web, downloads web pages, extracts useful information, and stores it somewhere for further study.

Text Analytics Example Process: 1. Collect raw text

- Unfortunately, it is nearly impossible to write a fits-all web scraper.
- This is because websites like online shops and review sites have different structures. It is common to customize a web scraper for a specific website.
- To build a web scraper for a specific website, one must study the HTML source code of its web pages to find patterns before extracting any useful content.

Text Analytics Example Process: 1. Collect raw text

- For example, the team may find out that each user comment in the HTML is enclosed by a DIV element inside another DIV with the ID *usrcommt*.

- The scraper can use the **curl** tool to fetch HTML source code given specific URLs, use **XPath and regular expressions** to select and extract the data that match the patterns, and write them into a data store.

- Regular expressions can find words and strings that match particular patterns in the text effectively and efficiently.

Text Analytics Example Process: 1. Collect raw text

- The following table show some regular expressions:

Regular Expression	Matches	Note
`b(P|p)hone`	bPhone, bphone	Pipe "|" means "or"
`bEbo*k`	bEbk, bEbok, bEbook, bEboook, bEbooook, bEboooook, …	"*" matches zero or more occurrences of the preceding letter
`bEbo+k`	bEbok, bEbook, bEboook, bEbooook, bEboooook, …	"+" matches one or more occurrences of the preceding letter
`bEbo{2,4}k`	bEbook, bEboook, bEbooook	"{2,4}" matches from two to four repetitions of the preceding letter "o"
`^I love`	Text starting with "I love"	"^" matches the start of a string
`ACME$`	Text ending with "ACME"	"$" matches the end of a string

Text Analytics Example Process: 1. Collect raw text

- **IMPORTANT NOTE:**
  - Depending on how the fetched raw data will be used, the Data Science team needs to be careful **not to violate the rights of the owner** of the information and user agreements about use of websites during the data collection.
  - Many websites place a file called **robots.txt** in the root directory—that is, http://.../robots.txt (for example, http://www.amazon.com/robots.txt).
  - It lists the directories and files that are allowed or disallowed to be visited so that web scrapers or web crawlers know how to treat the website correctly.

Text Analytics Example Process: 2. Representing Text

- After the previous step, the team now has some raw text to start with.
- In this data representation step, raw text is first transformed with text normalization techniques such as *tokenization* and *case folding*.
- Then it is represented in a **more structured way** for analysis.
- ***Tokenization*** is the task of separating (also called tokenizing) words from the body of text. Raw text is converted into collections of tokens after the tokenization, where each token is generally a word.
- Another text normalization technique is called ***case folding***, which reduces all letters to lowercase (or the opposite).

Text Analytics Example Process: 2. Representing Text

- **Tokenization** is a much more difficult task than one may expect.
  - For example, should words like state-of-the-art, Wi-Fi, and San Francisco be considered one token or more? Should words like résumé and resume all map to the same token?
  - There is no single tokenizer that will work in every scenario. The team needs to decide what counts as a token depending on the domain of the task and select an appropriate tokenization technique that fits most situations well.
  - It's common to pair a standard tokenization technique with a **lookup table** to address the contractions and terms that shouldn't be tokenized.

Text Analytics Example Process: 2. Representing Text

- One needs to be cautious applying **case folding** to tasks such as information extraction, sentiment analysis, and machine translation.
  - If implemented incorrectly, case folding may reduce or *change the meaning* of the text and *create additional noise*.
  - For example, when the abbreviation of the World Health Organization WHO become who, it may be interpreted as the pronoun who.
  - One way to reduce such problems is to create a **lookup table** of words not to be case folded. Alternatively, the team can come up with some **heuristics** or **rules-based strategies** for the case folding. For example, the program can be taught to ignore words that have uppercase in the middle of a sentence.

Text Analytics Example Process: 2. Representing Text

- After normalizing the text by tokenization and case folding, it needs to be represented in a more structured way. A simple yet widely used approach to represent text is called ***bag-of-words.***
  - Given a document, bag-of-words represents the **document as a set of terms**, ignoring information such as order and context.
  - **Each word is considered a term** or token. In many cases, bag-of-words additionally assumes every term in the document is independent
  - The document then becomes **a vector with one dimension** for every distinct term in the space, and the terms are unordered.

Text Analytics Example Process: 2. Representing Text

- Besides extracting the terms, their morphological ***features*** may need to be included.
- The morphological features specify additional information about the terms, which may include root words, part-of-speech tags, named entities,..etc.
- The features from this step contribute to the analysis in classification or sentiment analysis.
- The set of features that need to be extracted and stored highly depends on the specific task to be performed.

Text Analytics Example Process: 3. TFIDF

- **Term Frequency—Inverse Document Frequency (**TFIDF) is a measure widely used in information retrieval and text analytics.

- Using single words as identifiers with the bag-of-words representation, the ***term frequency*** (TF) of each word can be calculated.

- Term frequency represents the weight of each term in a document, and it is proportional to the number of occurrences of the term in that document.

Text Analytics Example Process: 3. TFIDF

- Given a term t and a document $d = \{t1, t2,..tn\}$ containing n terms, the simplest form of term frequency of t in d can be defined as the number of times t appears in d:

$$TF_1(t,d) = \sum_{i=1}^{n} f(t,t_i) \qquad t_i \in d; |d| = n$$

$$f(t,t') = \begin{cases} 1, & \text{if } t = t' \\ 0, & \text{otherwise} \end{cases}$$

Text Analytics Example Process: 3. TFIDF

- Because longer documents contain more terms, they tend to have higher term frequency values. They also tend to contain more distinct terms.
- These factors can raise the term frequency values of longer documents and lead to undesirable bias favoring longer documents.
- To address this problem, the term frequency can be normalized:

$$TF_2(t,d) = \frac{TF_1(t,d)}{n} \qquad |d| = n$$

Text Analytics Example Process: 3. TFIDF

- A term frequency vector can become very high dimensional.

- It is useful to store a term and its frequency only if the term appears at least once in a document.

- To reduce the dimensionality further, we can remove ***stop words*** such as: *the*, *a*, *of*, *and*, *to*.

- Some NLP techniques such as **lemmatization and stemming** can also reduce high dimensionality.

Text Analytics Example Process: 3. TFIDF

- Term frequency by itself suffers a critical problem: It regards that stand-alone document as the entire world.
- The importance of a term is solely based on its presence in this particular document.
- A fix for the problem is to introduce an additional variable that has a broader view of the world—considering the importance of a term not only in a single document but in a collection of documents (or corpus).
- That is the intention of the ***inverted document frequency*** (IDF). The IDF inversely corresponds to the ***document frequency*** (DF).

Text Analytics Example Process: 3. TFIDF

- ***Document frequency*** (DF) is defined to be the number of documents in the corpus that contain a term.
- Let a corpus *D* contain *N* documents. The document frequency of a term *t* in corpus $D=\{d1, d2, \ldots dN\}$ is defined as:

$$DF(t)=\sum_{i=1}^{N} f'(t,d_i) \qquad d_i \in D; |D| = N$$

$$f'(t,d')=\begin{cases} 1, & \text{if } t \in d' \\ 0, & \text{otherwise} \end{cases}$$

Text Analytics Example Process: 3. TFIDF

- The **Inverse document frequency** of a term *t* is obtained by dividing *N* by the document frequency of the term and then taking the logarithm of that quotient:

$$IDF_1(t) = \log \frac{N}{DF(t)}$$

- If the term is not in the corpus, it leads to a division-by-zero. A quick fix is to add 1 to the denominator:

$$IDF_2(t) = \log \frac{N}{DF(t)+1}$$

Text Analytics Example Process: 3. TFIDF

- Words with higher IDF tend to be more meaningful over the entire corpus.
- In other words, the IDF of a rare term would be high, and the IDF of a frequent term would be low.
- For example, if a corpus contains 1,000 documents, 1,000 of them might contain the word *the*, and 10 of them might contain the word *bPhone*.

Text Analytics Example Process: 3. TFIDF

- The **TFIDF (or TF-IDF)** is a measure that considers both the prevalence of a term within a document (TF) and the scarcity of the term over the entire corpus (IDF).
- The TFIDF of a term *t* in a document *d* is defined as the term frequency of *t* in *d* multiplying the document frequency of *t* in the corpus:

$$TFIDF(t,d) = TF(t,d) \times IDF(t)$$

TFIDF scores words higher that appear **more often in a document** but **occur less often across all documents** in the corpus.

Text Analytics Example Process: 4. Topic Modeling

- Topic modeling provides a way to quickly analyze large volumes of raw text and identify the latent topics.
- Probabilistic topic modeling is a suite of algorithms that aim to parse large archives of documents and discover and annotate the topics.
- With the reviews collected and represented, the data science team at ACME wants to categorize the reviews by topics.

Text Analytics Example Process: 4. Topic Modeling

- A **topic** consists of **a cluster of words** that frequently occur together and **share the same theme**.
- The topics of a document are not as straightforward. Consider these two reviews:
  1. The bPhone5x has coverage everywhere. It's much less flaky than my old bPhone4G.
  2. While I love ACME's bPhone series, I've been quite disappointed by the bEbook. The text is illegible, and it makes even my old NBook look blazingly fast.
- Is the first review about bPhone5x or bPhone4G? Is the second review about bPhone, bEbook, or NBook?

Text Analytics Example Process: 4. Topic Modeling

- Intuitively, if a review is talking about bPhone5x, the term *bPhone5x* and related terms (such as *phone* and *ACME*) are likely to appear frequently.

- A document typically consists of **multiple themes** running through the text in different proportions.

- **Document grouping** can be achieved with **clustering** methods such as *k*-means clustering or **classification** methods such as *k*-nearest neighbors or naïve Bayes.

Text Analytics Example Process: 4. Topic Modeling

- Topic modeling provides tools to automatically organize, search, understand, and summarize from vast amounts of information.

- ***Topic models*** are statistical models that examine words from a set of documents, determine the themes over the text, and discover how the themes are associated or change over time.
- The process of topic modeling can be simplified to the following:
  - **1.** Uncover the hidden topical patterns within a corpus.
  - **2.** Annotate documents according to these topics.
  - **3.** Use annotations to organize, search, and summarize texts.

Text Analytics Example Process: 4. Topic Modeling

- ***Latent Dirichlet allocation*** (LDA) is a generative probabilistic modeling technique.

- LDA can be viewed as a case of hierarchical Bayesian estimation with a posterior distribution to group data such as documents with similar topics.

- Many programming tools provide software packages that can perform LDA over datasets. R comes with an **lda** package that has built-in functions and sample datasets.

Text Analytics Example Process: 5. Sentiment Analysis

- In addition to the TFIDF and topic models, the Data Science team may want to identify the sentiments in user comments and reviews of the ACME products.

- ***Sentiment analysis*** refers to a group of tasks that use statistics and natural language processing to mine opinions to identify and extract subjective information from texts.
- **Classification** methods such as naïve Bayes, maximum entropy, and support vector machines (SVM) are often used to extract corpus statistics for sentiment analysis.

Text Analytics Example Process: 5. Sentiment Analysis

- Depending on the classifier, the data may need to be split into training and testing sets. For example, an 80/20 split would produce 80% of the data as the training set and 20% as the testing set.
- Next, one or more classifiers are trained over the training set to learn the characteristics or patterns residing in the data.
- After the training, classifiers are tested over the testing set to infer the sentiment tags.
- Finally, the result is compared against the original sentiment tags to evaluate the overall performance of the classifier.

Text Analytics Example Process: 6. Gain Insights

- So far we has discussed several text analysis tasks including text collection, text representation, TFIDF, topic models, and sentiment analysis.
- This section shows how ACME uses these techniques to gain insights into customer opinions about its products.
- We will consider only *bPhone* to illustrate the steps.
- Corresponding to the data collection phase, the Data Science team has used *bPhone* as the keyword to collect more than 300 reviews from a popular technical review website.

Text Analytics Example Process: 6. Gain Insights

- The 300 reviews are visualized as a word cloud after removing stop words. A ***word cloud*** (or ***tag cloud***) is a visual representation of textual data.
- Tags are generally single words, and the importance of each word is shown with font size or color.
- The reviews have been previously case folded and tokenized into lowercased words, and stop words have been removed from the text.
- TFIDF can be used to highlight the informative words in the reviews. Each word with a larger font size corresponds to a higher TFIDF value. Each review is considered a document.

Text Analytics Example Process: 6. Gain Insights

➢The following figure shows the word cloud built from the 300 reviews:

Overall, the graph reveals
little information.
The team needs to conduct further analyses on the data.

Text Analytics Example Process: 6. Gain Insights

- The popular technical review website allows users to provide **ratings** on a scale from one to five when they post reviews.
- The team can **divide the reviews** into subgroups using those ratings.
- To reveal more information, the team can remove words such as *phone*, *bPhone*, and *ACME*, which are not very useful for the study. Related research often refers to these words as ***domain-specific stop words***.

Text Analytics Example Process: 6. Gain Insights

- The following figure shows the word cloud corresponding to 50 five-star reviews extracted from the data:

The result suggests that customers are satisfied with the *seller*, the *brand*, and the *product*, and they *recommend* bPhone to their friends and families

Text Analytics Example Process: 6. Gain Insights

➢The following figure shows the word cloud of 70 one-star reviews:

The words `sim` and `button` occur frequently enough that it would be advisable to sample the reviews that contain these terms and determine what is being said about buttons and SIM cards.

Text Analytics Example Process: 6. Gain Insights

- Topic models such as LDA can categorize the reviews into topics.
- Each topic focuses on a different aspect that can characterize the reviews.
- For example, a topic from one-star reviews contains words such as *button*, *power*, and *broken*, which may indicate that bPhone has problems related to button and power supply.
- The Data Science team can track down these reviews and find out if that's really the case.

Thank You