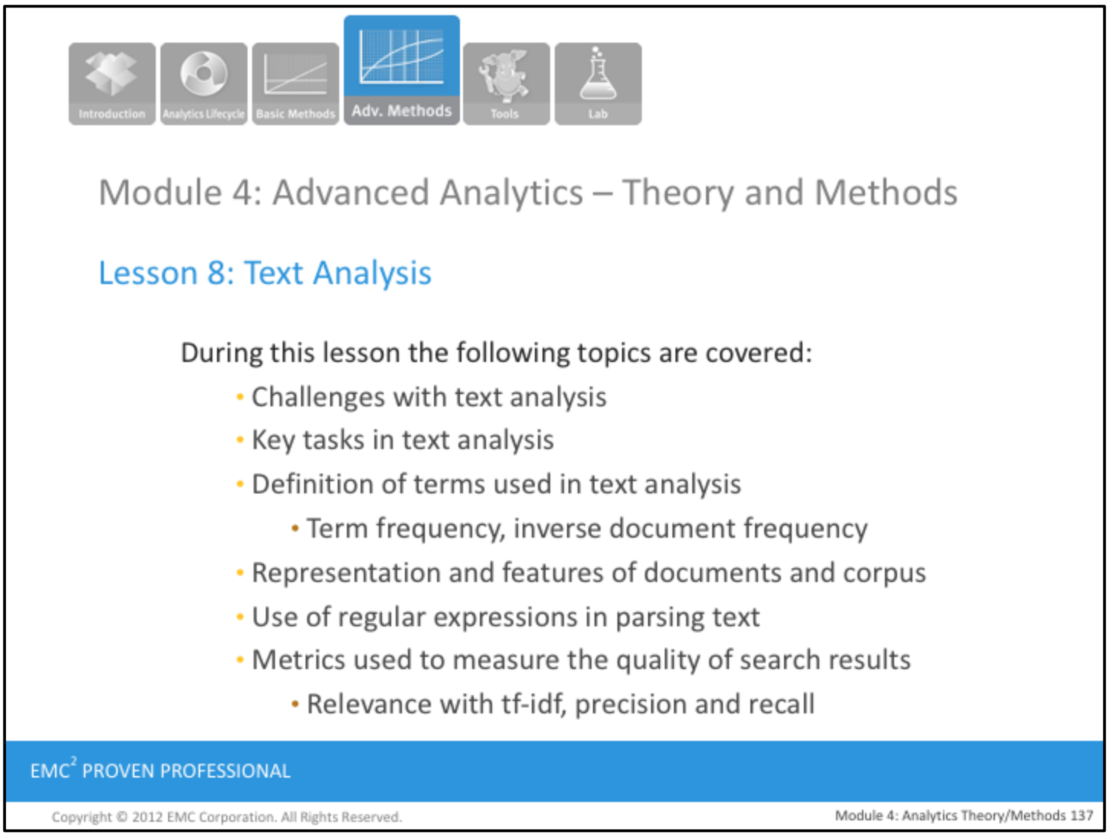

Introduction
Analytics Lifecycle
Basic Methods
Adv. Methods
Tools
Lab

Module 4: Advanced Analytics – Theory and Methods

Lesson 8: Text Analysis

During this lesson the following topics are covered:

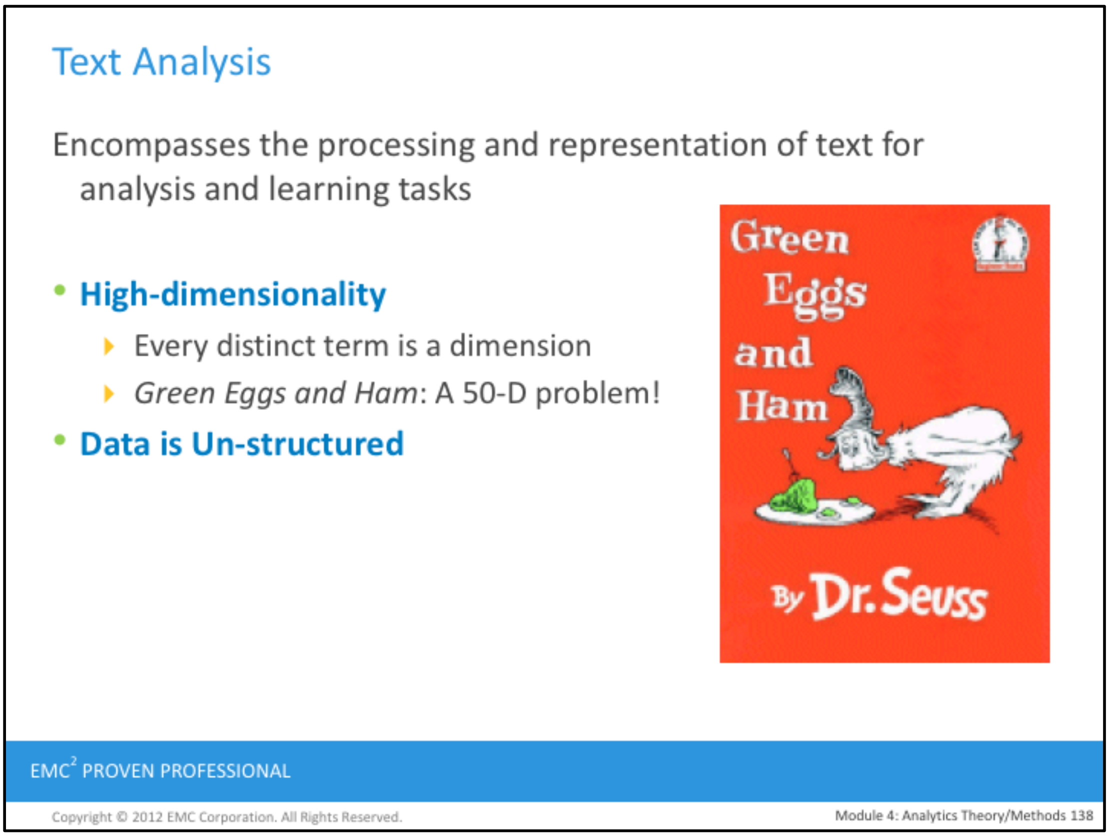
- Challenges with text analysis
- Key tasks in text analysis
- Definition of terms used in text analysis
  - Term frequency, inverse document frequency
- Representation and features of documents and corpus
- Use of regular expressions in parsing text
- Metrics used to measure the quality of search results
  - Relevance with tf-idf, precision and recall

EMC^2 PROVEN PROFESSIONAL

Copyright © 2012 EMC Corporation. All Rights Reserved.

Module 4: Analytics Theory/Methods 137

The topics covered in this lesson are listed.



Text Analysis

Encompasses the processing and representation of text for analysis and learning tasks

- **High-dimensionality**
  - Every distinct term is a dimension
  - *Green Eggs and Ham*: A 50-D problem!
- **Data is Un-structured**

Green
Eggs
and
Ham
By Dr. Seuss

EMC^2 PROVEN PROFESSIONAL

Module 4: Analytics Theory/Methods 138

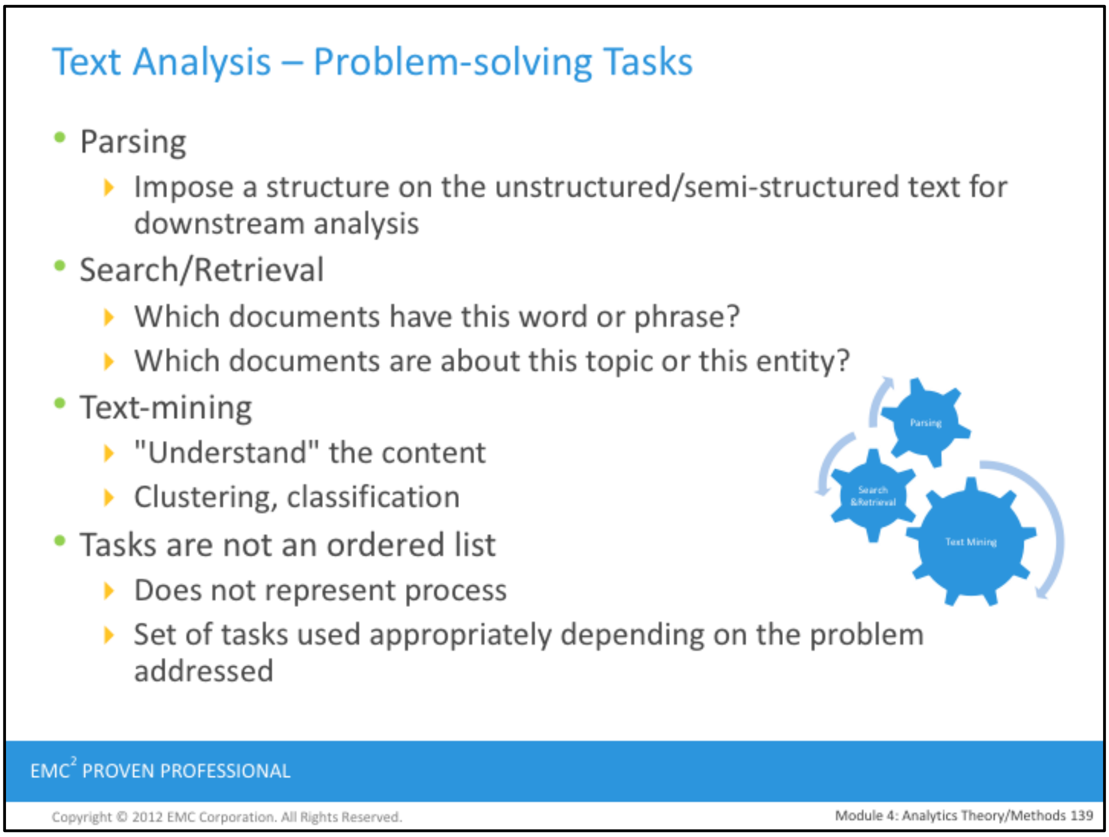
Text analysis is essentially the processing and representation of data that is in text form for the purpose of analyzing and learning new models from it.

The main challenge in text analysis is the problem of high dimensionality. When analyzing a document every possible word in the document represents a dimension.

Consider the book 'Green Eggs and Ham' by Dr. Seuss, which he wrote responding to a challenge to write a book with just fifty different words.

(http://en.wikipedia.org/wiki/Green_Eggs_and_Ham). Even this book represents a 50 dimension problem if we consider vectors in a text space.

The other major challenge with text analysis is that the data is unstructured.



Text Analysis – Problem-solving Tasks

- Parsing
  - Impose a structure on the unstructured/semi-structured text for downstream analysis
- Search/Retrieval
  - Which documents have this word or phrase?
  - Which documents are about this topic or this entity?
- Text-mining
  - "Understand" the content
  - Clustering, classification
- Tasks are not an ordered list
  - Does not represent process
  - Set of tasks used appropriately depending on the problem addressed

EMC^2 PROVEN PROFESSIONAL

Module 4: Analytics Theory/Methods 139

The process or the problem solving tasks in text analysis is composed of three important steps namely Parsing, Search/ Retrieval and Text mining.

Parsing is the process step that takes the un-structured or a semi-structured document and impose a structure for the downstream analysis. Parsing is basically reading the text which could be weblog, a RSS feed ,a XML or a HTML file or a word document. Parsing decomposes what is read in and renders it in a structure for the subsequent steps.

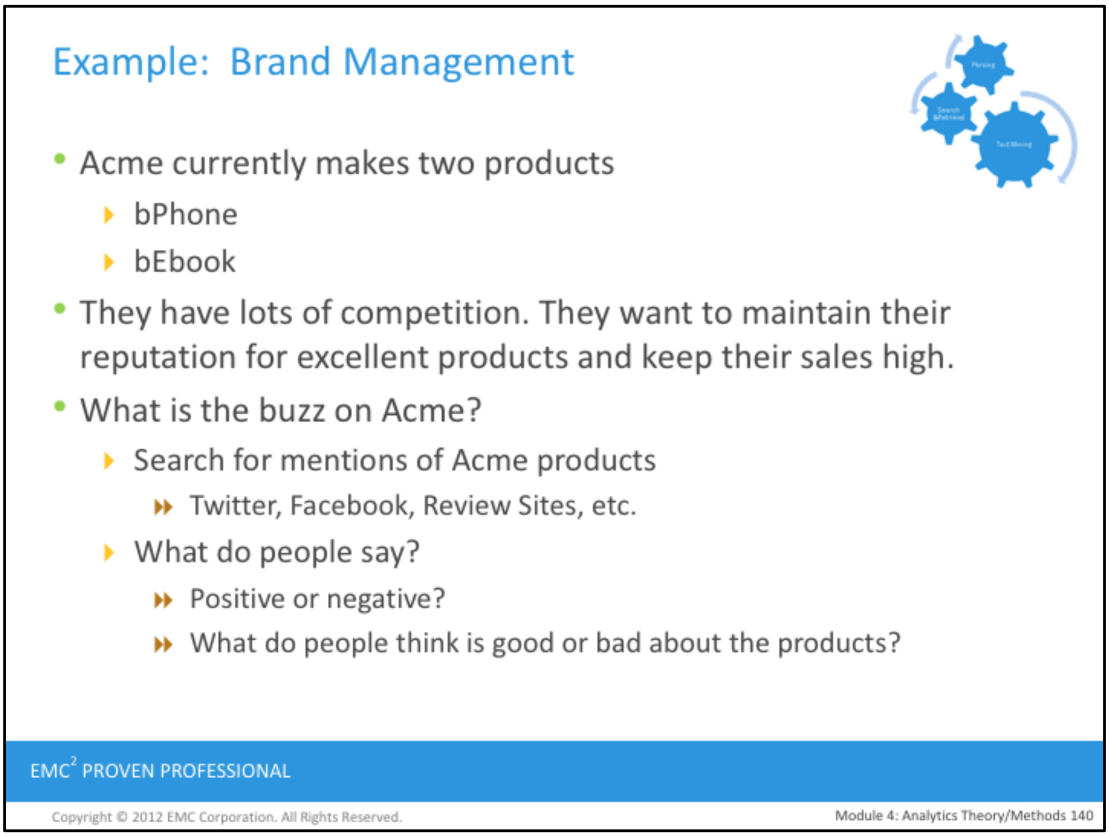
Once parsing is done, the problem focuses on **search and/or retrieval** of specific words or phrases or in finding a specific topic or an entity (a person or a corporation) in a document or a corpus (body of knowledge). All text representation takes place implicitly in the context of the corpus. All search and retrieval is something we are used to performing with search engines such as Google. Most of the techniques used in search and retrieval originated from the field of library science.

With the completion of these two steps, the output generated is a structured set of tokens or a bunch of key words that were searched, retrieved and organized. The third task is **mining the text** or understanding the content itself. Instead of treating the text as set of tokens or keywords, in this step we derive meaningful insights into the data pertaining to the domain of knowledge, business process or the problem we are trying to solve.

Many of the techniques that we mentioned in the previous lessons such as clustering and classification can be adapted to the text mining, with the proper representation of the text. We could use K-means clustering or other methods to tie the text into meaningful groups of subjects. Sentiment Analysis and Spam filtering are examples of a classification tasks in text mining. (recall that we listed Spam filtering as a prominent use case for Naïve Bayesian Classifier). In addition to traditional statistical methods, Natural Language processing methods are also used in this phase.

It should be noted the list of tasks are not ordered. One generally starts with the parsing, either with the intention of compiling them into a searchable corpus or catalog (maybe after some analytical tasks like tagging or categorization), OR specifically for the purpose of text mining. So it's not a process, it's a set of things that go into the text analysis task. Or maybe a tree, where you start with parsing, and go down to either search or to text-mining.

We will look into details of each of these steps in the rest of this lesson.


Parsing
Search &Retrieval
Text Mining

Example: Brand Management

- Acme currently makes two products
  - bPhone
  - bEbook
- They have lots of competition. They want to maintain their reputation for excellent products and keep their sales high.
- What is the buzz on Acme?
  - Search for mentions of Acme products
    - Twitter, Facebook, Review Sites, etc.
  - What do people say?
    - Positive or negative?
    - What do people think is good or bad about the products?

EMC² PROVEN PROFESSIONAL

Module 4: Analytics Theory/Methods 140

Here we present an example "Brand Management" to detail the concepts in text analysis throughout this lesson.

The company Acme makes two products bPhone and bEbook. Acme is not the only one in the market making similar products. The competition is stiff and they want to maintain the reputation they have among e-book readers as an excellent product offering and also to enhance their sales.

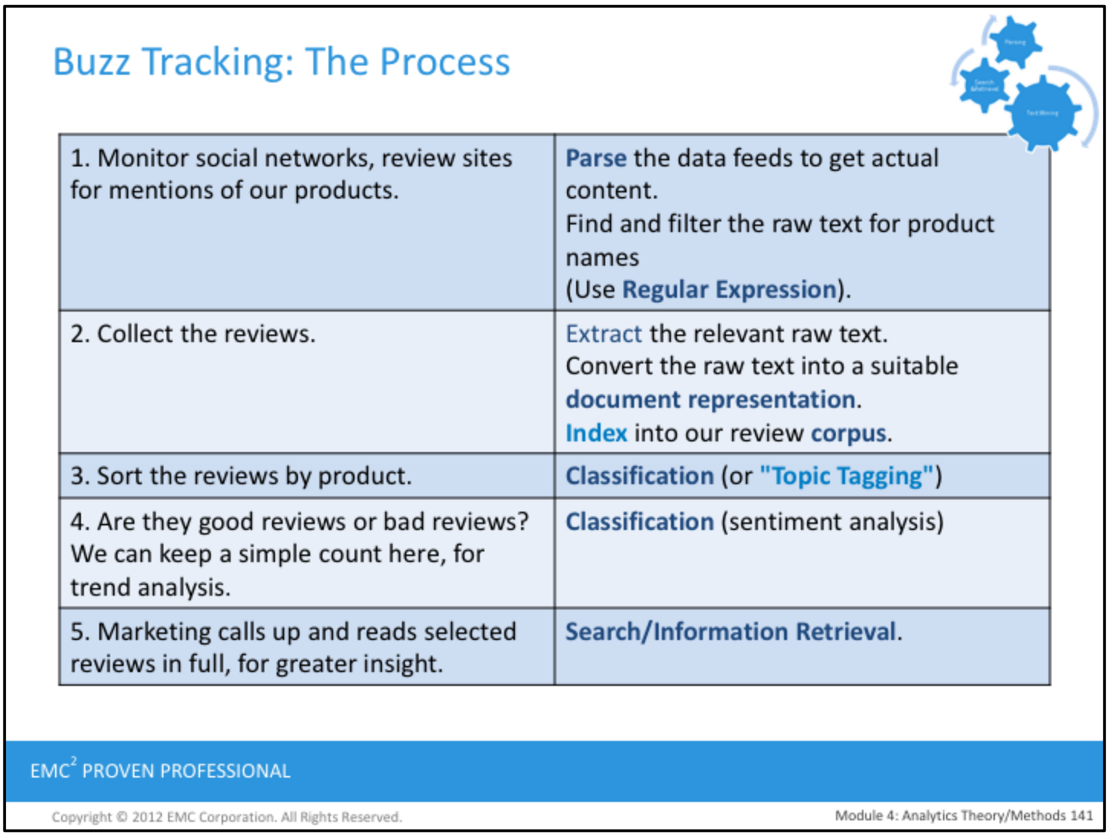
One of the ways they do this is to monitor what is being said about Acme products in the social media. In other words what is the buzz on Acme products. They want to search all that is said about Acme products in Twitter, Facebook and popular review sites (Amazon).

They want to know:

a) If people are mentioning their products?

b) What is being said – good or bad about the products. What people think is good or bad about Acme products. For example are they complaining about the battery life of the bPhone, or the latency in their bEbook.

A full example would ask "how does bPhone compare to the competition, but let's keep the example simple.


Parsing
Search &Retrieval
Text Mining

Buzz Tracking: The Process

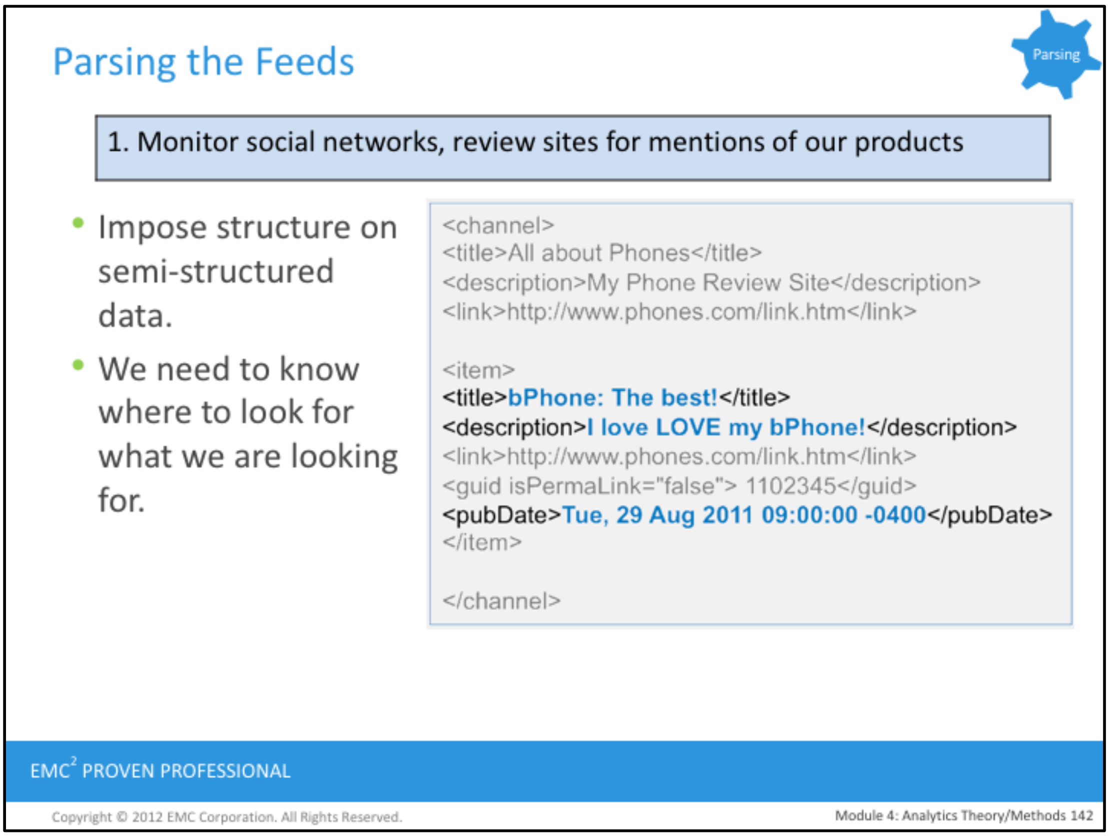
1. Monitor social networks, review sites for mentions of our products.	**Parse** the data feeds to get actual content. Find and filter the raw text for product names (Use **Regular Expression**).
2. Collect the reviews.	Extract the relevant raw text. Convert the raw text into a suitable **document representation**. **Index** into our review **corpus**.
3. Sort the reviews by product.	**Classification** (or **"Topic Tagging"**)
4. Are they good reviews or bad reviews? We can keep a simple count here, for trend analysis.	**Classification** (sentiment analysis)
5. Marketing calls up and reads selected reviews in full, for greater insight.	**Search/Information Retrieval**.

EMC^2 PROVEN PROFESSIONAL

Here we present a hypothetical and vastly oversimplified example of a process that you can adopt for the tracking what is said about Acme.

The first column of the table lists the tasks carried out for the buzz tracking and the second column lists the corresponding text analysis tasks associated with the established buzz tracking process.

The process is merely a way to organize the topics we present in this lesson, and to call out some of the difficulties that are unique to text mining.



Parsing the Feeds

Parsing

1. Monitor social networks, review sites for mentions of our products

- Impose structure on semi-structured data.
- We need to know where to look for what we are looking for.

```
<channel>
<title>All about Phones</title>
<description>My Phone Review Site</description>
<link>http://www.phones.com/link.htm</link>

<item>
<title>bPhone: The best!</title>
<description>I love LOVE my bPhone!</description>
<link>http://www.phones.com/link.htm</link>
<guid isPermaLink="false"> 1102345</guid>
<pubDate>Tue, 29 Aug 2011 09:00:00 -0400</pubDate>
</item>

</channel>
```

EMC^2 PROVEN PROFESSIONAL

Copyright © 2012 EMC Corporation. All Rights Reserved.

Module 4: Analytics Theory/Methods 142

Parsing in the linguistic sense means "to resolve a sentence into component parts of speech and explain syntactical relationships". (Merriam-Webster)

First, we want to monitor the data feeds, and parse them.

In this context, we are talking about parsing semi-structured data: html pages, RSS feeds, or whatever we may have.

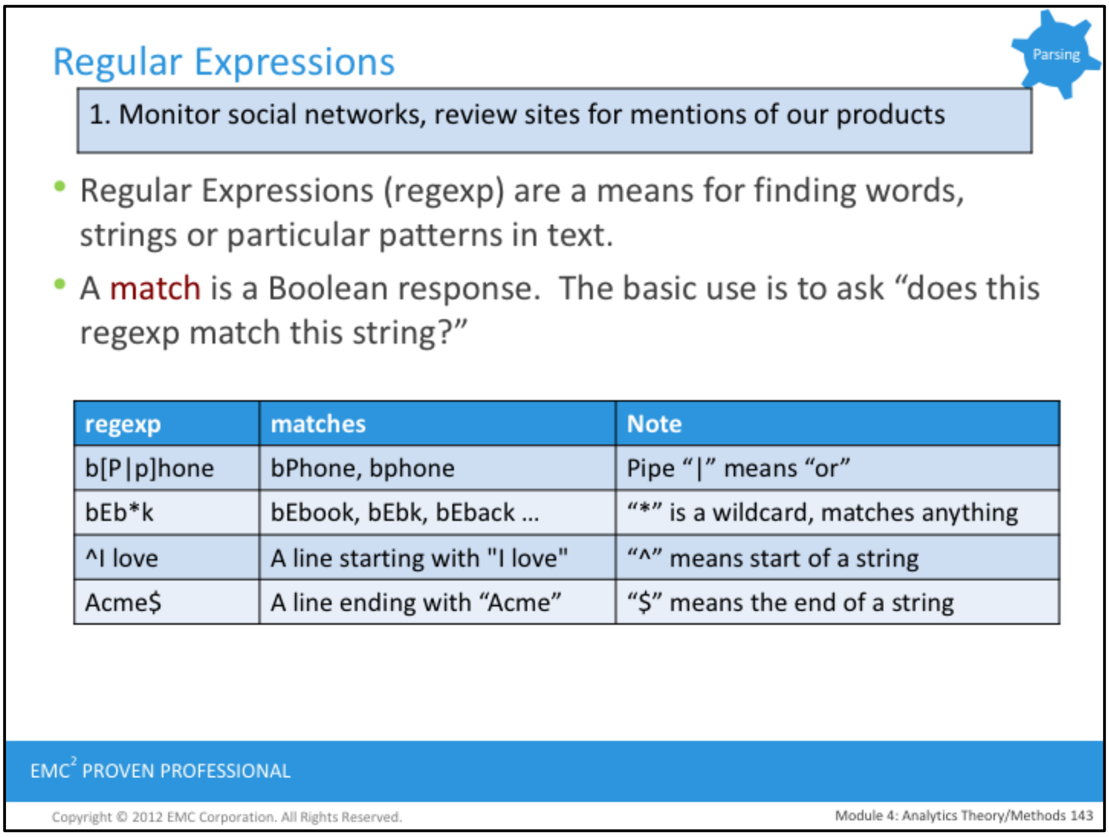
We need to impose enough structure so we can find the part of the raw text that we really care about -- in this case the actual content of review (including their titles), and when the reviews were posted.

This requires knowing the grammar of the data source. Sometimes it's relatively standard – HTML, RSS. Other times, it may not be quite as standard (web logs, for instance).

– **As an example**, An RSS (Really Simple Syndication) feed for a **smart phone review blog** is shown in the slide.

What is highlighted in the RSS feed shown here are the contents we are interested in. The “title”, “Description” and the “date”.

Once we know where to look, we can determine if it's what we are looking for.



Parsing

Regular Expressions

1. Monitor social networks, review sites for mentions of our products

- Regular Expressions (regexp) are a means for finding words, strings or particular patterns in text.
- A match is a Boolean response. The basic use is to ask “does this regexp match this string?”

regexp	matches	Note
b[P|p]hone	bPhone, bphone	Pipe “|” means “or”
bEb*k	bEbook, bEbk, bEback …	“*” is a wildcard, matches anything
^I love	A line starting with "I love"	“^” means start of a string
Acme$	A line ending with “Acme”	“$” means the end of a string

EMC^2 PROVEN PROFESSIONAL

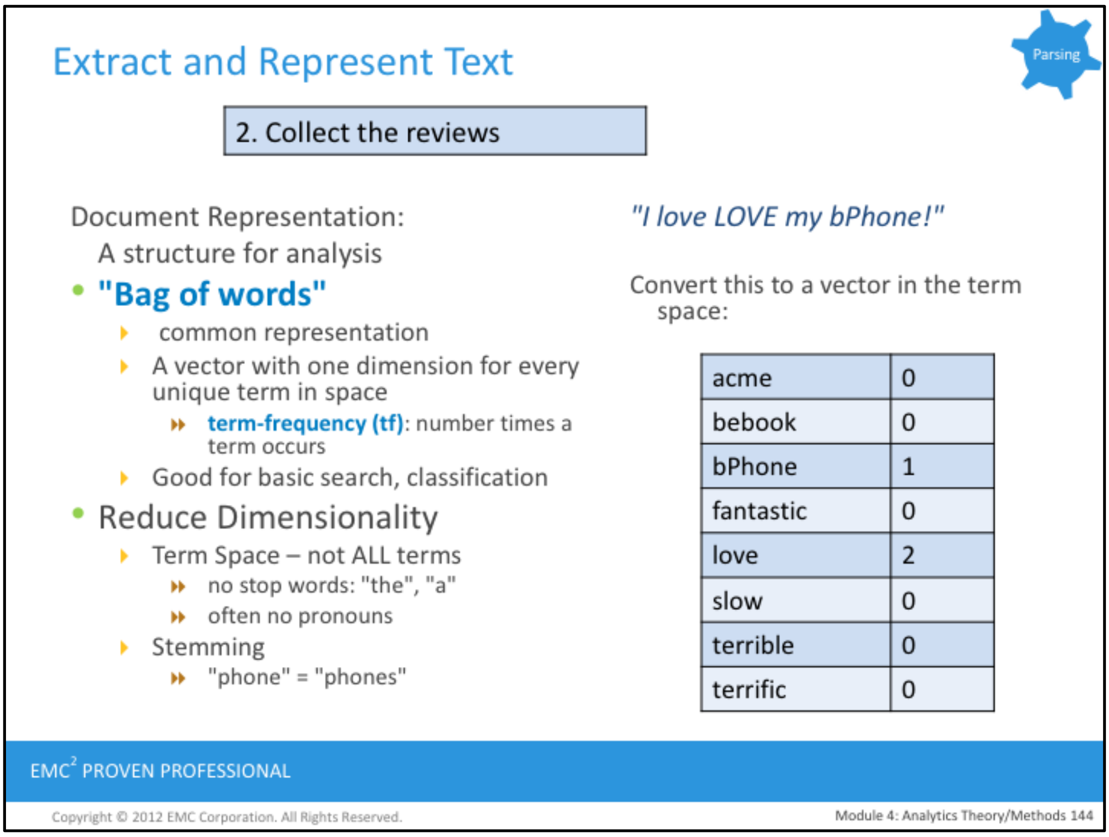
Module 4: Analytics Theory/Methods 143

Regular Expressions is a popular technique used for finding words, strings or a particular patterns in the text. We will explore regular expression later in detail in Module 5.

The basic use is to determine if the regular expression (regexp) matches this string.

We have shown some examples of syntax used in regexp above. It is beyond the scope of this lesson to go into the details of the regexp syntax. But the general idea is that once we have the content from the fields of interest, we want to know if it is of interest to us. In this case: do those fields mention bPhone, bEbook, or Acme?

With regular expressions we can take into account capitalization (or lack of it), common misspellings, common abbreviations etc.



Extract and Represent Text

Parsing

2. Collect the reviews

Document Representation:
A structure for analysis

- **"Bag of words"**
  - common representation
  - A vector with one dimension for every unique term in space
    - **term-frequency (tf)**: number times a term occurs
  - Good for basic search, classification
- Reduce Dimensionality
  - Term Space – not ALL terms
    - no stop words: "the", "a"
    - often no pronouns
  - Stemming
    - "phone" = "phones"

"I love LOVE my bPhone!"

Convert this to a vector in the term space:

acme	0
bebook	0
bPhone	1
fantastic	0
love	2
slow	0
terrible	0
terrific	0

EMC^2 PROVEN PROFESSIONAL

Copyright © 2012 EMC Corporation. All Rights Reserved.

Module 4: Analytics Theory/Methods 144

We are now in Step 2 . We have parsed all our data feeds and collected the phrases and words and we are ready to represent what we collected in a structured manner for down stream analysis.

The most common representation of the structure is known as the “bag of words”. The “Bag of words” is a vector with one dimension for every unique term in the space.

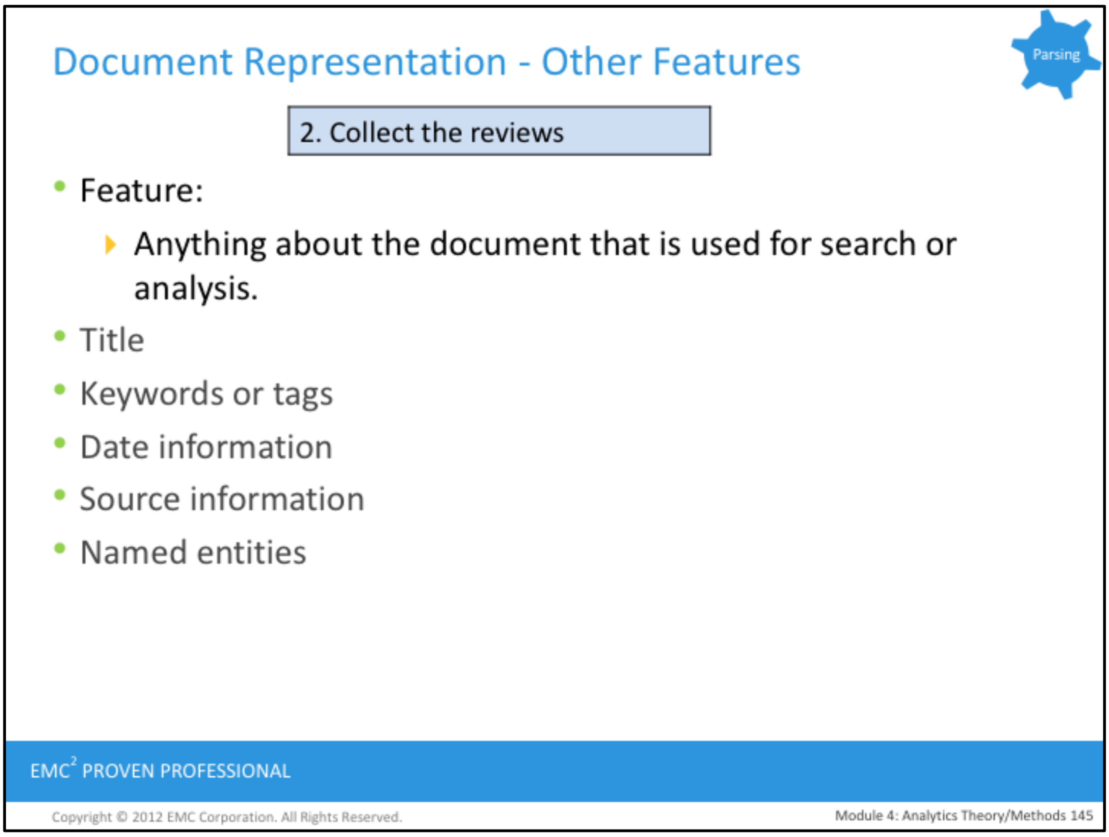
We also introduce the term “term-frequency” (tf) which is the number of times a term occurs in a vector.

Obviously the vector is VERY high-dimensional as we invariably end up with a significant number of unique words in a document. “Bag of words is a common representation and it is suited very well for search and classification. There are more sophisticated representations for sophisticated algorithms.

In the example above, the RSS feeds we parsed “I love LOVE my bPhone”, (we are only showing the part of our vector space).

We count the occurrences of the words in the text parsed and number of times the word is repeated and store word count as a part of the vector representation. In our example we see bPhone mentioned once and “love” mentioned twice.

In order to reduce the dimensionality we do not include all words in the English language. Normally we ignore some “stop” words such as “the” “a” etc. There are other methods such as stemming the words and avoiding pronouns in the term space. Vector space must be managed in a way so that it only contains words that are essential for the analysis. Stemming is done based on the context and corpus. In a completely unstructured document techniques such as “parts of speech tagging” are used for parsing.



Document Representation - Other Features

2. Collect the reviews

- Feature:
  - Anything about the document that is used for search or analysis.
- Title
- Keywords or tags
- Date information
- Source information
- Named entities

EMC^2 PROVEN PROFESSIONAL

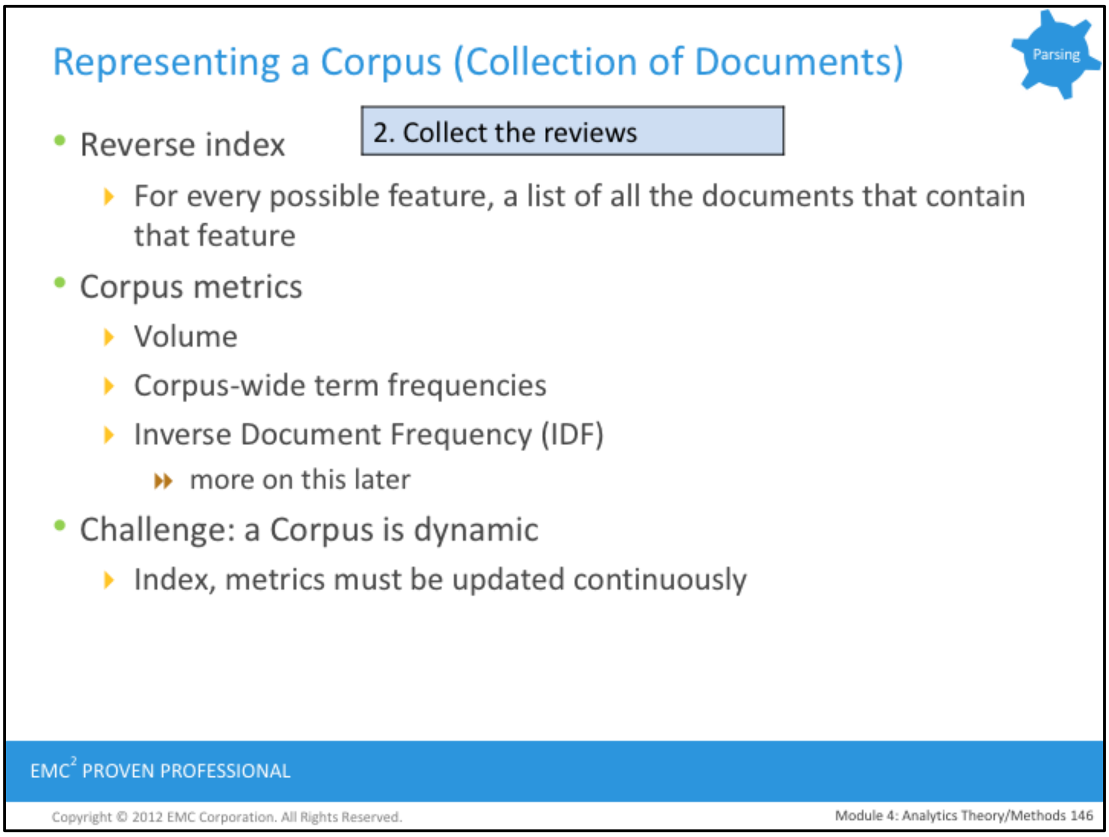
Copyright © 2012 EMC Corporation. All Rights Reserved.

Module 4: Analytics Theory/Methods 145

In addition to the “term, the features we store are the title of the document, any key words or tags attached to it, the date the document was created, the source from where the document was extracted (twitter, facebook, Amazon etc.) and some of the Named entities such as a mention of a competitor’s name (do they compare bPhone to iPhone ?).

Sometimes creating these features is a text analysis task all to itself, like topic tagging. Companies invest significant resources in creating these tags as a separate activity. You see people tag their blogs to enable easy search and retrieval.

These features help with down stream analysis in classification or sentiment analysis.



Representing a Corpus (Collection of Documents)

Parsing

2. Collect the reviews

- Reverse index
  - For every possible feature, a list of all the documents that contain that feature
- Corpus metrics
  - Volume
  - Corpus-wide term frequencies
  - Inverse Document Frequency (IDF)
    - more on this later
- Challenge: a Corpus is dynamic
  - Index, metrics must be updated continuously

EMC^2 PROVEN PROFESSIONAL

Module 4: Analytics Theory/Methods 146

It is important that we not only create a representation of the document but we also need to represent a corpus. What is the representation of a corpus?

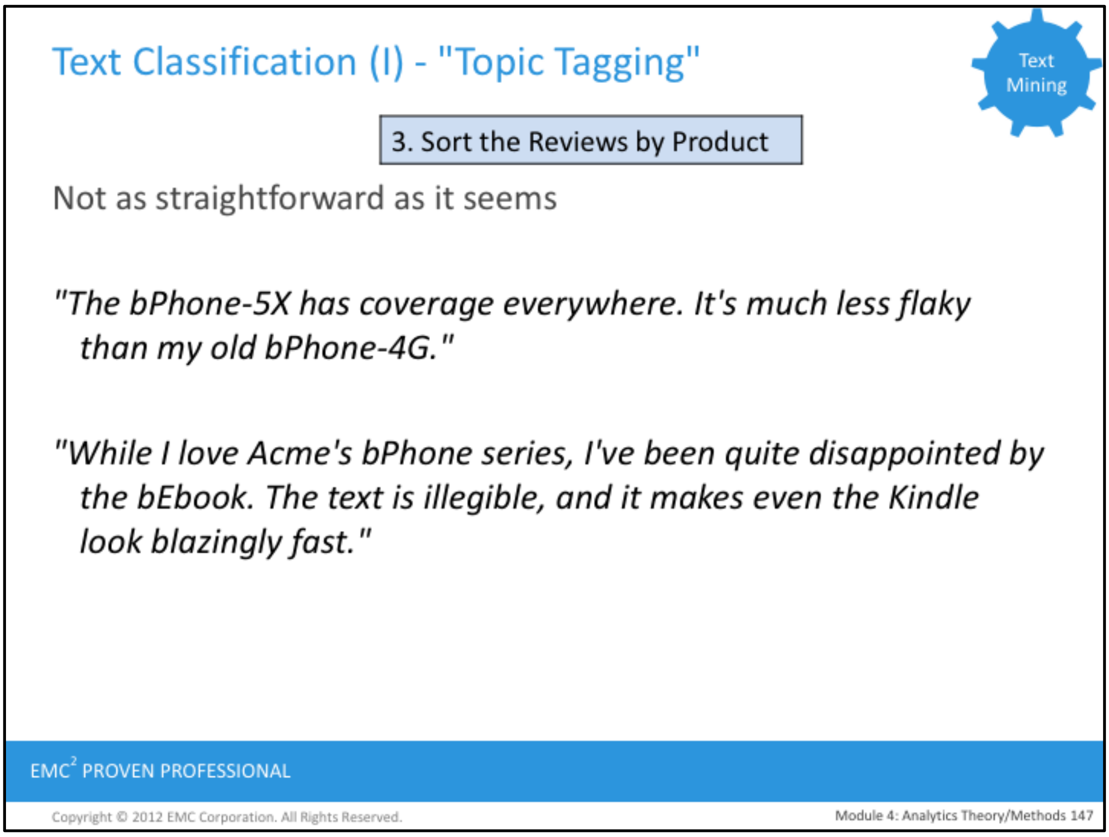
Now that we've collected the reviews and turned them into the proper representation, we want to archive them in a searchable archive for future reference and research. This is done with “reverse index” which provides a way of keeping track of list of all documents that contain a specific feature and for every possible feature.

The other corpus metrics such as volume and corpus-wide term frequency, *which specifies how the terms are distributed across the corpus, help with the down stream analysis of classification and searching. Search algorithms also* inverse document frequency which we define later in this lesson.

A fact that many people don't think about is that documents are often only relevant in the context of a corpus, or a specific collection of documents. Sometimes this is obvious, as in the case of search or retrieval. It is less obvious in the case of classification (for example, spam filtering, sentiment analysis) – but even in that case, the classifier has been trained on a specific set of documents, and the underlying assumption of all classifiers is that it will be deployed on a population that is similar to the population that it was trained on.

A primary challenge in text analysis and search is that a corpus changes constantly over time: not only do new documents get added (which means the metrics and indices must be updated), but word distributions can change over time (which will reduce the effectiveness of classifiers and filters, if they are not retrained – think about spam filters).

The corpus representation that we discuss here is primarily oriented towards search/retrieval, but some of the metrics, like IDF can also be relevant to classification as well.



Text Classification (I) - "Topic Tagging"

Text Mining

3. Sort the Reviews by Product

Not as straightforward as it seems

"The bPhone-5X has coverage everywhere. It's much less flaky than my old bPhone-4G."

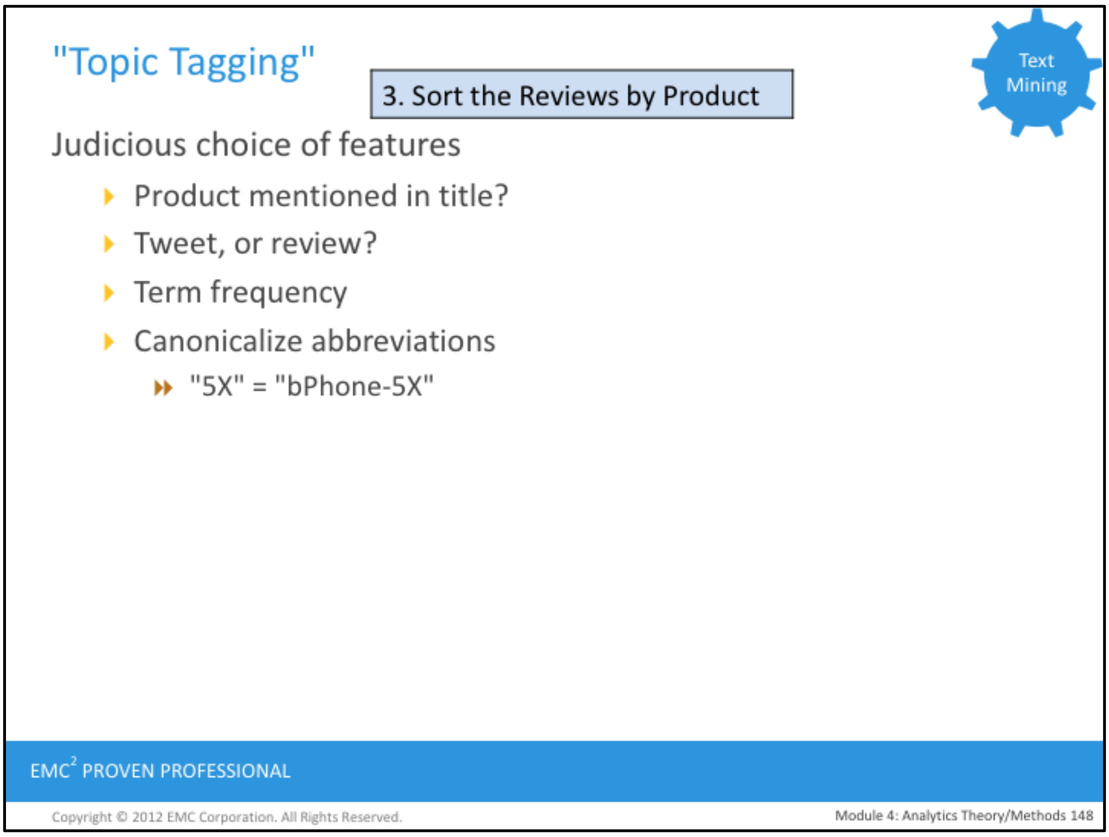
"While I love Acme's bPhone series, I've been quite disappointed by the bEbook. The text is illegible, and it makes even the Kindle look blazingly fast."

EMC^2 PROVEN PROFESSIONAL

Now all the reviews are collected and represented we want to sort them by product. This is done with topic tagging. For the two reviews shown:

- Is the first review about bPhone-5x or bPhone-4g?
- Is the second review is about bPhone or bEbook or Kindle?

It is a complex problem to properly tag a document and it is not as straightforward as it appears. There are several methods available such as simply counting the number of occurrences of a product name to many sophisticated methods. More on this in the following slide.



"Topic Tagging"

3. Sort the Reviews by Product

Text Mining

Judicious choice of features

- Product mentioned in title?
- Tweet, or review?
- Term frequency
- Canonicalize abbreviations
  - "5X" = "bPhone-5X"

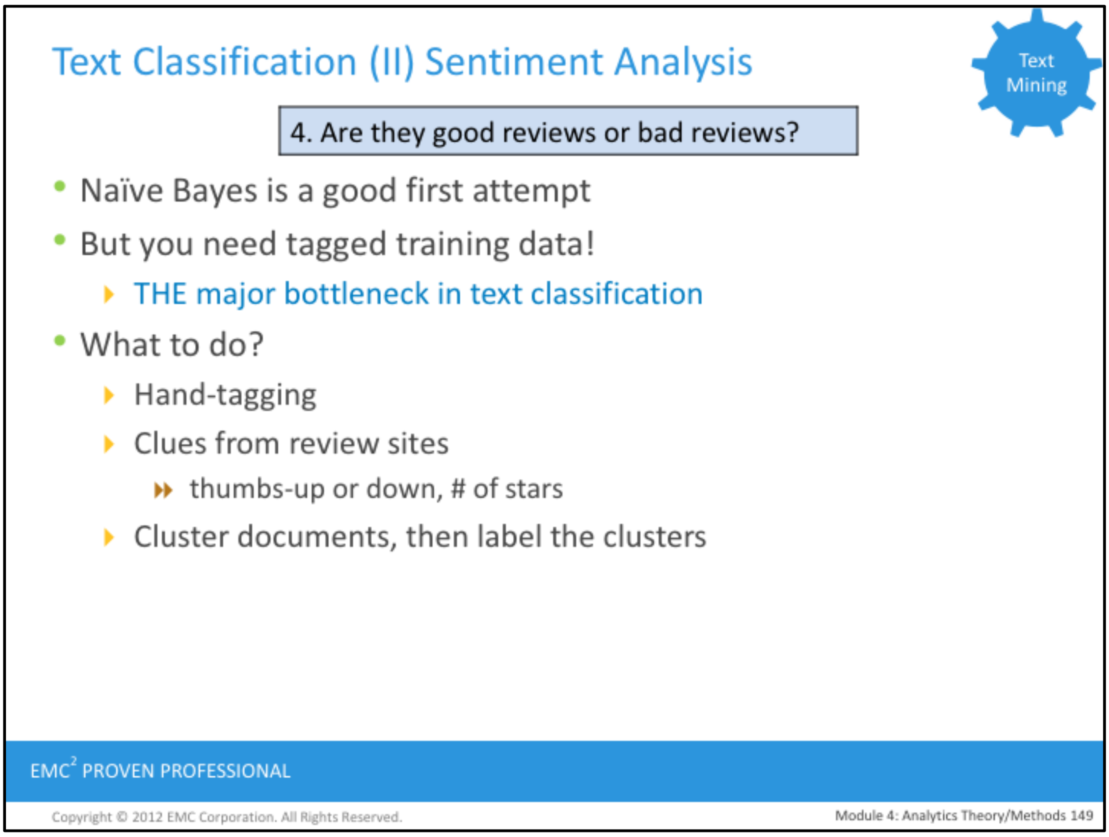
EMC^2 PROVEN PROFESSIONAL

Module 4: Analytics Theory/Methods 148

There are rules you can come up with to determine how to sort a document (in a given context).

If the bphone5X is mentioned in the title, then the document is likely to be about the 5X, and mentions of the 4G in the text may or may not be relevant (to tagging). A tweet that mentions the product is probably about the product (whereas a review may mention many products as comparisons). More frequent mentions of the product in the document are a clue. Somewhere, you need to resolve abbreviations into the correct product (in the term space).

One could manually compile these rules (dirty secret – many folks do). Ideally, the Data Scientist should have a good idea what the relevant features are for a given task, and structure the document representation to fit both the explanatory features, and the algorithm that is used to do the classification/tagging. This process is part of the Data Analytics Lifecycle we discussed in Module 2.



Text Classification (II) Sentiment Analysis

Text Mining

4. Are they good reviews or bad reviews?

- Naïve Bayes is a good first attempt
- But you need tagged training data!
  - THE major bottleneck in text classification
- What to do?
  - Hand-tagging
  - Clues from review sites
    - thumbs-up or down, # of stars
  - Cluster documents, then label the clusters

EMC^2 PROVEN PROFESSIONAL

Copyright © 2012 EMC Corporation. All Rights Reserved.

Module 4: Analytics Theory/Methods 149

At this point in the process, Acme already has a sentiment classification engine; here we are going to discuss how one might build one.

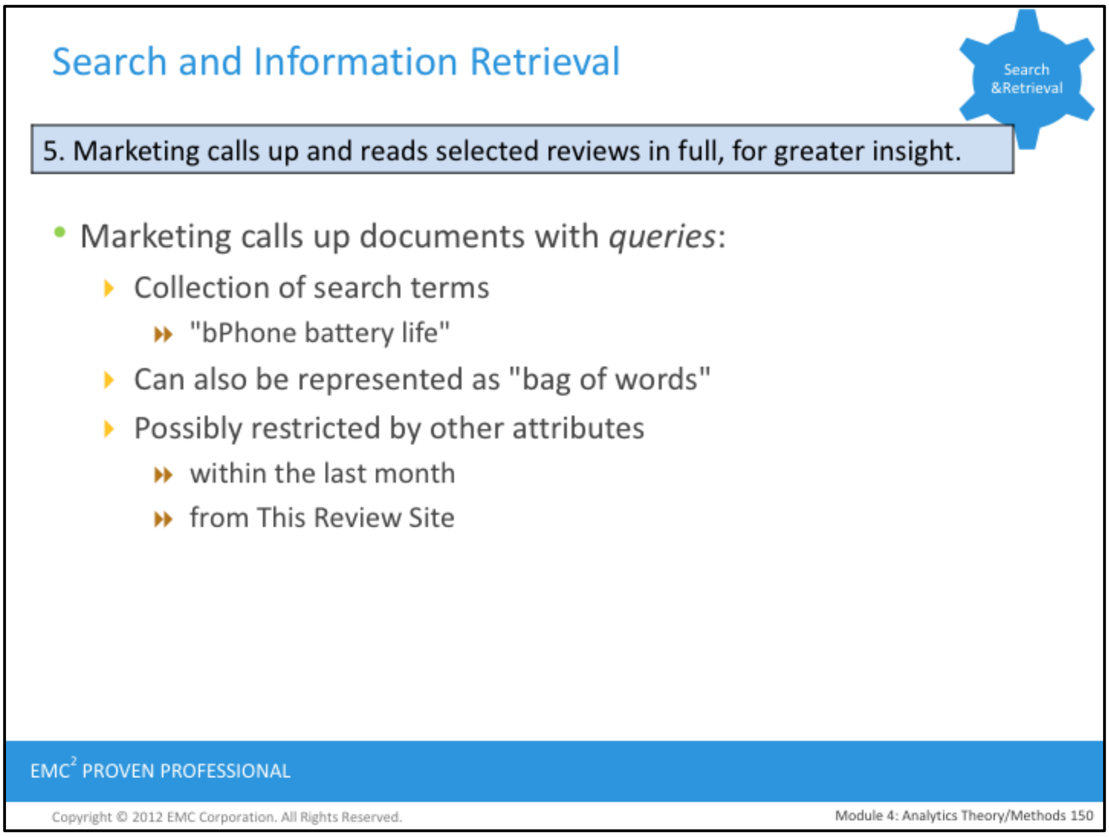
The take-away here is that the challenge in text classification is often not the algorithm; it's getting the tagged data.

Many companies, like Amazon or Shopping.com, rely on teams of hand-taggers to create training corpora to jump-start efforts in automated categorization. Hand-tagged data is slow to collect, and is prone to fatigue errors and inconsistent (subjective) tagging on the part of the taggers.

In the case of sentiment analysis, one could try creating training corpora based on sites that have quantitative ratings for the products; the resulting classifiers run the risk of only being effective on the reviews from sites that they came from (or for reviews from that product category), because of idiosyncratic terminology of the website community, or the product category. As an example, "lightweight" is a positive adjective for laptops, but not necessarily for wheelbarrows, or books. Classifiers built from reviews would almost definitely not work on tweets or blog comments.

Using unsupervised methods to cluster the documents, and then assigning labels based on whether or not the sampled documents from a cluster are positive or negative might work – but since the cluster is not built specifically on sentiment, it may not partition on sentiment.

There are other things you can do to track sentiment, besides classification: for instance, you can track the frequency with which certain words appear in reviews of your products, and then let a human decide if the overall trend looks positive or negative. The point of this discussion is not to cover all the possible ways of text mining, but to cover the basic concepts and issues.



Search and Information Retrieval

Search &Retrieval

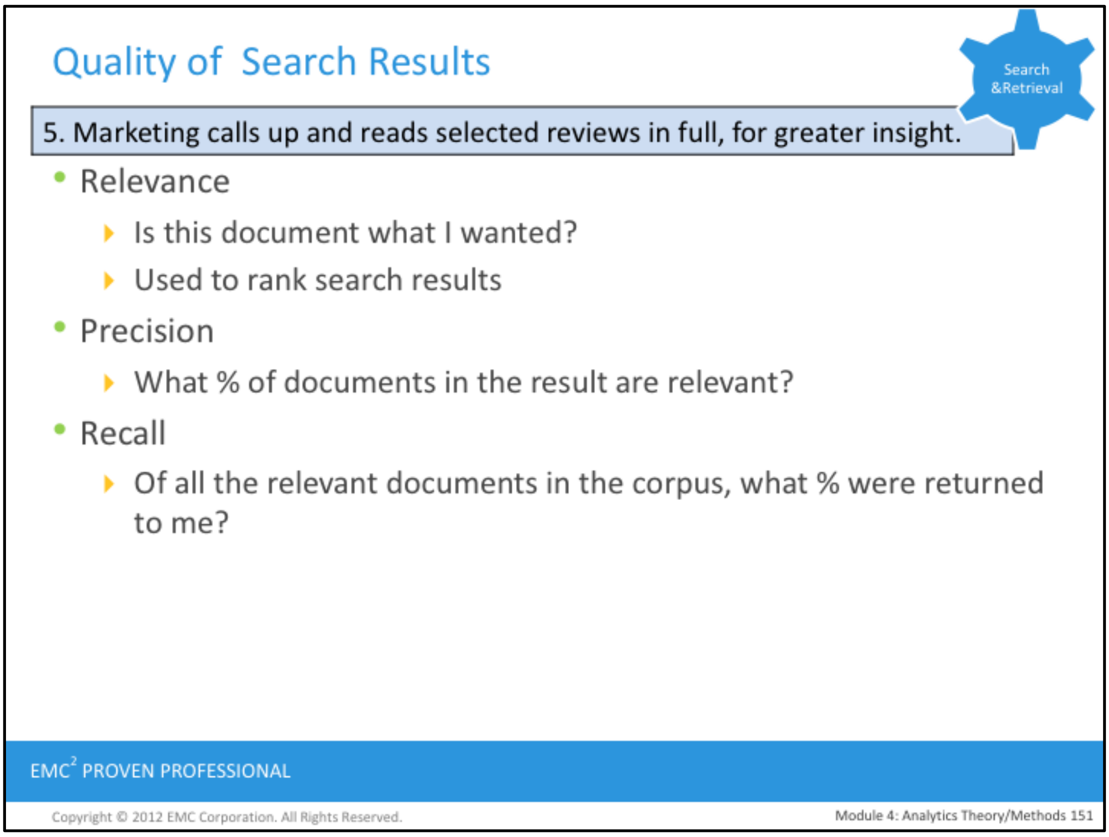
5. Marketing calls up and reads selected reviews in full, for greater insight.

- Marketing calls up documents with *queries*:
  - Collection of search terms
    - "bPhone battery life"
  - Can also be represented as "bag of words"
  - Possibly restricted by other attributes
    - within the last month
    - from This Review Site

EMC^2 PROVEN PROFESSIONAL

Module 4: Analytics Theory/Methods 150

Finally we got our corpus created tags and we have some sentiment analysis and the marketing team wants to call up these documents. This is typically done with a query which may specify calling up of documents from a particular site or reviews in a specific data range. This basically is a search problem, finding the document that meets the search criteria.



Quality of Search Results

Search &Retrieval

5. Marketing calls up and reads selected reviews in full, for greater insight.

- Relevance
  - Is this document what I wanted?
  - Used to rank search results
- Precision
  - What % of documents in the result are relevant?
- Recall
  - Of all the relevant documents in the corpus, what % were returned to me?

EMC^2 PROVEN PROFESSIONAL

Module 4: Analytics Theory/Methods 151

Let us now focus on the quality of search results. It basically is determining if the results you receive are indeed the ones you wanted or not. Relevance, precision and recall are the metrics that are used to determine the quality of search results.

We come up with an objective measure of relevance (Is this the document the user wanted) and rank the search results based on **Relevance** and provide users the most relevant documents ahead of those that score low on relevance.

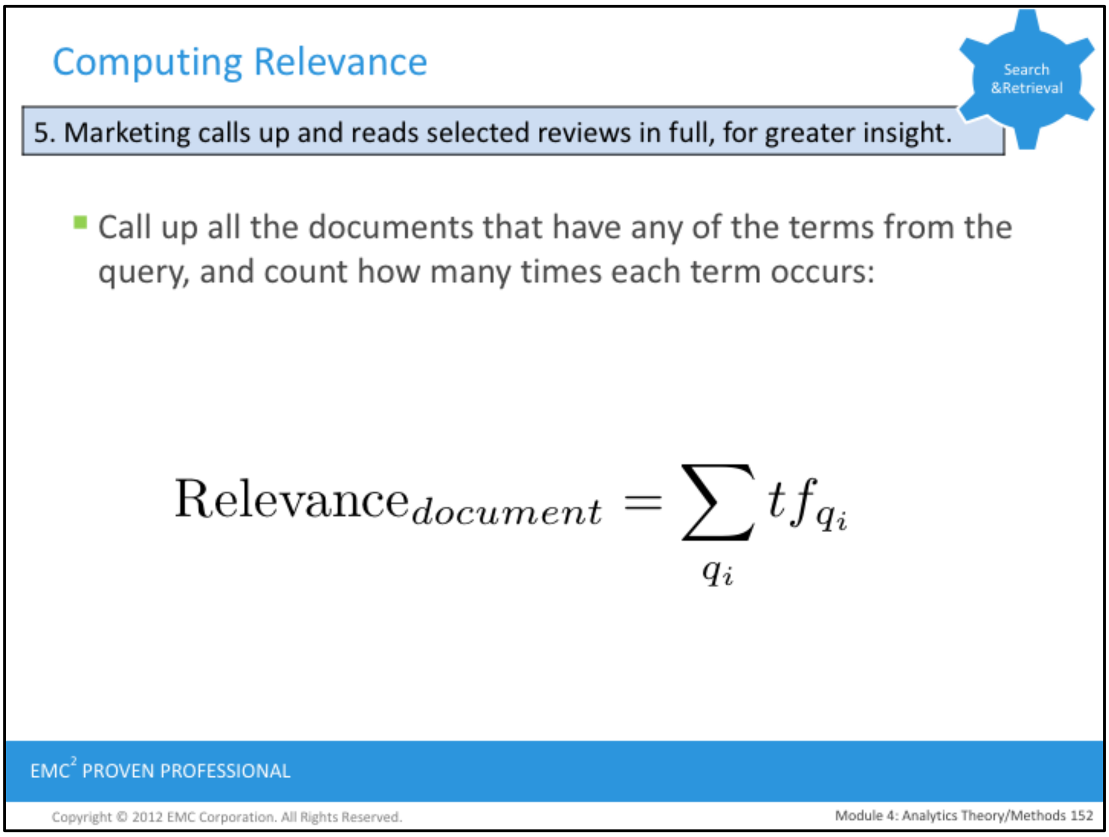
Precision and **Recall** are measures of accuracy of the search. Precision is defined as the % of documents in the results that are relevant. If we say bPhone and it gives back a 100 documents and 70 of them are relevant the precision is 70%.

Recall is the % of returned documents among all relevant documents in the corpus.

Relevance and Precision are always important concepts, whether you are talking about a web search or information retrieval from a finite corpus (like our review archive).

Recall is basically a meaningless concept when you are discussing general web search. Or to put it another way: it will probably always be low, you just hope it's not zero. But it may be relevant in finite corpus.

Search algorithms (and classification algorithms, in general) are usually evaluated in terms of precision and recall by the computer science community.



Computing Relevance

Search
&Retrieval

5. Marketing calls up and reads selected reviews in full, for greater insight.

- Call up all the documents that have any of the terms from the query, and count how many times each term occurs:

$$\text{Relevance}_{document} = \sum_{q_i} tf_{q_i}$$

EMC^2 PROVEN PROFESSIONAL

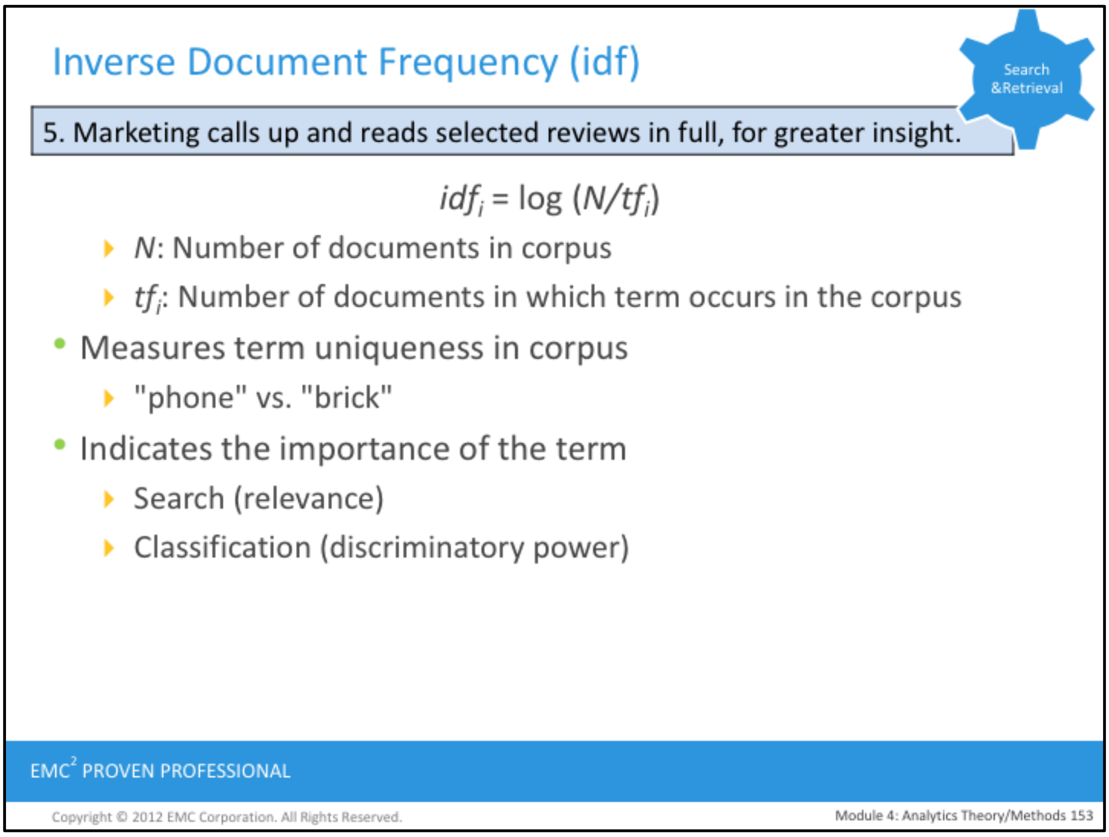
Copyright © 2012 EMC Corporation. All Rights Reserved.

Module 4: Analytics Theory/Methods 152

Here we present a simple example of how relevance might be computed.

We call up all the documents that have any of the terms from the query and count how many times each term occurs. For example the more often “bPhone” and “Battery Life” are mentioned in the document the more relevant the document is.

Obviously, there are ways to improve this method. For example, one might prefer documents that include ALL the terms, not just any. Also, one might want to limit the weight accorded to any one term ("Spam spam spam spam, wonderful spam….").



Inverse Document Frequency (idf)

Search
&Retrieval

5. Marketing calls up and reads selected reviews in full, for greater insight.

$$idf_i = \log(N/tf_i)$$

- *N*: Number of documents in corpus
- tf_i: Number of documents in which term occurs in the corpus

- Measures term uniqueness in corpus
  - "phone" vs. "brick"
- Indicates the importance of the term
  - Search (relevance)
  - Classification (discriminatory power)

EMC² PROVEN PROFESSIONAL

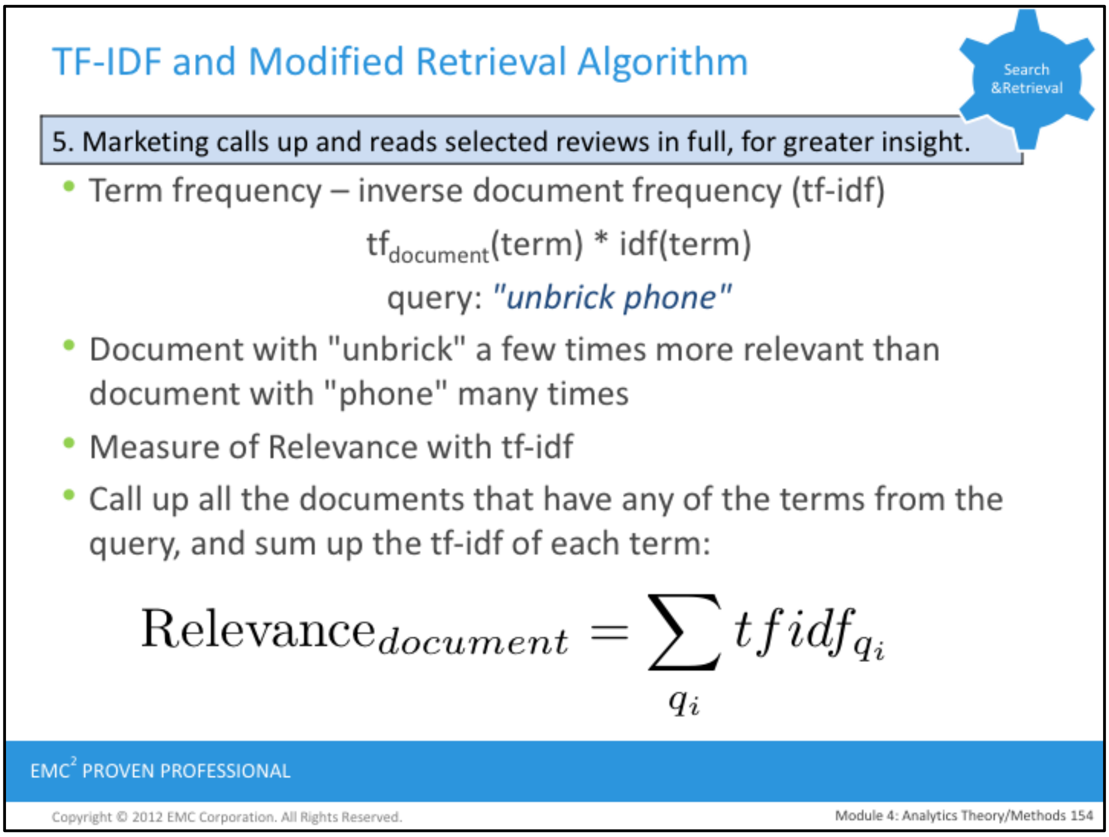
Copyright © 2012 EMC Corporation. All Rights Reserved.

Module 4: Analytics Theory/Methods 153

We now define Inverse Document Frequency and look into how we can improve our search algorithm with idf.

idf measures the uniqueness of a term in the corpus. If a term shows up only in 10% of the documents then it is unique. If a term shows up in 90% of the documents then it is not all that unique. It indicates the importance of the term (that appears in 10% of documents) and **provides relevance to the search by weighing the rare term higher**.

In a corpus of phone reviews, the word "phone" is probably pretty common; in particular it shows up in both good and bad reviews. The term "brick" is probably less common. So it is an important term when it shows up in a query (it discriminates relevant documents better than "phone" does), and potentially is distributed differently in good reviews and bad reviews. IDF reflects the fact that "brick" is potentially an interesting feature of a document.



TF-IDF and Modified Retrieval Algorithm

Search
&Retrieval

5. Marketing calls up and reads selected reviews in full, for greater insight.

- Term frequency – inverse document frequency (tf-idf)

  $\text{tf}_{\text{document}}(\text{term}) * \text{idf}(\text{term})$

  query: *"unbrick phone"*

- Document with "unbrick" a few times more relevant than document with "phone" many times
- Measure of Relevance with tf-idf
- Call up all the documents that have any of the terms from the query, and sum up the tf-idf of each term:

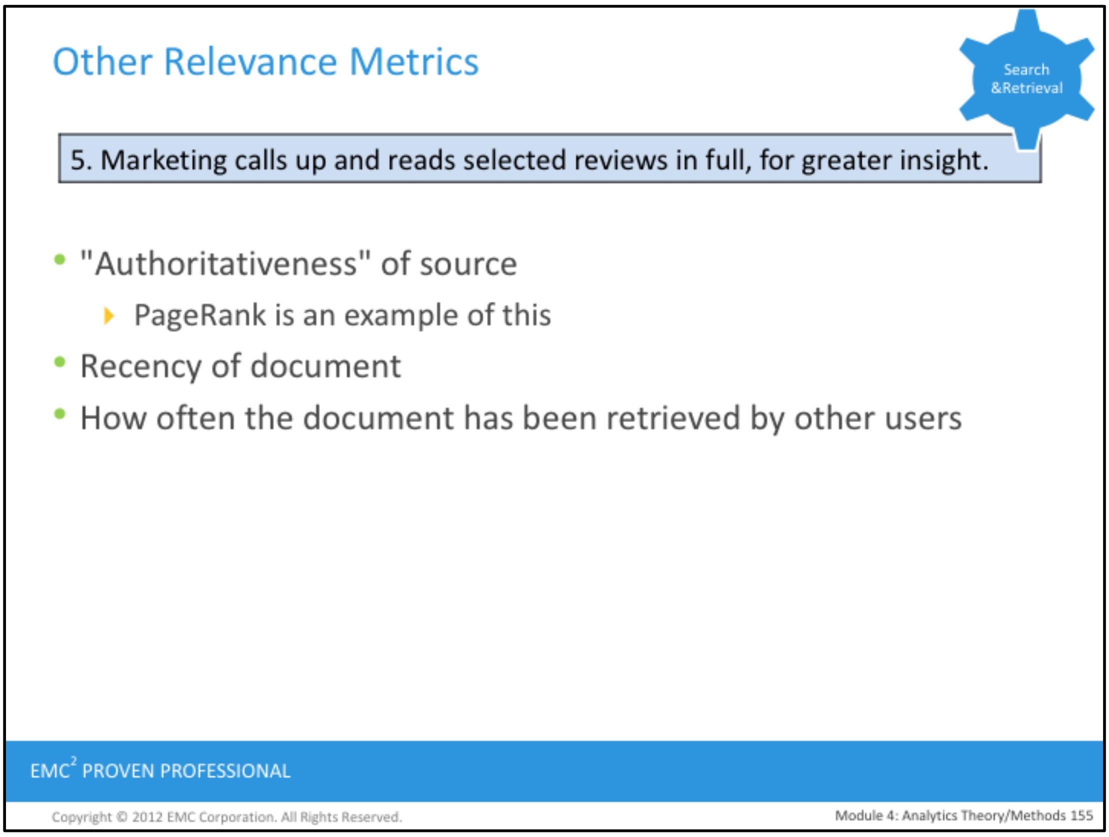
$$\text{Relevance}_{document} = \sum_{q_i} tfidf_{q_i}$$

EMC^2 PROVEN PROFESSIONAL

tf-idf is the product of term frequency (tf) and inverse document frequency (idf). **It provides measure that will weight the presence of unusual terms in the query as higher indications of document relevance than the presence of more common terms.**

In our query example “unbrick phone” tf-idf ensures that documents with “unbrick” are made more relevant than the document with “phone”.

We use the relevance as the sum of tf-idf and this modification to the search algorithm will yield better results in this corpus.



Other Relevance Metrics

Search
&Retrieval

5. Marketing calls up and reads selected reviews in full, for greater insight.

- "Authoritativeness" of source
  - PageRank is an example of this
- Recency of document
- How often the document has been retrieved by other users

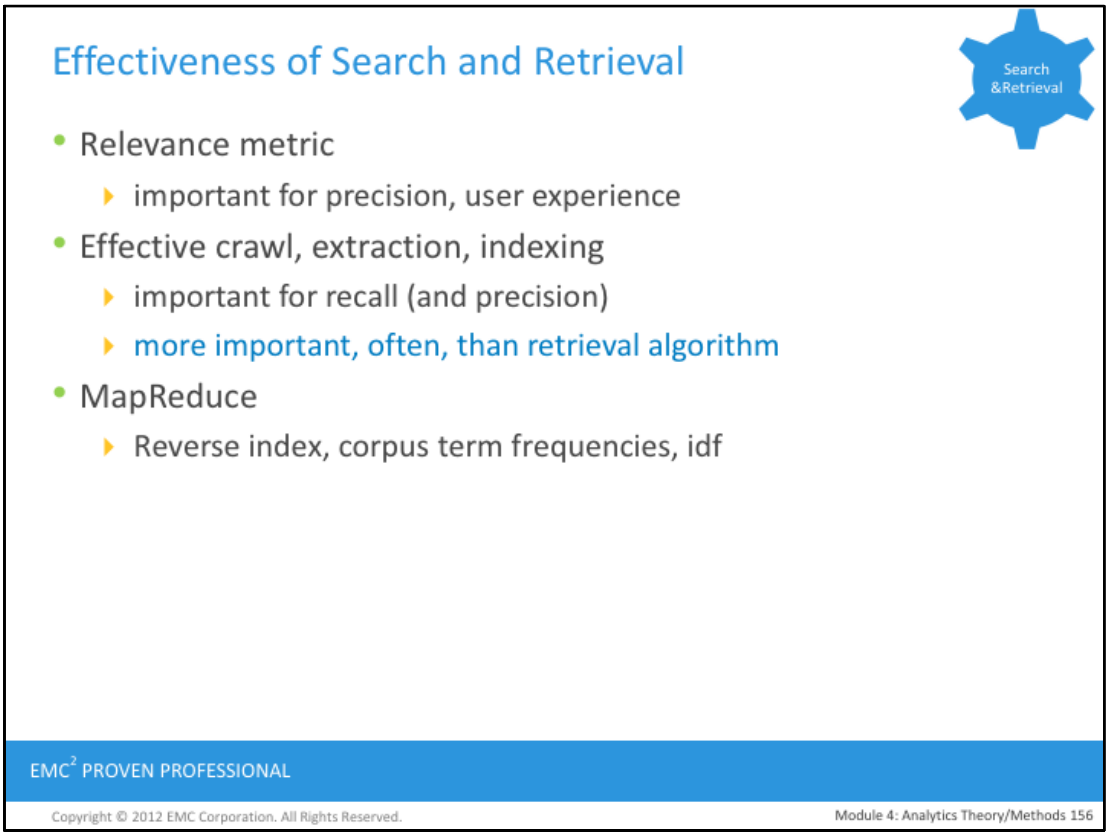
EMC^2 PROVEN PROFESSIONAL

There are other measures of relevance that are usually used in conjunction with term-based (for example, tfidf) relevance.

Authoritativeness of source is one such measure (PageRank – used by Google is an example)

Recency – new documents are more relevant than old ones

Keeping records of how often a document is retrieved as part of the corpus metrics by other users also provides a relevancy measure.



Effectiveness of Search and Retrieval

Search &Retrieval

- Relevance metric
  - important for precision, user experience
- Effective crawl, extraction, indexing
  - important for recall (and precision)
  - more important, often, than retrieval algorithm
- MapReduce
  - Reverse index, corpus term frequencies, idf

EMC^2 PROVEN PROFESSIONAL

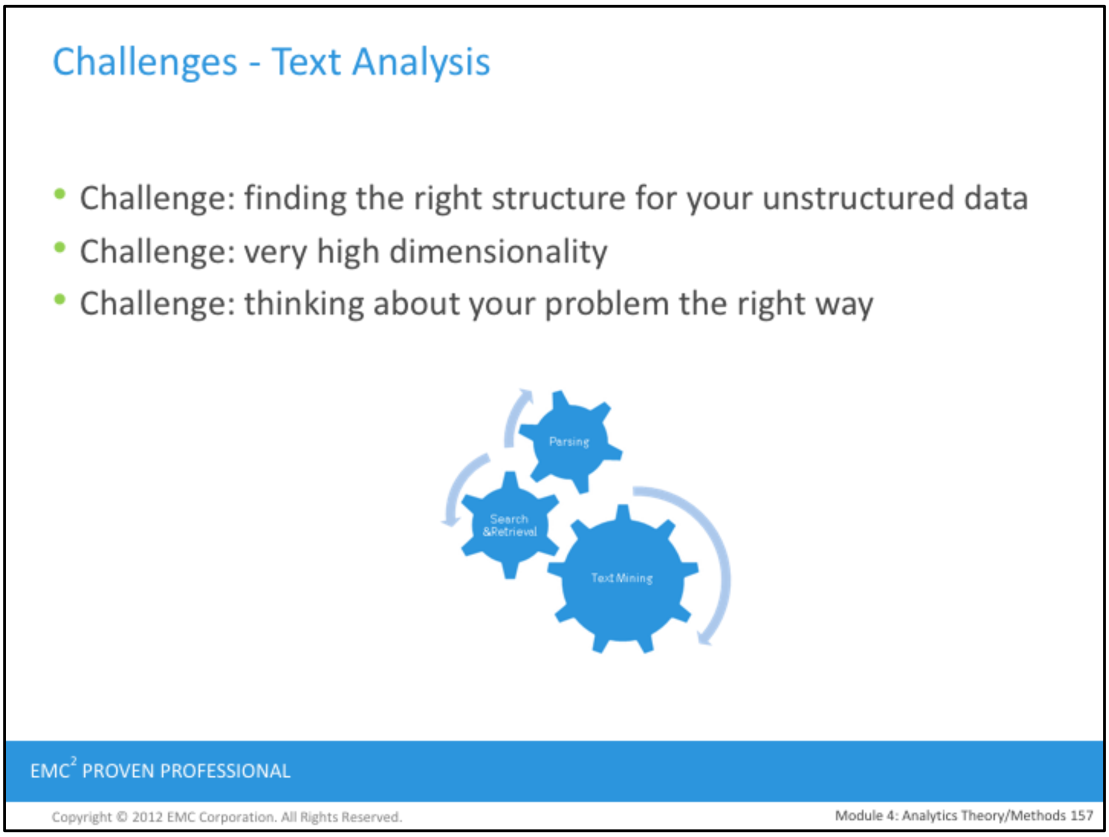
Module 4: Analytics Theory/Methods 156

There are other retrieval algorithms, probably more effective than the basic one that we described. But the important thing is that the documents be available for search.

The relevance metric is important for the precision and user experience. Crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages to provide fast searches.

The search engineers who provide the infrastructure for the search and retrieval process play a key role in “text analysis”. More so than played by Data Scientists.

The tasks such as reverse indexing, finding the idfs and corpus term frequencies are implemented effectively with map and reduce algorithms that we will detail in Module 5.



Challenges - Text Analysis

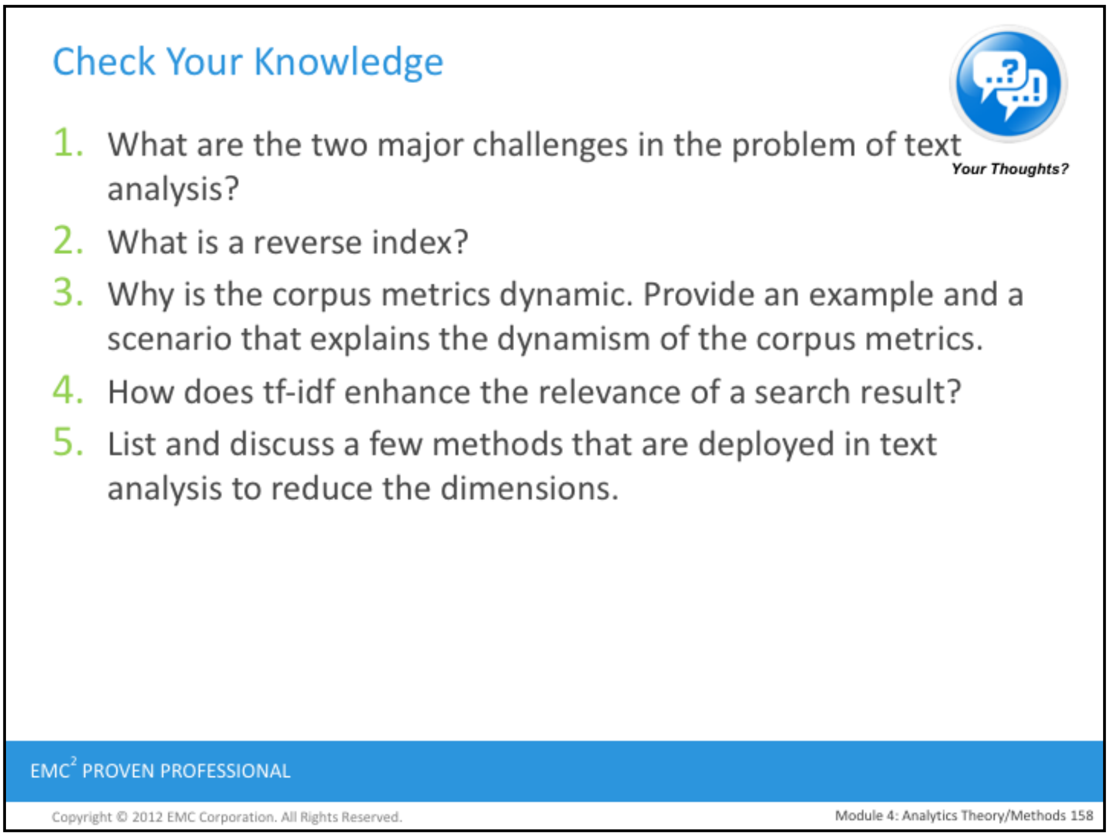
- Challenge: finding the right structure for your unstructured data
- Challenge: very high dimensionality
- Challenge: thinking about your problem the right way

EMC² PROVEN PROFESSIONAL

Module 4: Analytics Theory/Methods 157

We again recap on the key challenges with text analysis.

As we saw in Module 2, the most challenging aspect of data analytics problems often isn't the statistics or mathematical algorithms; it's formulating the problem, getting the data, and preparing the data. This is especially true for text analysis.



Check Your Knowledge

Your Thoughts?

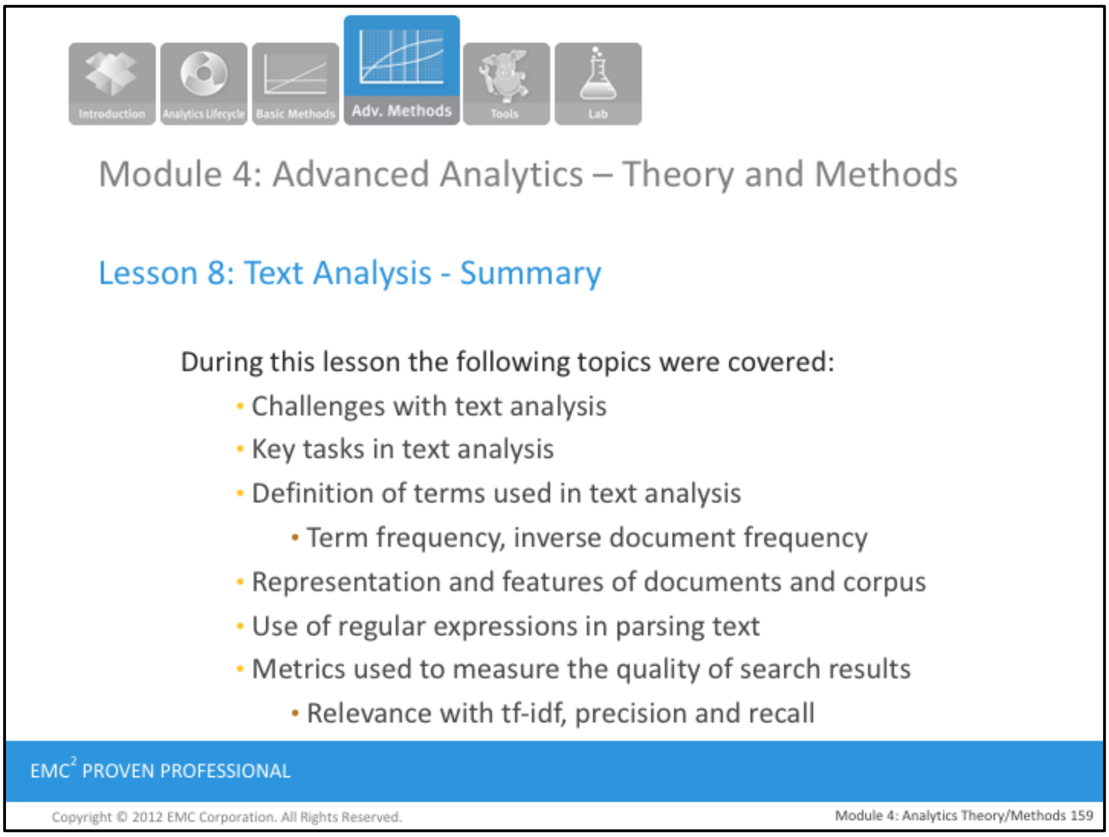
1. What are the two major challenges in the problem of text analysis?
2. What is a reverse index?
3. Why is the corpus metrics dynamic. Provide an example and a scenario that explains the dynamism of the corpus metrics.
4. How does tf-idf enhance the relevance of a search result?
5. List and discuss a few methods that are deployed in text analysis to reduce the dimensions.

EMC^2 PROVEN PROFESSIONAL

Copyright © 2012 EMC Corporation. All Rights Reserved.

Module 4: Analytics Theory/Methods 158

Record your answers here.


Introduction
Analytics Lifecycle
Basic Methods
Adv. Methods
Tools
Lab

Module 4: Advanced Analytics – Theory and Methods

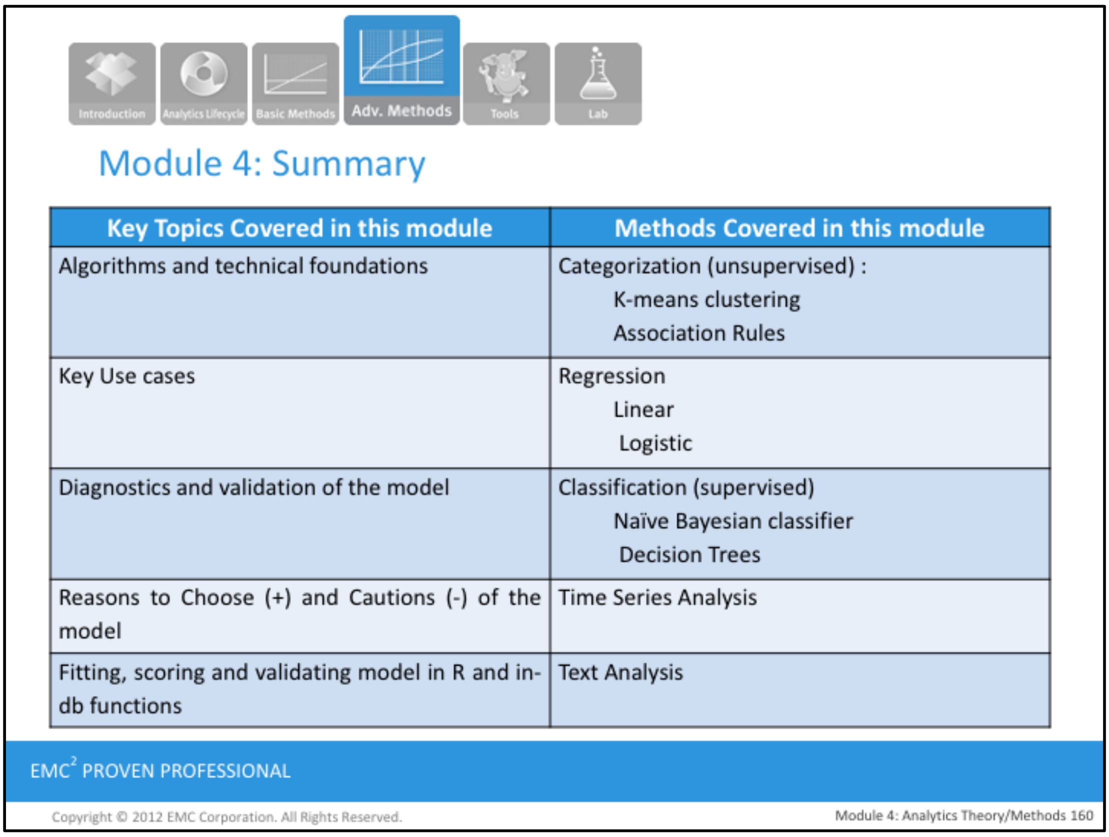
Lesson 8: Text Analysis - Summary

During this lesson the following topics were covered:

- Challenges with text analysis
- Key tasks in text analysis
- Definition of terms used in text analysis
  - Term frequency, inverse document frequency
- Representation and features of documents and corpus
- Use of regular expressions in parsing text
- Metrics used to measure the quality of search results
  - Relevance with tf-idf, precision and recall

EMC^2 PROVEN PROFESSIONAL

This lesson covered these topics. Please take a moment to review them.


Introduction
Analytics Lifecycle
Basic Methods
Adv. Methods
Tools
Lab

Module 4: Summary

Key Topics Covered in this module	Methods Covered in this module
Algorithms and technical foundations	Categorization (unsupervised) : K-means clustering Association Rules
Key Use cases	Regression Linear Logistic
Diagnostics and validation of the model	Classification (supervised) Naïve Bayesian classifier Decision Trees
Reasons to Choose (+) and Cautions (-) of the model	Time Series Analysis
Fitting, scoring and validating model in R and in-db functions	Text Analysis

EMC^2 PROVEN PROFESSIONAL

Module 4: Analytics Theory/Methods 160

Summary of key-topics presented in this Module are listed.