What Ties to What?
Visualizing Customer Text Data with a Bipartite Graph

Mark Turner
Text Analytics Lead
Teradata Advanced Analytics
What Ties to What?

• “Which stores are mentioned most often when customers complain about delays?”
• “Are customer complaints mentioning an emerging safety issue with a part in one of our products?”
• “In click-to-chat dialogues, which brand do customers mention when they ask about one of our product lines?”
• . . . and many more

• All of these are questions about association.
Why Isn’t it Simple?

• So what’s wrong with just finding out which words occur together?

• Real text data is more complicated:
  • It’s noisy: full of typos, abbreviations, misspellings, differences in transliteration, OCR errors, and more.
  • Individual words are not enough: we need names and phrases.
    • New York ≠ New + York
  • We may be interested in classes of words or phrases. Example: types of products, categories of customer issues.
  • We are often interested in sentiment.
Steps

• Major steps:
  1. Extracting information from text
  2. Removing noise
  3. Categorizing into classes
  4. Finding and visualizing associations

• Not all of the steps apply to all data and business problems.
## Extracting Information from Text

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>nGram</strong></td>
<td>Find consecutive strings of words in text</td>
</tr>
<tr>
<td><strong>TextParser</strong></td>
<td>Find each word, its position, and frequency.</td>
</tr>
<tr>
<td><strong>TF-IDF</strong></td>
<td>Finds the relative importance of each word in each document.</td>
</tr>
<tr>
<td><strong>Named Entity Recognition</strong></td>
<td>Find persons, places, dates, times, model numbers, etc.</td>
</tr>
<tr>
<td><strong>PoSTagger</strong></td>
<td>Identify each word as a noun, verb, etc.</td>
</tr>
<tr>
<td><strong>TextChunker</strong></td>
<td>Find syntactic units (noun phrases, verb phrases, etc.)</td>
</tr>
<tr>
<td><strong>Sentenizer</strong></td>
<td>Extracts sentences</td>
</tr>
<tr>
<td><strong>TextMorph</strong></td>
<td>Finds lemmas of words.</td>
</tr>
<tr>
<td><strong>TextTokenizer</strong></td>
<td>Finds words in Chinese text.</td>
</tr>
</tbody>
</table>
Extracting Information from Text: N-Grams

• A method of “atomizing the text”.
• Captures virtually **all** names and phrases without a dictionary, thesaurus, or ontology.
• Example input:

<table>
<thead>
<tr>
<th>ID</th>
<th>Src</th>
<th>Txt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Wikipedia</td>
<td>the quick brown fox jumps over the lazy dog</td>
</tr>
<tr>
<td>2</td>
<td>Sampledoc</td>
<td>hello world. again, I say hello world</td>
</tr>
</tbody>
</table>

• Example output (n=2):

<table>
<thead>
<tr>
<th>ID</th>
<th>SRC</th>
<th>Ngram</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Wikipedia</td>
<td>the quick</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>Wikipedia</td>
<td>quick brown</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>Wikipedia</td>
<td>brown fox</td>
<td>1</td>
</tr>
</tbody>
</table>
Extracting Information from Text: N-Grams

- Generally, we use n-grams as **pre-processing** for other text analysis functions such as:
  - finding associations among terms,
  - finding time patterns ("what's trending?")
  - clustering documents,
- to aim for meanings that are **not carried by individual words**.
- **High recall**: this method captures virtually all words, names, and phrases without a dictionary, taxonomy, or ontology
- A **scalable platform** is important:
  - Output is often $\geq 100\text{M}$ records.
  - Most of the information is in a small portion.
Extracting Information from Text: Named-Entity Recognition

• What is it?
  • Finding names of people, organizations, places, products, parts, genes, dates, times, dollar amounts, etc.
  • These are *high-value* terms in text data.

• Why do we need a function to find them?
  • Words can be ordinary nouns ("cook") or proper nouns ("Ralph Cook").
  • They are often multi-word: "New York", "Ralph Cook", "Gap"
  • Titles and upper/lowercase aren’t always used.
    • "**COOK** ISSUED THE REPORT." → last name?
    • "**COOK** THE CASSEROLE AT 350" → verb
    • "PLEASE FIND THE **COOK**" → noun
Finding Information from Text: Named Entity Recognition

• Example input:

<table>
<thead>
<tr>
<th>ID</th>
<th>Src</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>wiki</td>
<td>U. S. President Barack Obama has arrived in South Korea, where he is expected to show solidarity with the president in demanding North Korea move toward ending its nuclear weapons programs.</td>
</tr>
<tr>
<td>2</td>
<td>wiki</td>
<td>Please send me email via <a href="mailto:john@teradata.com">john@teradata.com</a>.</td>
</tr>
</tbody>
</table>

• Example output:

<table>
<thead>
<tr>
<th>ID</th>
<th>Src</th>
<th>Entity</th>
<th>Type</th>
<th>Start</th>
<th>End</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Wiki</td>
<td>Barack Obama</td>
<td>Person</td>
<td>18</td>
<td>20</td>
<td>…S . President Barack Obama has arrived in…</td>
</tr>
<tr>
<td>1</td>
<td>Wiki</td>
<td>U. S.</td>
<td>Location</td>
<td>0</td>
<td>7</td>
<td>U.S. President Barack Obama…</td>
</tr>
<tr>
<td>1</td>
<td>Wiki</td>
<td>South Korea</td>
<td>Location</td>
<td>46</td>
<td>57</td>
<td>… has arrived in South Korea, where he is…</td>
</tr>
</tbody>
</table>
Named-Entity Recognition

• Methods of finding named entities:
  • List (‘lexicon’)
  • Patterns (regular expressions)
  • Statistically models (use built-in model, or train with your own ground truth)

• User can mix and match, i.e. use different models for each type of named entity.

• To achieve high recall (catch every occurrence), an ensemble method may help.

• Experience is (very) helpful in selecting the right methods for the data.
Sentiment Analysis

• Sentiment analysis looks for the writer’s emotions: positive, negative, or neutral.
• Operates at the sentence or document level.
• Example:
  “I live in Mongolia and bought Kindle Fire. Now it turns out that "due to my geographical location" I can not purchase/ download a single application, game, movie... nothing. Nada. The only thing I can do is download and read books. I am disappointed.”

Output

<table>
<thead>
<tr>
<th>Out_content</th>
<th>Out_feature</th>
<th>Out_polarity</th>
<th>Out_strength</th>
<th>Out_sentiment_words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document</td>
<td>Neg</td>
<td></td>
<td>2</td>
<td>disappointed</td>
</tr>
</tbody>
</table>
Sentiment Analysis: How is it Done?

• Dictionary approach:
  • Each word has a default sentiment polarity and strength.
  • We get overall sentiment and strength by combining the individual word scores.
  • Need rules for negative terms ("no", "not", etc.).

• Statistical approach:
  • Treats sentiment extraction as a classification problem.
  • Can be trained with your own data, or
  • Use an out-of-the-box model
Extracting Information from Text: Summary

- Each text extraction technique contributes insights into the text.

  This bicycle was generally okay. There were real problems with the delivery time and setup; I wouldn’t trust them the next time I shop for something like this. Maybe Acme does better.
Another Issue: Noise

• What is noise?
• In text data, noise is variations from:
  • typos
  • spelling mistakes
  • abbreviations
  • OCR output
  • transliteration
• All of these interfere with accurately tracking words, phrases, and names over time.
Transliterations

- Representing a word or name in one language in the writing system of another.
- For many languages (Arabic, Russian, others), there is no standard way to do this.
- This problem comes up often with names of people and locations.

معمر محمد أبو منيار القذافي

- Qaddafi, Muammar
- Al-Qadhafi, Muammar
- al-Qadhafi, Muammar
- Al Qadhafi, Mu’ammar
- Al Qadhafi, Muammar
- El Qaddafi, Moammar
- El Kadafi, Moammar
- El Kaddafi, Moammer
- El Qadhafi, Mu’Ammar
- Qadafi, Muammar
- Qaddafi, Moamar
- Qadafi, Moammar
- Qadhafi, Mo’Ammar
- Qadhafi, Muammar
- Al-Qadhafi, Moammar
- Al Qadhafi, Mu’ammar
- Al Qadhafi, Muammar
- Qaddafi, Muammar
- Qadafi, Muammar
- Qadafi, Mu’Ammar
- Qadhafi, Mu’ammar
- Qadafi, Mu’ammar
- Qadhafi, Mu’ammar
- Qadafi, Mu’ammar
- Qadhafi, Mu’ammar
- Qadafi, Mu’ammar
- Qadhafi, Mu’ammar
- Qadafi, Mu’ammar
- Qadhafi, Mu’ammar
- Qadafi, Mu’ammar
- Qadhafi, Mu’ammar
- Qadafi, Mu’ammar
- Qadhafi, Mu’ammar

Moamer Gaddafi
- Moamer Kadafi
- Moammar Gaddafi
- Moammar Gadafi
- Moamar Gadafi
- Moamar Ghaddafy
- Moammar Khaddafy
- Moammar el Gadhafi
- Moammer Gaddafi
- Moammer al Qaddafi
- Omar Gadafi
- Omar Kadafi
- Omar Mouammar Al Qaddafi
- Omar Muammar Al Qadafi
- Omar Muammad Al Qadafi
- Omar Muammar Gaddafi
- Omar al Ghaddafy

people and locations
Removing Noise by Matching

- Need to find *similarity* of strings in the data and a master list.
- This is a large-scale (m x n) matching problem.
  - Words: >= 5K
  - N-grams: >= 125M
  - Entities: (varies)
- Types of match:
  - exact
  - inexact (fuzzy)
What is Fuzzy Matching?

• Also called “approximate match” or “looks-alike matching”. Describes how similar two strings are.

• Example:

  fettucine
  fettuccine
  fettucini
  fetaccini
  fettaccine
  :    :  

  ➔ fettuccine
Fuzzy Match: Options

• Options for fuzzy match:

<table>
<thead>
<tr>
<th>Type</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>String Edit Distance</td>
<td>Levenshtein Distance</td>
</tr>
<tr>
<td></td>
<td>Damereau-Levenshtein Distance</td>
</tr>
<tr>
<td></td>
<td>Jaro</td>
</tr>
<tr>
<td></td>
<td>Jaro-Winkler</td>
</tr>
<tr>
<td>Global Alignment</td>
<td>Needleman-Wunsch</td>
</tr>
<tr>
<td>Set Comparison</td>
<td>Jaccard Distance</td>
</tr>
<tr>
<td>Vector</td>
<td>Cosine</td>
</tr>
</tbody>
</table>

• Which to use? Experience and understanding the data are key.
Finding Classes

• Motivation: find sequential patterns of classes, rather than specific symbols or words.

• Use a taxonomy of symbols and their class membership.

<table>
<thead>
<tr>
<th>Word</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>sherbet</td>
<td>frozen</td>
</tr>
<tr>
<td>peaches</td>
<td>produce</td>
</tr>
<tr>
<td>cornish game hen</td>
<td>poultry</td>
</tr>
<tr>
<td>ramen</td>
<td>soup</td>
</tr>
<tr>
<td>bleach</td>
<td>laundry</td>
</tr>
<tr>
<td>napkins</td>
<td>paper</td>
</tr>
<tr>
<td>fudgesicles</td>
<td>frozen</td>
</tr>
<tr>
<td>strawberries</td>
<td>produce</td>
</tr>
</tbody>
</table>
Finding Associations

• Starting point: how often do two words, phrases, names, n-grams, or classes occur together?
• Shopping cart analogy: “If there are chips in the market basket, how likely is it that there will also be salsa?”
• Document (review, tweet, email, etc.) = shopping cart
• Text units (word, phrase, name, n-gram, sentiment) = goods in the cart
How Do We Measure Association?

- Many measures are available:

<table>
<thead>
<tr>
<th>Measures for Associative Mining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Φ-coefficient</td>
</tr>
<tr>
<td>Yule’s Y</td>
</tr>
<tr>
<td>Gini index</td>
</tr>
<tr>
<td>Laplace</td>
</tr>
<tr>
<td>Jaccard</td>
</tr>
<tr>
<td>Chi-square</td>
</tr>
</tbody>
</table>

- Each provides a different insight or point of view.
- We use a **collaborative filter** to find the associations.
- The tool and platform should be scalable to the data: for n-grams in text, over 100M pairs can occur.
How Do We **Visualize** Association?

- Need a tool to visualize the associations found by the collaborative filter
- A useful technique is the *bipartite graph*.  
  - On one side: values of one category: brand, sentiment, etc.  
  - On the other side: terms (words, n-grams, named entities).  
  - The arcs (links) indicate *how strong* the association is, using color and/or thickness.  
  - The size of the nodes show the *frequency*.  

![Bipartite Graph Example](image)

- **Brand X**  
  - Fresh  
  - Cheap  
  - Packaging

- **Acme**  
  - Strong association
Combining the Elements

Text Data

Text Extraction

Fuzzy Match

Find Classes

Find Associations

Taxonomy

• Words
• Entities
• Phrases
• N-grams
• Sentiment

Standard Forms

Classes

Visualize
Case Study: What Do Customers Like and Dislike?

• Data Source
  – Product reviews for a popular moisturizer sold by a major TV and online retailer.

• Problem
  – The retailer wants to know what aspects of the product are getting positive and negative comments.

• Multi-Genre™ Approach
  – Combine text analytics with structured data.

• Value Generated
  – Finds specific issues and satisfaction for the product line, without surveys or focus groups.
Case Study: Approach

• Break text into *n-grams* (sets of consecutive words).

• Find associations between n-grams and sentiment (star ratings) with a *collaborative filter*.

• Do a *bipartite sigma graph* with one variable versus the other.
## Case Study: Reviewing the Data

<table>
<thead>
<tr>
<th>Positive</th>
<th>i've always had pretty good skin and i love love love this moisturizer!!! it isn't too heavy yet delivers moisture like you wouldn't believe. my skin is beautiful and flawless.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative</td>
<td>This is my 3rd jar of this cream and I am going to return it ,it smells like a nasty wet dog, all your products do not smell the way they used too, no perfume smell at all and I have just about all of them. . .</td>
</tr>
</tbody>
</table>
Summary

• Text data contains a wealth of information about customer reactions and preferences.

• Text extraction methods, combined with association techniques, brings out important differences between dimensions of the customer experience: brands, products, features, stores, sentiment, and many others.

• Experience in selecting the right methods for each data source will get the full value from the data.
Contact

Mark Turner: mark.turner@teradata.com