Machine Learning vs. Natural Language Processing

February 28, 2018 13 m, 2 s

What is Machine Learning?

Machine Learning in the context of text analytics is a set of statistical techniques for identifying parts of speech, entities, sentiment, and other aspects of text. The techniques can be expressed as a model that is then applied to other text, also known as supervised machine learning. It also could be a set of algorithms that work across large sets of data to extract meaning, which is known as unsupervised machine learning. It's important to understand the difference between supervised and unsupervised learning, and how you can get the best of both in one system.

Supervised Machine Learning

In supervised machine learning a bunch of documents are "tagged" for some feature found in speech. These documents are used to "train" the statistical model, which then is applied to new text. If you want a better or larger dataset to get improved results you can retrain the model as it "learns" more about the documents it analyzes. This supervised approach also applies to the sort of re-training that can happen with some models where some viewer gives a "star" rating – and the algorithm adds that rating to its ongoing processing.

There are many different model types you'll see mentioned often:

- Support Vector Machines
- Bayesian Networks
- Maximum Entropy
- Conditional Random Field
- Neural Networks/Deep Learning

All you really need to know if come across these terms is that they represent a set of machine learning algorithms that are guided along in some way.

Unsupervised Machine Learning

These are statistical techniques to tease meaning out of a collection of text without pre-training a model. Some are very easy to understand like "**clustering**" which just means grouping "like" documents together into sets called "clusters." These can be sorted based on importance using hierarchical clustering from the bottom-up or the top-down.

Another type of unsupervised learning is known as "**Latent Semantic Indexing**." This extracts important words/phrases that occur in conjunction with each other in the text. You would use this for faceted search or returning search results that aren't exactly the phrases used for searches. So, if the terms "manifold" and "exhaust" are closely related in lots of documents, if you search for "manifold," you'll get documents back that contain the word "exhaust" as well. There's also **Matrix Factorization** which is a technique that allows you to "factor" down a very large matrix into the combination of two smaller matrices, using what are called "latent factors". Latent factors are similarities between the items. Think about it like this – if you see the word "throw" in a sentence, that's probably going to be associated with the word "ball" than the word "mountain" in the phrase: "I threw the ball over the mountain." While humans have a natural ability to understand the factors that make something "throwable" a computer or program must be taught this difference. Matrix factorization came to prominence with the Netflix® challenge – where teams competed for a million-dollar prize to increase the recommendation ratings accuracy by 10%.

Natural Language Processing (NLP)

Natural Language Processing broadly refers to the study and development of computer systems that can interpret speech and text as humans naturally speak and type it. Human communication is frustratingly vague at times; we all use colloquialisms, abbreviations, and don't often bother to correct misspellings. These inconsistencies make computer analysis of natural language difficult at best, but in the last decade NLP and machine learning has progressed immeasurably. At Lexalytics, we're focused on the text side of things; and we've been developing and innovating since 2004.

There are three aspects of a given chunk of text, each of which must be understood:

Semantic Information

Semantic information is the specific meaning of an individual word. A phrase like "the bat flew through the air" can have multiple meanings depending on the definition of bat: winged mammal, wooden stick, or something else entirely? Knowing the relevant definition is vital for understanding the meaning of a sentence.

Another example: "Billy hit the ball over the house." As the reader, you may assume that the ball in question is a baseball, but how do you know? The ball could be a volleyball, a tennis ball, or even a bocce ball. We assume baseball because they are the balls most often "hit" in such a way, but without machine learning of natural language a computer wouldn't.

Syntax Information

The second key component of text is sentence or phrase structure, known as syntax information. Take the sentence, "Sarah joined the group already with some search experience." Who exactly has the search experience here? Sarah, or the group? Depending on how you read it, the sentence has very different meaning with respect to Sarah's abilities.

Context Information

Finally, you must understand the context that a word, phrase, or sentence appears in. What is the concept being discussed? If a person says that something is "sick", are they talking about healthcare or video games? The implication of "sick" is often positive when mentioned in a context of gaming, but almost always negative when discussing healthcare.

Putting it all together

Let's return to the sentence, "Billy hit the ball over the house." Taken separately, the three types of information would return results that run along the lines of:

- Semantic information: *person – act of striking an object with another object – spherical play item – place people live*
- Syntax information: *subject – action – direct object – indirect object*
- Context information: *this sentence is about a child playing with a ball*

These aren't very helpful by themselves. They indicate a vague idea of what the sentence is about, but full understanding requires the successful combination of all three components.

This analysis can be accomplished in a number of ways, through machine learning models or by inputting rules for a computer to follow when analyzing text. Alone, however, these methods don't work so well. Machine learning models are great at recognizing entities and overall sentiment for a document, but they struggle to extract themes and topics; what's more, they're less-than-adept at referring sentiment back to individual entities or themes.

Alternatively, you can teach your system to identify rules and patterns basic rules and patterns laid down by the language the text is written in. In many languages for example, a proper noun followed by the word "street" probably denotes the name of a street; similarly, a number followed by a proper noun followed by the word "street" is probably a street address. People's names usually follow generalized two- or three-word formulas of proper nouns and nouns.

Recording and implementing these takes an exorbitant amount of time, and you must be painstaking in your definitions. What's more, rules and patterns cannot possibly keep up with the evolution of language: the Internet has absolutely butchered traditional conventions of the English language, and no set of rules can possibly encompass every inconsistency and new language trend that pops up in your input text. Any nlp in machine learning must be constantly learning English.

Very early text mining systems were entirely based on rules and patterns. On the other side, as natural language processing and machine learning techniques have evolved over the last decade an increasing number of companies have popped up offering software that relies exclusively on machine learning methods. As explained just above, these systems can only offer limited insight.

That's why at Lexalytics, we utilize a variety of supervised and unsupervised models that work in tandem with a number of rules and patterns we've spent years refining. By taking this hybrid approach, our systems are infinitely customizable to return the exact level of detail our customers desire.

How Lexalytics uses Machine Learning:

Lexalytics uses a combination of supervised and unsupervised machine learning.

Supervised Learning

Tokenization

We use supervised machine learning for a number of natural language processing tasks such as "tokenization" which means defining what a "word" is in NLP machine learning algorithms. English is really easy – see all this white space between the letters and paragraphs? That makes it really easy to tokenize. So, we just use a simple set of rules for English tokenization.

But what if you're not working with English speaking input? For example, Mandarin Chinese has no whitespace, so machine learning is important for tokenization. This is what the input would look like for that: 中国有没有空格，所以机器学习是符号化的重要。Mandarin Chinese follows rules and patterns just like English, and with NLP and ML you can train a model to identify them

Parts of Speech Tagging

We use Parts of Speech Tagging for a number of important Natural Language Processing tasks. We need to know POS to recognize Named Entities, to extract themes, and to process sentiment. So, Lexalytics has a highly robust model for doing parts of speech tagging with >90% accuracy, even for short, gnarly social media posts.

Named Entity Recognition

We use a machine learning model that we've trained on large amounts of content to recognize People, Places, Companies, and Product entities. It is important to note that the Named Entity Recognition model requires Parts of Speech as an input feature, so, this model is reliant on the Part of Speech tagging model.

Sentiment

We also have another machine learning model that can be trained by our customers to recognize entity types that we don't include in our pre-trained model (e.g. "trees" or "types of cancer"). We've built a number of **sentiment classification models** for different languages.

Classification

Our customers can come to us with a set of pre-tagged content. If they've been hand-analyzing surveys for a while, they have a bunch of categories with associated tagged content. We help them train up a classification model that exactly matches how they've been scoring content. It's essentially the same process they were doing before, just much easier and faster.

How Lexalytics uses unsupervised learning:

Lexalytics uses unsupervised learning to produce some "basic understanding" of how language works. In order to interpret the meaning of a set of words, you need three things: semantics, syntax, and context. We can extract certain important patterns with large enough corpora of text to help us make the most likely interpretation.

Concept Matrix™:

The Concept Matrix™ is unsupervised learning done across Wikipedia™ – it allows us to understand things like "apple" is close to "fruit" is close to "tree" but is far away from "lion", but is closer to "lion" than it is to, say, "linear algebra." It forms the base of our semantic understanding

Syntax Matrix™

The Syntax Matrix™ is unsupervised learning done on a massive corpus of content (many billions of sentences) using the aforementioned matrix factorization technique. It helps us understand the most likely parsing of a sentence – forming the base of our understanding of syntax.

In a sentence

In a sentence we work to develop our natural language processing capabilities by utilizing machine learning techniques that work in tandem with traditional rules and patterns to work through a series of low-, mid-, and high-level text functions that deduce the semantic, syntax, and context information found in any block of unstructured input text.

This analysis can be accomplished in a number of ways, through machine learning models or by inputting rules for a computer to follow when analyzing text. Alone, however, these methods don't work so well. Machine learning models are great at recognizing entities and overall sentiment for a document, but they struggle to extract themes and topics; what's more, they're less-than-adept at referring sentiment back to individual entities or themes.

Alternatively, you can teach your system a number of rules and patterns to identify. Much of text language follows some basic rules and patterns laid down by the language the text is written in. In many languages for example, a proper noun followed by the word "street" is probably denoting the name of a street; similarly, a number followed by a proper noun followed by the word "street" is probably a street address (versus an email address, you can see how syntax information is important). People's names usually follow generalized two- or three-word formulas of proper nouns and nouns.

But recording and implementing these rules and patterns takes an exorbitant amount of time, and you must be painstaking in your definitions. What's more, rules and patterns cannot possibly keep up with the evolution of language: the Internet has absolutely butchered traditional conventions of the English language, and no set of rules can possibly encompass every inconsistency and new language trend that pops up in your input text.

Very early text mining systems were entirely based on rules and patterns. On the other side, as natural language processing and machine learning techniques have evolved over the last decade an increasing number of companies have popped up offering software that relies exclusively on machine learning methods. As explained just above, these systems can only offer limited insight.

That's why at Lexalytics, our data scientists utilize a variety of supervised and unsupervised models that work in tandem with a number of rules and patterns we've spent years refining. By taking this hybrid approach, our systems are infinitely customizable to return the exact level of detail our customers desire.

So, here's how that hybrid system works.

Our text analysis functions are based on patterns and rules. Each time we add a new language to our capabilities, we begin by inputting the patterns and rules that language traditionally follows. Then our supervised and unsupervised machine learning models keep those rules in mind when developing their classifiers. We apply variations on this system for low-, mid-, and high-level text functions.

Low-level text functions are the initial processes through which you run any text input. These functions are the first step in turning unstructured text into structured data; thus, these low-level functions form the base layer of information from which our mid-level functions draw on. Those mid-level text functions involve extracting the real content of a document of text, determining who is speaking, what they are saying, and what they are talking about. The high-level function of sentiment analysis is the last step, determining and applying sentiment on the entity, theme, and document levels.

Low-Level:

- **Tokenization**: ML + Rules
- **PoS Tagging**: Machine Learning
- **Chunking**: Rules
- **Sentence Boundaries**: ML + Rules
- **Syntax Analysis**: ML + Rules

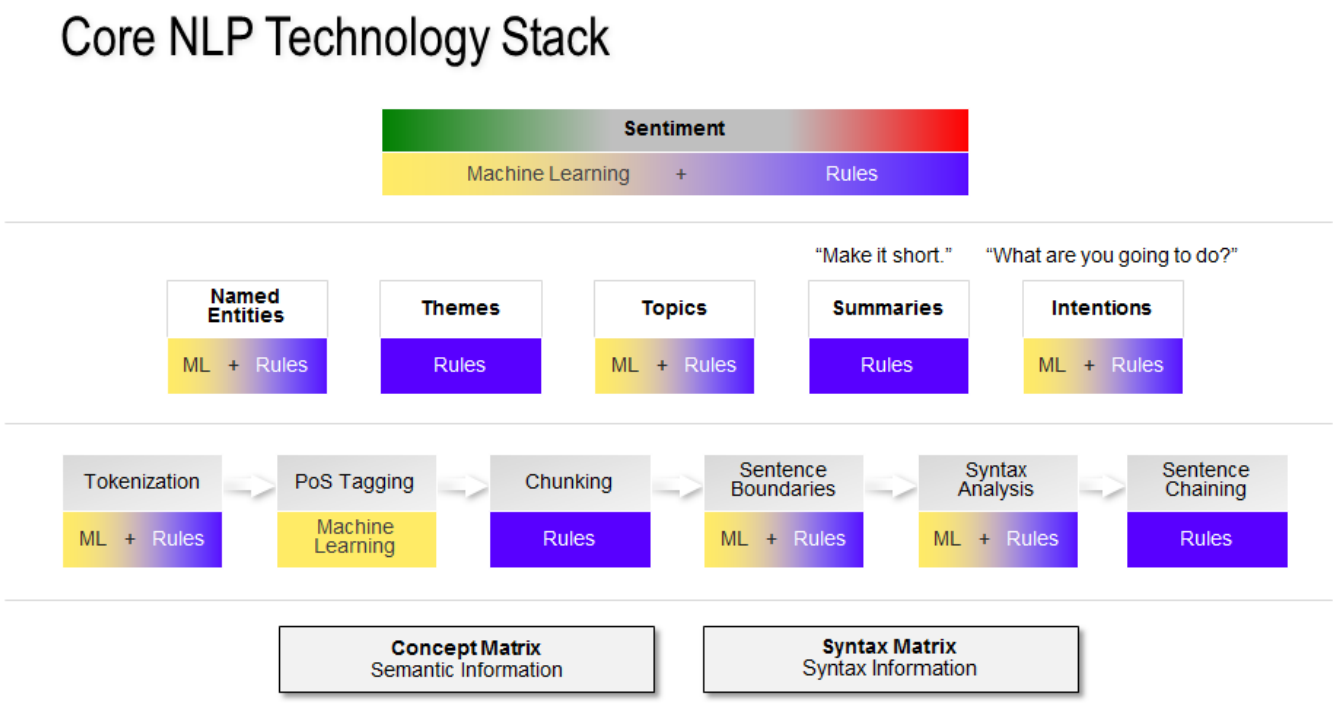
Mid-Level:

- **Entities**: ML + Rules to determine "Who, What, Where"
- **Themes**: Rules "What's the buzz?"
- **Topics**: ML + Rules "About this?"
- **Summaries**: Rules "Make it short"
- **Intentions**: ML + Rules "What are you going to do?"
  - Intentions uses the syntax matrix to extract the intender, intendee, and intent
  - We use ML to train models for the different types of intent
  - We use rules to whitelist or blacklist certain words
  - Multilayered approach to get you the best accuracy

High-Level:

- **Apply Sentiment**: ML + Rules "How do you feel about that?"

You can see how this system pans out in the flowchart below:



(https://lexalytics.com/lexablog/wp-content/uploads/2012/02/technology-stack1.png)

We've optimized each of these functions to return the most accurate, most reliable results. Through adoption of a hybrid approach to text analytics, utilizing machine learning models in tandem with preset rules and patterns, our text mining software is fluent in natural language of all types: even emojis, smiley faces, and acronyms. Our text analytics solutions provide more and better contextual information than any other offering on the market, and through innovative technologies like the Concept Matrix and Syntax Matrix we continue to lead the way in text analytics development.

Still hungry for more information? Try the demo (https://www.lexalytics.com/demo), download a free trial (https://www.lexalytics.com/support/apps/excel), check out our whitepapers (https://www.lexalytics.com/resources) on each category of text function, or see the technical information (https://www.lexalytics.com/technology) pages of our website.

Categories: Natural Language Processing (https://www.lexalytics.com/lexablog/category/natural-language-processing)

One Response to "Machine Learning vs. Natural Language Processing"

1. *Jaime (https://www.ziptask.com)* May 19th, 2016 (https://www.lexalytics.com/lexablog/machine-learning-vs-natural-language-processing-part-1#comment-19968)

   Great post. Check this one out as well. Natural Language Processing with Python https://www.ziptask.com/Natural-Language-Processing-with-Python (https://www.ziptask.com/Natural-Language-Processing-with-Python)

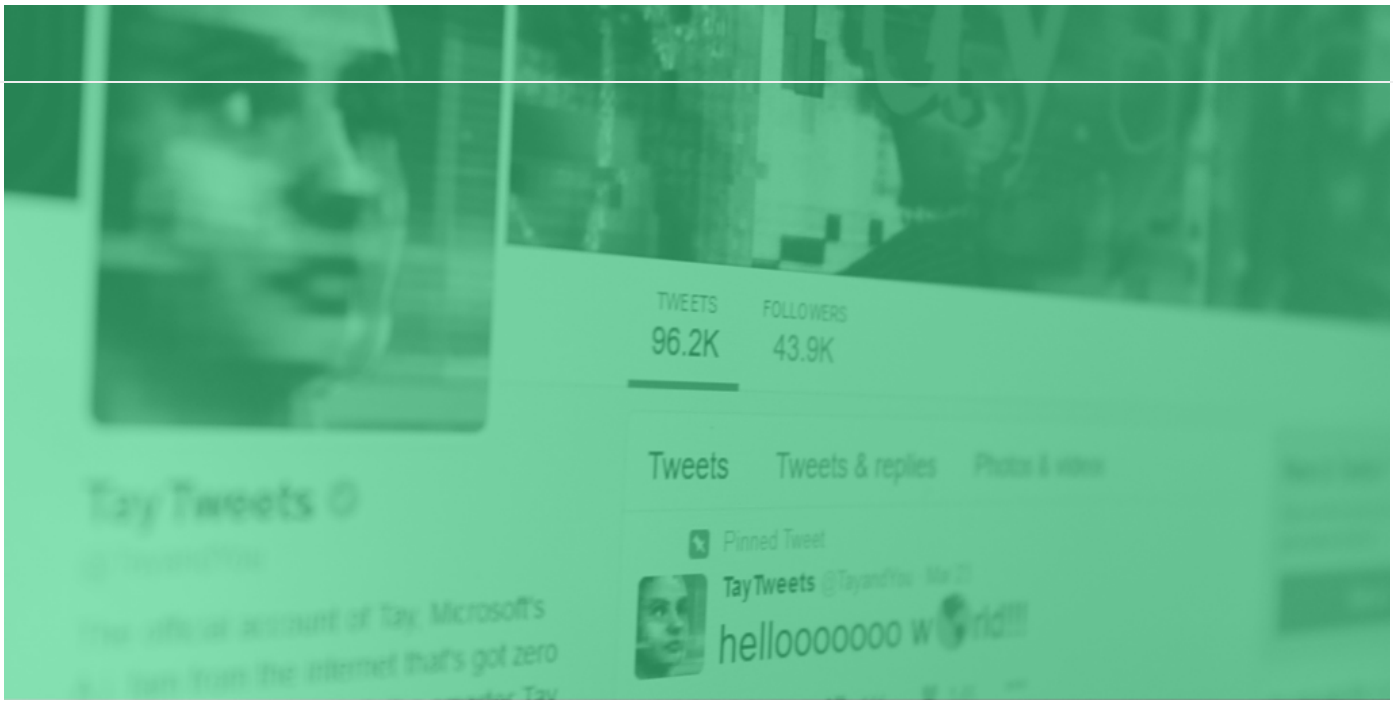
Seth Redmore

(http://twitter.com/sredmore) (http://www.linkedin.com/in/sethredmore)

Seth was into data before it was big and has been dealing with unstructured data problems for over ten years now.

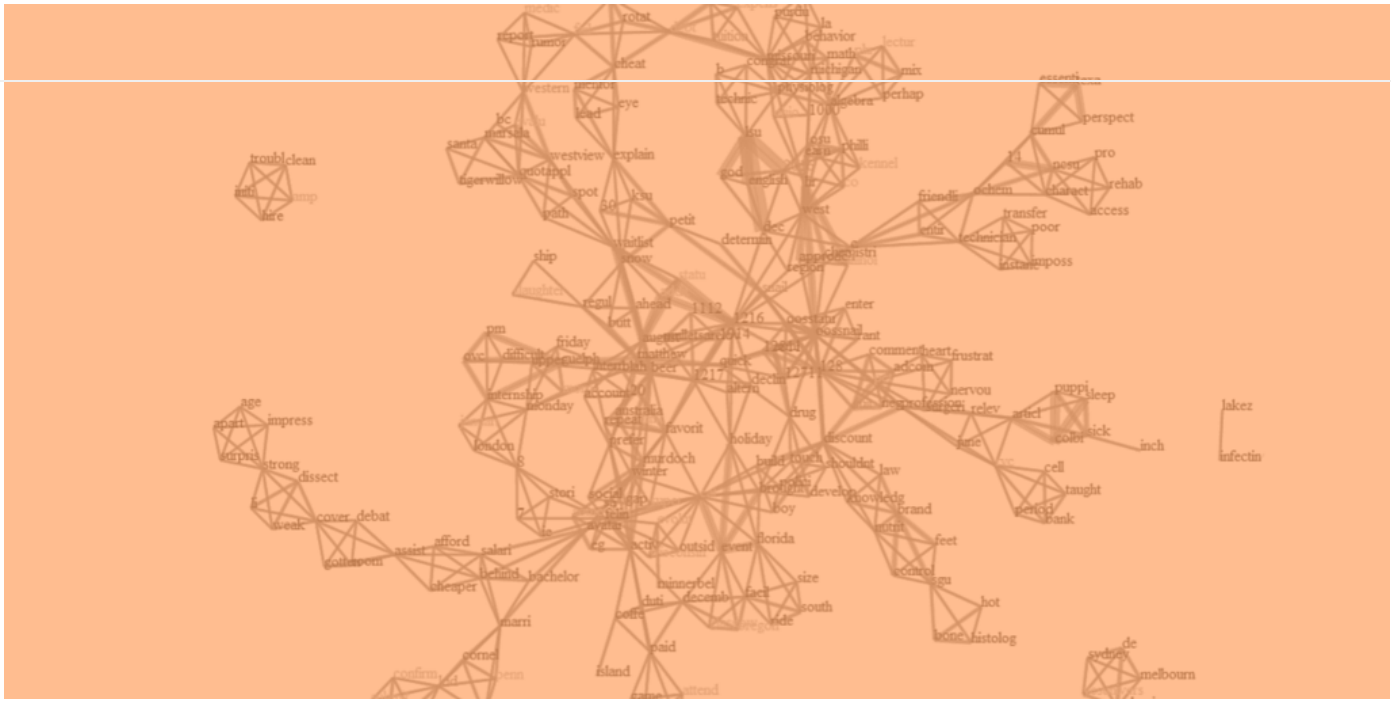
Related articles

Exploratory Data Analysis in 1 Minute (https://www.lexalytics.com/lexablog/exploratory-data-analysis)



Don't Crash and Burn with Your Bot (https://www.lexalytics.com/lexablog/chatbot)

Text Analytics Reduces Churn (https://www.lexalytics.com/lexablog/voziq)



Natural Language Processing in 5 Minutes (https://www.lexalytics.com/lexablog/what-is-natural-language-processing)

Get a FREE trial (/signup) Schedule full demo (/contact?action=demo)

I just want a taste

Enter a URL... Try it!

try the demo (/demo) with a URL or plain text for a bit of what we do

Or call us at **1-800-377-8036**

Try our demo ()

Or call us at **1-800-377-8036**

Products & Services

AI Assembler (/assembler)
Salience On-Premise (/salience/server)
Semantria Cloud API (/semantria)
Semantria for Excel (/semantria/excel)
Salience Mobile (/salience/mobile)
Semantria Storage and Visualization (beta) (/storage)
Professional Services (/services)

Resources

Demo (/demo)
Blog (/lexablog)
Technology (/technology/text-analytics)
Case Studies & Papers (/resources)
Industries (/industries)
Applications (/applications)

Support

Support home (/support/)
Tuning guides (/support/tuning/)
Semantria for Excel (/support/apps/excel)
API integration (https://semantria.readme.io/docs)
Salience Wiki (http://dev.lexalytics.com/wiki/)

Company

About (/about)
Company News (/news/company-news/)
Press Releases (/news/press-releases/)
Privacy Policy (/privacy-policy)
Security Overview (/security-overview)
Contact us (/contact)

(https://www.facebook.com/pages/Lexalytics/124936814233570)
(http://www.twitter.com/lexalytics)
(https://www.youtube.com/user/Lexalytics)
(https://www.quora.com/topic/Lexalytics)
(https://www.linkedin.com/company/lexalytics-inc-)

(https://www.facebook.com/pages/Lexalytics/124936814233570)

(http://www.twitter.com/lexalytics)

(https://www.youtube.com/user/Lexalytics)

(https://www.quora.com/topic/Lexalytics)

(https://www.linkedin.com/company/lexalytics-inc-)

Feedback