Tune First, Then Train
Introduction

Reviews are highly context dependent. Two business owners may take the exact same sentence and ascribe a different sentiment to it. Take the review, “Food was slow to arrive.” A restaurant owner would read this very differently than the owner of a motel that doesn’t serve food. For one it’s crucial feedback. For the other, it’s something that falls beyond the remit of their business. This context dependence is why it’s essential that businesses using text analytics tune their solution to suit their particular needs.

However, all too often, businesses skip this critical step. From our experience, about half of our customers take our engine and run it without ever tuning it. Of the other half, about 50% will do an initial tuning run along with an occasional checkup. The remaining group will tune on a regular basis, always seeking the highest levels of precision and recall.

Over the past decade and a half, we’ve developed a particular philosophy around how to customize a text analytics system to better fit your business.

This philosophy boils down to the following two points:

1 Tuning first, then train
2 Train as small as you can, but as big as necessary

Now let’s take those rather gnomic statements and expand them.
**TUNE FIRST, THEN TRAIN**

Tuning is akin to telling a system what to do. When tuning, you write a few lines of code that the system then applies immediately. Training is more about convincing a system to do something. It involves feeding a dataset to a machine learning model for it to “learn” from. As the algorithm learns, it creates progressively better models.

Tuning lets you efficiently solve exactly the case that you specified – no more and no less. Training gives you the ability to handle novel cases that may not have shown up in the training data, and can be more flexible.

Most systems allow some tuning. Entering keywords is a basic example: when you type in a keyword you’re telling the system to return specific results. Lexalytics’ products let you tune all kinds of things. You can add new sentiment phrases or change the weighting of sentiment phrases. You can use patterns to define what will be considered a sentiment phrase. You can add entities, or define part-of-speech patterns for intentions and themes.

Of course, you can also train our systems. We’ve created custom machine learning models for many clients, and we’re constantly renewing our existing models. Our core text analytics engine, Salience, contains many machine learning models working in concert, any one of which can be updated to provide the best possible results for your data.
SO, WHY TUNE BEFORE TRAINING?

Tuning is immediate

When the Unicode 11.0 specification came out, we immediately supported the new Emoji set in both our sentiment engine and in our categorization systems. A pizza emoji, for example, was categorized as food. All we had to do was write the code, and our engine knew what to do. However, if training was the only option, we would have had to first gather real examples of usage to score with sentiment before being able to feed them to the system to learn on.

Tuning is precise

When training, the system doesn’t know exactly what you’re training for: if you mark up a document as negative, all of the words in the document are seen as contributors to the negativity of the document. This can have unforeseen consequences. Imagine you’re training sentiment on a financial news dataset during a bull market. Each document will contain everyday phrases like “4th quarter.” These words don’t actually carry sentiment; they’re simply markers of time. But if you don’t counterbalance these documents with some from a bear market, these everyday phrases will come up with a positive sentiment association.

Tuning is transparent

With tuning, input governs output. Putting in the keyword “Lexalytics” will bring up every mention of the word Lexalytics. Give the word “cool” a +0.5 sentiment score, and “cool” will receive a +0.5 sentiment score every time it appears. There’s strength in knowing exactly what’s going to happen.

Trouble is, terms like “cool” can be ambiguous, giving you output you didn’t anticipate. That’s where training comes in.
Train as small as you can, but as big as necessary

Tuning is immediate, precise, and transparent – but it isn’t flexible. It can only handle those situations it has been tuned to handle. In order to get the desired flexibility, you need to train the system on the general class of problems you need it to solve.

When training it’s important to be very clear about what you’re trying to get the system to do. Fortunately, you don’t need to train the system to do absolutely everything. For example, our text analytics system has an extensive sentiment dictionary that easily scores most everyday terms. All that’s left is to adjust the system so that it captures the specific words and phrases that your industry uses in an idiosyncratic way.

Take the term “gate change.” In everyday life this is a perfectly neutral term, but in the airline industry this phrase is negative. A gate change means a change of schedule, which often means delays – and disgruntled customers.

If you know these terms and how to score them, you can tune your system to capture them. For those that you can’t easily capture, there’s training. To achieve this, it’s most useful to train “micromodels” rather than training a large model across a bunch of varied content.
WHAT’S A MICROMODEL?

A micromodel is a model that’s responsible only for a single word or phrase.

Micromodels work best in situations where you’ve already used dictionaries or large models to achieve good results analyzing the easiest words and phrases.

If you’re starting from scratch, we recommend going big first unless it’s blindingly obvious that just a few small phrases are responsible for sentiment weight in your content.

Lexalytics uses machine learning models for many of our text analytics functions, including tokenization, part of speech tagging and language-specific grammar rules. We didn’t start out by scoring the really ambiguous things. We took large chunks of content and marked it up, precisely and tediously, over time. We created millions of annotations that we’ve continued to update over the past decade and a half. In all, we’ve spent tens of thousands of hours getting our “macromodel”-based machine learning systems into shape.

As a result, we have a system that works – but that still needs to be adjusted for your particular needs.

Take the company name Apple. For a human it’s simple enough to differentiate the company from the fruit, but for a machine it’s a non-trivial case.

Even though most data analytics companies have a general named entity recognizer that looks for “company” earmarks, many struggle to handle Apple well. This is particularly the case when the entity appears within short, poorly formed content, such as tweets.

But this confusion can be avoided by creating a micromodel whose sole job is to determine which instances of “apple” relate to the fruit and which relate to the company.
Micromodels have several advantages over training monolithic macromodels

Micromodels can be made arbitrarily precise.
You can keep pouring content into them, and since their focus is a single word or phrase, they will exceed the accuracy of even the best monolithic models.

Micromodels require less data than monolithic models
Because they’re designed to handle a single word or concept, they require less data to train than “macromodels.” Gathering and marking up training content is also a much simpler process, even for a non-technical user. Gather content using a simple keyword search, then annotate it, and finally train the model.

Micromodels have fewer unwanted side effects
Because you’re using very specific, relatively short data to train your model, there’s a much lower chance that you’ll create unintended consequences like training the model on an unintentional word.

Monolithic machine learning models, trained on large sets of data with general annotations, do have their place. But it’s smarter to think about the problem, see what terms are really important to you as a business, and then work to address those specifically.
SUMMARY / CONCLUSION

Modern text analytics systems are remarkably powerful right out of the box, but the extent of your competitive advantage is dependent on how good the results from your analytics systems are. For this reason, you should always do some customization of your system so that your results more precisely meet your business needs.

The aim is to do as little as possible, as specifically as possible, with as few far-reaching effects as you can manage. So, tune first, then train – and train as small as you can, but as big as necessary.

to explore how Lexalytics can help your business at lexalytics.com/contact
Lexalytics processes BILLIONS of unstructured documents every day, GLOBALLY.

We transform unstructured text into usable data and powerful stories.

Our on-premise Salience® engine, SaaS Semantria® API, and end-to-end Lexalytics Intelligence Platform® combine natural language processing with artificial intelligence to reveal context-rich patterns and insights within comments, reviews, surveys, and other text documents.

Data analytics and data analyst companies rely on Lexalytics to build better products, share insights between engineering, marketing, PR, and support teams, and drive business growth.

For more information, visit www.lexalytics.com or call 1-800-377-8036