thematic

UPDATED FEBUARY 2026

From Text Analytics To Generative AI: A Practical Guide for CX Teams

A review by Dr. Alyona Medelyan, PhD in Artificial Intelligence

Contents

Introduction

I've spent the last 25 years in Artificial Intelligence (AI), specifically in the area of making sense of text using algorithms: researching, creating, applying, and selling the technology behind it.

My academic research resulted in algorithms used by hundreds of organizations. I'm the author of early open-source thematic analysis algorithms KEA and Maui. While at Google, I wrote an algorithm that can analyze text in languages I don't speak.

Throughout my career, I've spoken with many Customer Experience (CX) and Insights practitioners who are living through the pain of analyzing unstructured feedback such as survey responses, call transcripts, reviews, or complaints.

In my 10 years as the CEO of customer intelligence platform Thematic, I've learned about what works in practice, and what doesn't. And most recently, I've seen the opportunities and challenges that Generative AI brought to this field. This perspective gives me a front-row seat to what's changing and what isn't.

For many years, text analytics was the common solution. The arrival of Generative AI is rapidly reshaping how teams analyze text. But interestingly, the pain remains when it comes to scaling, accuracy, governance, and actionability of the analysis.

In-house teams create impressive prototypes that never make it into production due to production constraints (cost, latency, security). General-purpose AI assistants are used informally by teams to summarize siloed feedback. Even when the output is plausible, this approach can miss cross-channel themes and create inconsistent or partial insights. Some established CX vendors are too slow to adopt AI and often rebrand pre-LLM approaches as "advanced AI". Their incentives favor services-heavy implementations.

In this guide, I'll outline how text analytics evolved, compare today's approaches (including LLM-based workflows), and highlight the practical risks such as cost, governance, and consistency that CX teams need to manage.

Finally, I'll explain how to move beyond text analytics toward Customer Intelligence, where feedback is analyzed consistently across channels and connected to outcomes.

Alyona Medelyan PhD
CEO and Co-Founder Thematic

Are you receiving more feedback than you could ever read, let alone summarize? Do you want to unlock insights from call center conversations?

Have you used a CX platform (Medallia's Text Analytics or Qualtrics' TextIQ or DiscoverXM), a DIY solution in Python or Claude, or a general-purpose AI assistant (ChatGPT, Copilot)?

Text analytics has evolved from simple techniques like word matching to crafting prompts for a Large Language Model.

What is text analytics?

Text analytics is the process of extracting meaning from text. For example, this can be analyzing survey responses, social media posts, call transcripts or employee feedback. The ultimate goal is to find common themes and trends to inform data-driven decisions and drive better business outcomes. Recently, Generative AI has changed how teams approach text analytics. Instead of building every rule or model from scratch, practitioners can use prompts to generate summaries, extract structured fields, and draft themes, often in minutes. The challenge is making those outputs reliable and repeatable in production.

How is text analytics used by companies?

At Thematic, for example, we analyze text submitted by customers and employees through various channels. Before bringing us onboard, teams analyze feedback using manual methods or setup-heavy tools. Modern text analytics approaches unify feedback across channels, discover consistent themes adapted to different use cases. Once this is done, CX teams can surface insights and easily answer questions in the feedback. They can track trends and issues, create visual reports, trigger workflows, and close the loop with the end customer. As a result, companies not only save time and resources, but also improve their prioritization and processes.

3 Text Analytics Approaches and Examples

Here is my summary to break down these methods into 3 key approaches that are commonly used today.

TEXT ANALYTICS APPROACH 1

Manual Rules

The manual rules approach has been one of the most popular text analytics approaches for many years. The main idea is to define patterns (simple or complex) that tag text into categories. The rules are often created using Boolean operators or regular expressions. Here is a rule for assigning the category **"Staff Knowledge"** from an enterprise platform such as Medallia:

"Staff knowledge"

i. knowledge

ii. knowledge NEAR where WITH 1 words

iii. knowledge CONNECTED TO products

The rule checks if the word "knowledge" is used in proximity of "where" and is mentioned in connection to "product" to catch questions or complaints about stuff knowledge on where to find products.

Manual rules are widely used by text analytics software and customer experience management providers who offer text analytics as part of a broader platform. Their interfaces often make it easy to create and manage such rules, and many also offer professional services to help with the creation and maintenance of taxonomies.

A key advantage of manual rules is that they are human-readable and auditable. They are explainable, and therefore can be tweaked and adjusted when needed. But the bottom line is that creating these rules takes a lot of effort. You also need to ensure that they are accurate and maintain them over time.

To get you started, some platforms ship with pre-packaged rules, already organized into a taxonomy. For example, they might have a category "Price", with hundreds of words and phrases already included, and underneath they might have sub-categories such as "Cheap" and "Expensive". They may also have industry-specific taxonomies (e.g., for banking). And if you are a bank, you can add your product names into this taxonomy and get started quickly.

The benefit of this approach is that once configured, you can process millions of feedback comments and get a good overview of the core categories mentioned in the text.

But there are plenty of limitations to this approach:

1. Multiple word meanings make it hard to create rules

One of the most common reason rules fail is **polysemy**: the same word can have different meanings in different contexts. That makes it hard to write a rule that works reliably. Here is an example of a rule for “Staff friendliness” and two examples (out of five) where it misclassifies feedback.

Friendly OR friendliness	Staff friendliness	
I was impressed by how **friendly** the person on the other end of the line was	Staff friendliness	✓
The lady who helped me was **friendly**	Staff friendliness	✓
Friendliness of staff	Staff friendliness	✓
Your website is very user **friendly**	Staff friendliness	✗
The young man on the phone was very **friendly**	Other	✗

2. Mentioned word ≠ core topic

Just because a word or a phrase is mentioned in a comment, it doesn’t necessarily mean that the comment is about that topic.

For example, a customer may provide context for a problem: “My credit card got declined and the cashier was super helpful, waiting patiently while I searched for cash in my bag.” This comment is not primarily about credit cards or cash, it’s about the behavior of the staff.

3. Rules cannot capture sentiment

Knowing the general category alone isn't enough.

How do people think about "Price", are they happy or not?

Capturing sentiment with handcrafted rules is extremely difficult to do reliably at scale. It's easy to underestimate how diverse and nuanced language is.

A pricing sub-category like "expensive" is a good example. Someone might say: "I did not think this product was expensive." A keyword rule will still match "expensive" and misclassify the sentiment. To correctly map this to something like "good price," you need more than a regular expression—you need to detect negation and its scope. A simple regex won't cut it.

4. Taxonomies don't exist for software products and many other businesses

The pre-set taxonomies with rules rarely exist out of the box for specialized products or services, or they don't transfer well. This is particularly problematic for the software industry, where each product has unique feature names, integrations, workflows and bugs. Teams either have to start from scratch or end up with overly generic categories that miss what matters.

5. Not everyone can maintain rules

In any industry, even if you have a working rule-based taxonomy, a skilled analyst/language-savvy expert would need to constantly maintain the rules to make sure all of the feedback is categorized consistently and accurately. This person would need to scan for new expressions that people create so easily on the fly, and for any emerging themes that weren't considered previously. It's a never-ending process that becomes expensive quickly.

And yet, despite these disadvantages and the advance of Generative AI, this approach is still the most widely used in commercial platforms. There is no simple fix, only incremental improvements.

So, are manual rules good enough for analyzing feedback?

My answer to this is **No**.

Most people who use manual rules are dissatisfied with the setup time and the maintenance costs. Manual rules only persist because they're explainable and familiar, but those benefits don't remove the core limitations

thematic

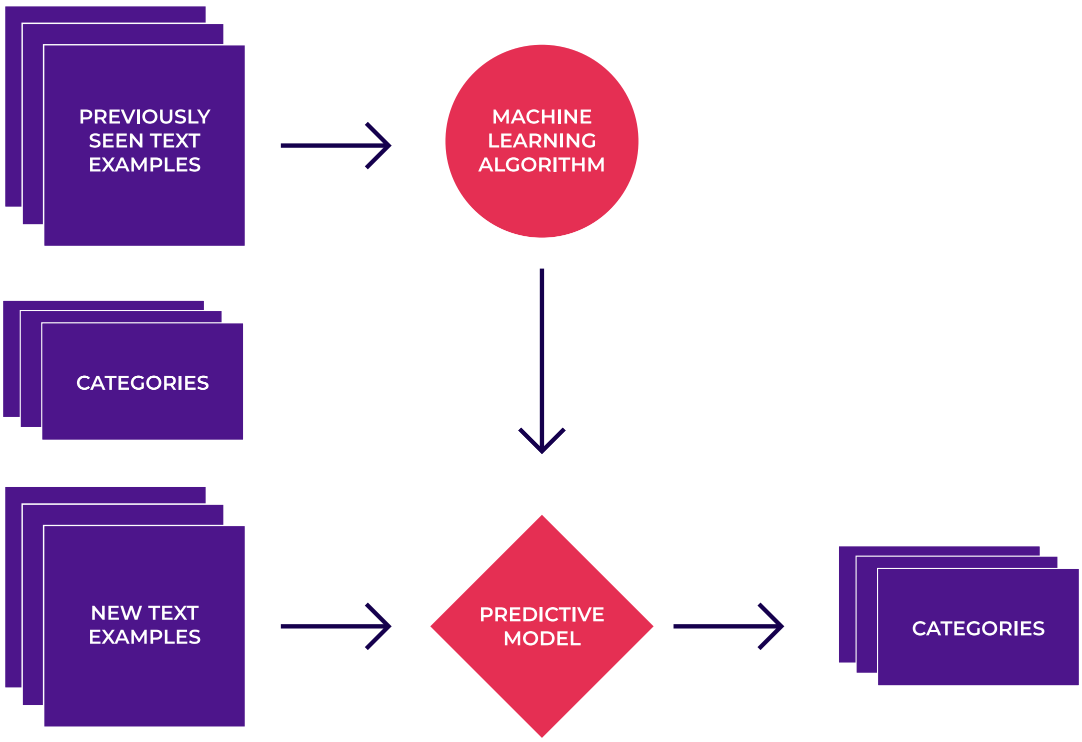
TEXT ANALYTICS APPROACH 2

Text Categorization

This is where text analytics became genuinely algorithmic. Text Categorization was the first widely adopted approach that moved beyond handcrafted rules by training models on labeled data. It has been widely used by data science teams, before Generative AI changed what's possible.

What is text categorization?

Text categorization is powered by machine learning. The basic idea is that an algorithm learns from previously labeled examples (training data) and then applies what it learned to categorize new feedback. Because it learns from labeled examples, it's called a **supervised** approach. In contrast, Large Language Models learn in an unsupervised way.



A key advantage of text categorization is that you can provide examples rather than manually creating patterns or rules.

Another advantage of text categorization is that, in theory, it can capture the relative importance of a word or phrase in context. Let's revisit the example from earlier. A customer may be explaining the situation that leads to an issue:

> "My credit card got declined and the cashier was super helpful, waiting patiently while I searched for cash in my bag."

This comment is not about credit cards or cash; it's about the behaviour of the staff. The mention of "credit card" isn't the main point, but "helpfulness" and "patience" are. With the right training data, a text categorization model can learn to focus on what matters.

It all comes down seeing similar examples in the training data.

Near perfect accuracy... but only with the right training data

Academic research has shown that text categorization can achieve very high accuracy in controlled settings. Data scientists can increase accuracy by changing the model or tweaking its parameters. And yet, all researchers agree that **the algorithm isn't as important as the training data.**

The quality and quantity of training data are the deciding factors in how successful this approach is for analyzing feedback. So, how much is enough? It depends on the number of categories and the method used to build the categorization model.

The more categories you have and the more closely related they are, the more training data is needed to help the algorithm to differentiate between them.

Some solutions that rely on text categorization provide tools that make it easier for teams to label examples and improve models over time.

But do you have time to wait for the model to improve, or do you need to act on customer feedback today?

Apart from needing to train the algorithm, here are four other problems with using text categorization for analyzing feedback:

1. You won't notice emerging themes

You will only learn insights about categories that you trained for, and will miss the **unknown unknowns.** This is the same disadvantage that manual rules has: you need to continuously monitor the incoming feedback for emerging themes, and mis-categorized items.

2. Lack of transparency

Even if the algorithm improves over time, it can be very difficult to understand why it works the way it does, and therefore to tweak the results in a targeted way. Qualitative researchers have told me that this lack of transparency is a main reason why text categorization did not take off in their world. For example, if accuracy suddenly drops when differentiating between two themes like "wait time to install fiber" and "wait time on the phone to set up fiber," how much training data do you need to add before the model stops making these mistakes?

3. Preparing and managing training data is hard

Lack of training data is a real issue. It's hard to start from scratch, and most companies don't have enough (or accurate enough) labeled data to train algorithms well. In fact, teams often overestimate how much usable training data they have, which makes implementations fall below expectations. And if you need to refine one specific category, you may need to re-label large parts of the dataset.

4. Re-training for each new dataset

Transferability can be a real problem. Imagine you have a working text categorization solution for one department (e.g., Support) and now want to analyze feedback from customer surveys like NPS or CSAT. You may need to re-train the algorithm.

A CX practitioner recently shared with me how a team of data scientists spent many months building a solution that she ultimately had to dismiss due to lack of accuracy. The company didn't have time to wait for the algorithm to improve over time.

TEXT ANALYTICS APPROACH 3

Large Language Models

All modern text analytics approaches today rely on language models in some form. Before we get into Generative AI, let's clarify what a language model is, and what makes Large Language Models (LLMs) different.

What is a language model?

A language model is trained by reading a large amount of text and learning patterns in language. At a basic level, it learns: which words and phrases tend to appear together (similarity), and what text is likely to come next (prediction).

Language models place words into a "meaning space", so that similar ideas end up near each other. This meaning representation is called embeddings and it's critical for text analysis.

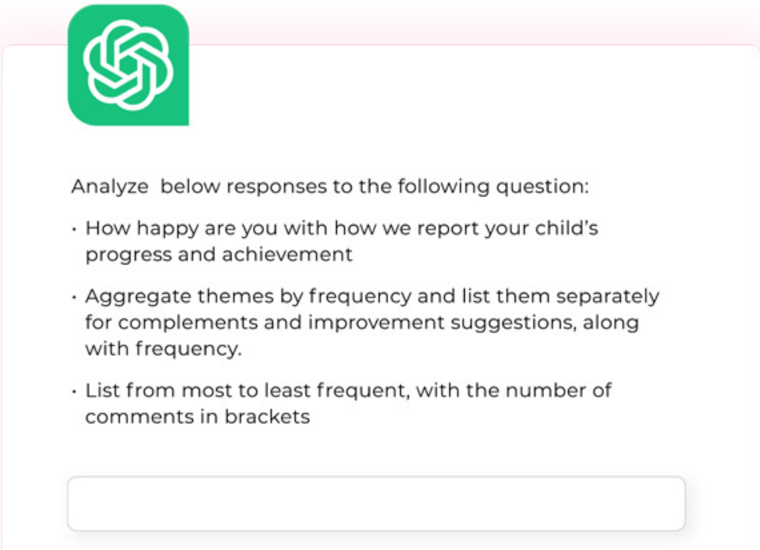
How are LLMs different, and where does Generative AI fit?

Large language models (or LLMs) are much larger (trained on billions of data points with many more parameters) and transformer-based, which means that they can learn complex patterns and handle longer context.

Generative AI is the practical use of these models to analyze and interpret text, discover themes, and even recommendations.

To analyze text, such as customer feedback, you typically provide an instruction, also called a prompt, to define the task and the desired output format. The more specific the instructions (and the more clearly you define edge cases, labels and structure), the more consistent the results.

Here's an example of how I instructed an LLM to analyze a school feedback dataset with 100 parent comments:

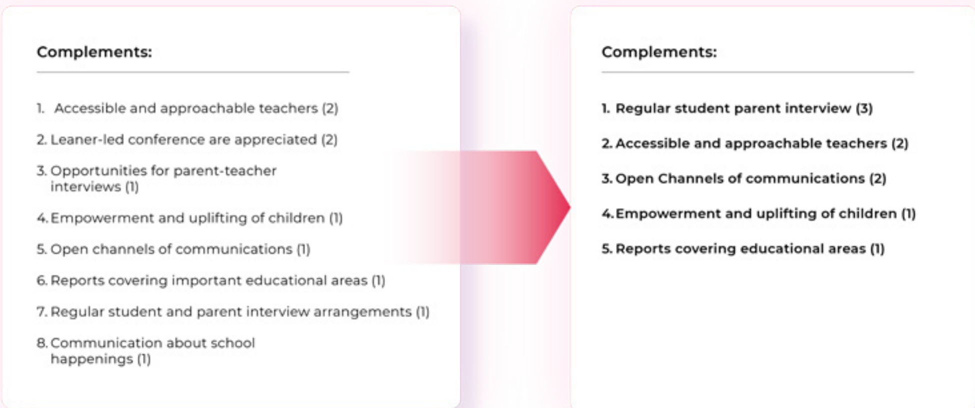


You might get satisfactory results straight away. But more often than not, you'll need to fix errors.

Often LLMs can generate duplicate or overlapping themes.

Here's an example of me correcting issues that resulted from the prompt above.

Even the most recent models with advanced reasoning can make the same mistakes.



Read the full article:

[How to analyze feedback using ChatGPT here.]

What's great about Large Language Models:

While the jury is still out on whether LLMs truly understand language, they are the closest we've come so far. They can interpret acronyms, resolve pronouns, and summarize specialized text surprisingly well, often at a level that feels above what an average person could do quickly.

Compared to manual rules or training a categorization model from scratch, LLMs can generate themes and insights with very little setup. You can often point them at a dataset and get a useful first pass in minutes, without building taxonomies or labeling training examples upfront. In many cases, the biggest "setup" is writing a clear prompt and shaping the output into a format that works for reporting.

Where LLMs fall short:

You can very efficiently analyze a small dataset of 100-500 rows of feedback, and interact with the AI assistant like ChatGPT or Gemina via a chat interface to resolve any issues with analysis, or even interrogate the data without having to tag it with themes. At first, this can feel almost magical, but once you start doing it at scale, you run into many issues.

Non-deterministic results

Outputs can shift based on prompt wording, the order of comments, or subtle context changes. We found that LLMs struggle to manage more than 20 themes. They end up creating duplicate themes or often miss themes that are present in the data.

Hallucinations and overconfidence

Once there is a lot more data, LLMs start to hallucinate. They can produce plausible explanations that aren't fully supported by the data. Plausible does not always mean correct or complete. This was especially a big issue with earlier LLMs. But even newer models like to latch on to proper names to sound knowledgeable.

Reasoning errors:

LLMs still make mistakes in multi-step logic, especially when asked to compare, quantify, or draw conclusions across many comments.

Scale and cost

Very large inputs are expensive and usually require chunking, summarization, and aggregation. This introduces its own trade-offs.

All of this means you need a method of verification and governance, not just a one-off prompt. And when you're working with high volumes of feedback over time, LLMs alone usually aren't sufficient. You need an engineered workflow that reduces hallucinations, supports discovery of emerging themes, keeps themes consistent, and tracks trends reliably.

Finally, it's worth noting that new models are constantly marketed as "the best," whether it's GPT, Gemini, or Grok. Higher benchmark scores don't necessarily mean a model will understand your business or your customers better. In practice, results can vary widely by model version and configuration. All improvements need to be validated against your own data and use cases.

Why LLMs aren't a valid approach for analyzing feedback at scale:

In CX, these limitations matter because the job isn't just to generate a plausible summary, it's to produce insights that are **consistent, repeatable, and trustworthy enough to drive decisions over time**. If results shift with small prompt changes, model updates, or comment ordering, teams can't reliably track trends, compare periods, or align stakeholders on priorities.

Hallucinations, reasoning mistakes, and "plausible but incomplete" outputs can send teams after the wrong issues, eroding trust in the voice of the customer. And at scale, practical constraints mentioned above make it hard to operationalize LLM analysis as a sustainable, governed workflow rather than a one-off experiment.

Thematic Analysis

All approaches mentioned have disadvantages.

In the best case, you'll get OK results only after spending many months setting things up. And you may still miss the **unknown unknowns.**

The cost of acting late or missing out on crucial insights is huge! It can lead to lost customers and stagnant growth. On the flip side, when companies align on the right approach to analyze feedback and act on feedback early they grow faster. According to an American Customer Satisfaction Index analysis comparing customer satisfaction leaders vs. the S&P 500, companies that invest in customer insights can achieve significantly better stock returns (reported as ~4x in that study).

When it comes to customer feedback, three things matter:

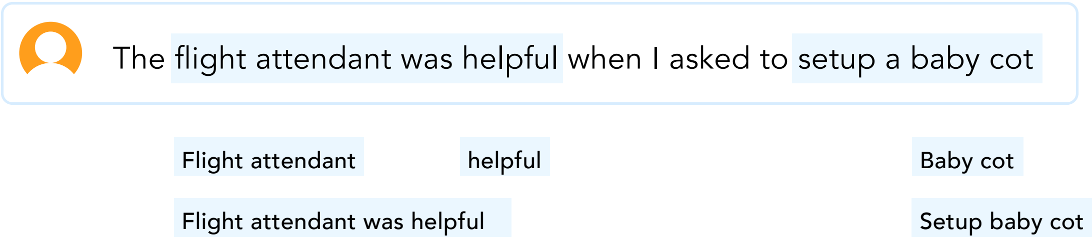
1. Accurate, specific and actionable analysis
2. The ability to spot emerging themes fast, without months of setup
3. Transparency in how results are created, so domain expertise and common sense can be applied

Text analytics has shifted from picking a single method to building a workflow. CX teams can get more value out of customer feedback by using a hybrid system where scalable pattern discovery is combined with LLM-based interpretation and a governed process for humans to review and refine themes. At Thematic, we achieve this using a Generative AI enhanced Thematic Analysis.

Thematic Analysis: How it works

Thematic Analysis doesn't start by forcing text into predefined categories. Instead, it works bottom-up by inferring themes from what people actually say.

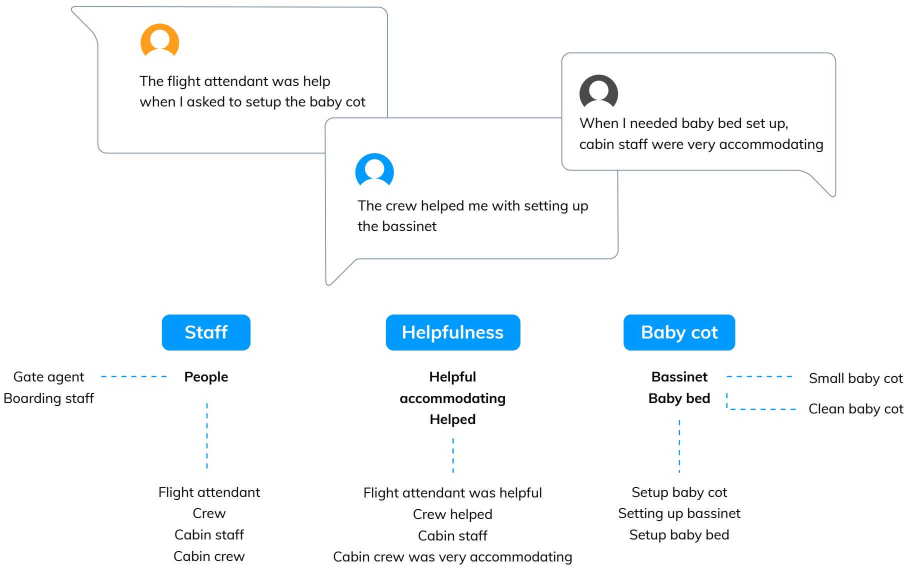
For example, given a comment like: "The flight attendant was helpful when I asked to set up a baby cot," the goal is to identify the key ideas and concepts expressed, e.g. flight attendant, helpfulness, and the baby cot request, and capture them as themes that can be aggregated and analyzed across the full dataset.



However, the most crucial step in a Thematic Analysis approach is merging similar ideas into themes and organizing them in a way that's easy for people to review and edit. Historically, this was often done with semantic similarity and clustering to measure semantic similarity. Today, newer LLMs make this step noticeably better. We found that the right LLMs can help reduce duplicates, pick clearer labels, and build a sensible hierarchy of themes and sub-themes.

In practice, we use a hybrid workflow: scalable clustering to group similar phrases, and LLMs to help refine, merge, and organize those groups into a taxonomy of base themes and sub-themes. The key functionality unique to Thematic is keeping the result transparent and reviewable: each theme can be traced back to the underlying phrases and source comments.

For example, here is how three people talk about the same thing, and how we at Thematic group the results into themes and sub-themes:



Advantages and disadvantages of Thematic Analysis

The advantage of Thematic Analysis is that this approach is unsupervised, meaning that you don't need to set up categories in advance and you don't need to train a model on labeled examples. As a result, it can capture the unknown unknowns, i.e. themes you didn't think to look for.

The disadvantage is that it's difficult to implement correctly in-house. A good approach must be able to merge and organize themes in a meaningful way, producing a set of themes that is not too generic and not too large. Ideally, the themes should cover a large portion of verbatims (people's comments), and the theme inference should handle language complexity such as negation, for example: "I did not think this was a good coffee."

Who does Thematic Analysis?

Theme discovery and hierarchy-building are now used in many places: CX platforms, specialist feedback tools, and even LLM-driven workflows published by model providers. The difference is usually not whether themes can be generated, but whether they are grouped into a stable hierarchy, traceable back to evidence, and easy to review and maintain over time.

Human in the loop

A themes editing interface is critical so that insights professionals can refine themes to suit their business and reporting needs.

For example, an initial model might find phrases such as "fast delivery," "quick and easy," "an hour wait," "slow service," and "delays in delivery," and group them under "speed of service." One team might re-group these into "slow" and "fast" under "speed of service," while another might build a deeper hierarchy (e.g., "fast service" → "quick and easy," and "slow service" → "an hour wait," "delays in delivery"). It's a subjective task, and that's why transparency and easy editing matter.

CUSTOMER INTELLIGENCE

From Text Analytics to Customer Intelligence

Text analytics used to be hard. It required specialist skills, months of setup, and a lot of compromise. That is no longer true. Today, text analytics is close to a commodity. Most CX platforms have some form of analysis built in, and it is easy to build an OK DIY solution using off-the-shelf tools, embeddings, or even a general-purpose AI assistant. For many teams, getting to a first-pass summary of feedback is no longer the bottleneck.

The bigger problem now is fragmentation.

Customer feedback lives across many tools and channels: your survey platform, app reviews, social media monitoring, complaints, support tickets, and call center conversations. Each tool gives you a partial view, and each analysis is done in isolation. Even when every channel is analyzed well, the organization still lacks a single source of truth for what customers are experiencing. Themes are named differently, definitions drift, and insights cannot be compared over time or connected to decisions. This is why teams often feel like they have more “insights” than ever, but less clarity on what to prioritize.

This is where Customer Intelligence starts.

Customer Intelligence is not just better text analytics. It is an operating model where feedback is consistently analyzed across the business and connected to outcomes. In practice, three things matter.

First, apply the same analysis method across all text sources and write the results back into a central database. Think of it like master data management, but for customer signals. You want one set of theme definitions, one way of handling duplicates, one approach to sentiment and qualifiers, and one version of the truth that every team can use. This usually requires a pipeline that ingests feedback from every channel, applies consistent theme logic, and stores results in a system that supports reporting and downstream use.

Second, governance needs to sit with the CX team. If analysis is a black box, or if every analyst has their own prompt and their own labels, you will never get consistency. Customer Intelligence requires that themes are reviewable, editable, and stable over time. It should be easy to manage how themes are defined, merged, split, and named, and to keep those decisions consistent as new feedback arrives. This is also how you prevent the common failure mode of LLM-based analysis: plausible results that change depending on wording, ordering, or model version.

Third, combine qualitative and quantitative data, including financial outcomes. Themes become much more valuable when you can answer questions like: Which themes drive detractors, churn, refunds, repeat contacts, or low adoption? Which problems are costly, and which improvements move revenue or retention? Customer Intelligence links what people say to who they are, what happened in their journey, and what it impacted. That is how feedback becomes prioritization, not just reporting.

If you do these three things, you move from channel-by-channel summaries to a coherent system of customer signals. You can spot emerging themes early, trend them reliably, assign ownership, trigger workflows, and measure whether the changes you make are actually improving customer and employee experience.

Thematic Text Analytics Cheat Sheet

Approach	Thematic Analysis	Manual Rules and Taxonomies	Large Language Models	Text Categorization
How it works	Themes are extracted from text, similar ones merged	Manually crafted and maintained rules	Write a prompt to interpret the data	Categories trained on pre-categorized data
Who is it best for	Companies with small analyst teams looking for productivity gains. Companies in non-standard industries.	Companies in well established industries that do not make major changes to their offering	Companies with expert AI teams for a one-off analysis for quick insights	Companies who have been manually and consistently tagging feedback and do not make major changes to their offering
Data Volume Requirements	300+ feedback pieces/ month	Any volume	Best for small dataset that fits a single prompt	300+ feedback pieces per category
Advantages	No data training required, captures unknowns, easy to use, highly accurate, captures context	Easy to understand	No training required	Can be highly accurate and captures context
Disadvantages	Depends on availability of AI models	Labor intensive, and can't capture unknowns or sentiment	Making changes is labor intensive	Data model requires training, unknowns not captures, changes are labor intensive
Effort to setup	Days	Months	Days to Weeks	Months
Effort to maintain accuracy	**Low** Anyone can maintain, 1-2h per week	**High** Professional services 1 person 1 day a week	**Medium** Requires prompt engineering and management skills	**Medium** Low if categories don't change High if new categories need to be added
Transparency	Very Good	Good	Poor	Poor