

7 time-saving GenAI use cases in KNIME | KNIME

22–28 Minuten

Most data scientists have already experimented with using [GenAI](#) to automate away repetitive tasks. But did you know you can build GenAI functionalities into your KNIME workflows to speed up basic tasks, build basic workflows, and reach the insights you need faster? When scaled across your organization this can deliver huge time and cost savings.

In this blog, we'll look at 7 common GenAI-related use cases you can take advantage of using KNIME to save time, money, and effort! Many of these workflows are live on [KNIME's Community Hub](#) and ready to be used. Where possible we've dropped in a link to the workflow so you can get started using KNIME with GenAI in just a few clicks.

1. Data cleaning and preparation

Every data science project begins with cleaning data so it's in the right shape to be analyzed and worked with. This can be a time-consuming process – especially if you are merging and manipulating data from multiple sources into one dataset.

With KNIME you have two options for using GenAI to help you with this process:

- **Ask K-AI:** KNIME's in-built GenAI chat function can build a small KNIME workflow for you that will clean data

according to your prompt. The benefit of this approach is that you get a KNIME workflow where each data-cleaning step is clear to see. That makes it easy for you to check GenAI's work and correct or amend it as needed.

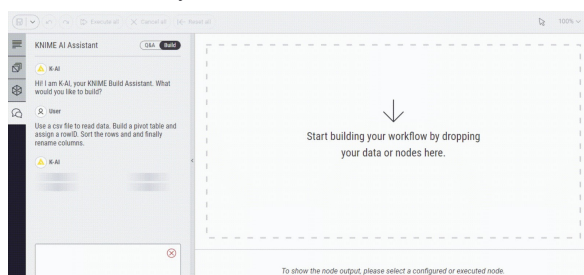
- **Use AI Extension:** KNIME's AI extension can be configured and you can ask your own GenAI provider to take care of cleaning the data for you. In this case, you will not get a KNIME workflow showing the data cleaning steps, because GenAI will take care of this step for you.

Data cleaning and preparation are things all of us need to take care of. So this use case can save you a lot of time and also prevent human error in the process.

Example prompts

- Please merge the two data sources and remove the "country ID" column.
- Please replace any unknown integer or float values with the average (mean).
- Please replace any missing string values with the following string: "N/A".

Here is an example of how K-AI can help you build a workflow and figure out how to clean your data. You need to provide a prompt of what you would like to build and it does the rest for you.



2. Text generation

Text generation can help you draft emails, reports, or articles. It is the process where an AI system automatically produces coherent and contextually relevant text based on a given input imitating human language patterns and styles. It uses large language models to generate content such as articles, summaries, and responses.

Beyond that, data scientists can use it for various natural language processing (NLP) tasks such as summarization, sentiment analysis, translation, and so on. We will go through these in the following sections of this blog.

Data scientists may find text generation useful when personalizing user experiences, creating conversational agents or a chatbot that can understand user queries, provide relevant information, and resolve issues autonomously, or even when automating the generation of reports, dashboards, and summaries based on data analysis.

Here's a step-by-step process for text generation with GenAI in KNIME:

1. **Data Collection:** Gather your text-based dataset. This could include product information documents, clinical trial reports, regulatory documents, or customer reviews. This data forms the knowledge base for the model, providing the context and content from which the language model will generate text.
2. **Preprocessing:** Clean the text data by removing special characters, numbers, and stopwords (GenAI can help with this tool). Then, [tokenize the text data](#) and convert it into a format suitable for input into the language model.
3. **Prompt Engineering:** Create specific prompts that define the task or context for text generation. You'll want to make sure your prompts are specific and provide relevant information for the model to generate text.
4. **Retrieval Augmented Generation:** Use [RAG techniques in KNIME](#) to enhance text generation by providing additional context. This involves retrieving relevant information which means to pull out relevant excerpts or data points from the knowledge base that directly relate to the prompt.
5. **Generate text:** Feed prompts and retrieved information into the language model. The model will use both the prompt and the retrieved context to produce a more accurate and relevant output.
6. **Evaluation:** Evaluate and refine the text generation process based on the quality of the generated text. This can be done by comparing it to human-written benchmarks or predefined quality criteria.
7. **Iterate:** Continuously refine your prompt engineering and RAG processes based on the evaluation results. Iterate to improve the quality and relevance of the generated text.

Let's say, for example, you are a data scientist in the retail industry and you want to use text generation to make the customer experience better by using chatbots and automating dashboards. By doing so you can create a personalized user experience that tailors product recommendations based on individual customer preferences and behaviors. Traditionally reviewing customer feedback, sales reports, or product descriptions can be very time-consuming.

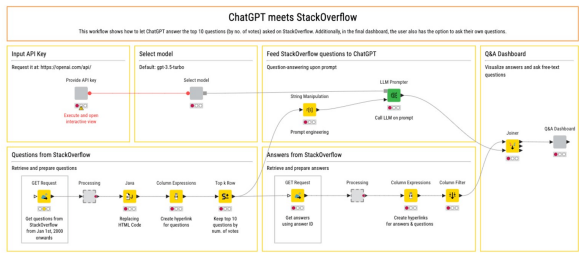
This process can be scaled up to generate text in a fast and efficient way by using a Gen AI workflow. Using KNIME, you can input text data from multiple sources and use a prompt that will guide the LLM in creating the desired text.

Example prompts:

- "Create a product recommendation list for a customer based on their recent purchase history."

- "Generate a detailed response for a customer inquiring about the tracking status of their order."
- "Generate a monthly dashboard summary of customer feedback and satisfaction ratings."

[Here](#) is a KNIME workflow that scrapes the top 10 questions from StackOverflow and uses OpenAI's ChatGPT to generate human-like answers. Check it out!



3. Sentiment analysis

Running a sentiment analysis involves taking advantage of algorithms and natural language processing to effectively classify the sentiment of a text. One approach is to categorize the sentiment into three categories: positive, negative, and neutral. This can help you classify and monitor product reviews or check the tone of emails before you send them, for example.

When running a sentiment analysis without GenAI, you must choose which algorithm you will employ and configure this yourself manually in KNIME. To skip that step GenAI can help you through the process.

Here's a step-by-step process of how to do sentiment analysis with GenAI in KNIME:

1. Data Collection: Gather your text-based dataset. This could be employee feedback, social media posts, customer reviews, or any other text data that contains

sentiment information.

2. Preprocessing: Clean the text data by removing special characters, numbers, and stopwords (GenAI can help with this too!). Then, [tokenize the text data](#) and convert it into a format suitable for input into the language model.

3. GenAI Prompt Engineering: Create prompts that guide the language model to perform sentiment analysis. You'll want to make sure your prompts include specific instructions on how you would like sentiment to be categorized (positive, negative, neutral) and any other relevant information to guide the output you want to obtain.

4. Retrieval Augmented Generation (RAG): Use [RAG techniques in KNIME](#) to enhance the language model's ability to generate accurate sentiment analysis. This process involves retrieving relevant information from the dataset to provide context for the language model.

5. Sentiment Analysis: Input the preprocessed text data and prompts into the language model using KNIME. The language model will generate sentiment analysis predictions based on the provided prompts and context retrieved from the dataset.

6. Evaluation: Evaluate the sentiment analysis results generated by the language model to assess accuracy and performance. It is important to always have a human in the loop to check this. Make any necessary adjustments to the GenAI prompts or RAG techniques to improve the accuracy results.

Let's say, for example, you're a data scientist working for an e-commerce platform, and you want to gain insights into customer sentiment from product reviews. Understanding sentiment orientations – whether they're positive, negative, or neutral – can provide invaluable insights into customer satisfaction, product performance, and areas for improvement. By classifying reviews

accurately, you can identify trends, address issues promptly, and ultimately enhance the overall customer experience.

Reading through customer reviews, labeling each review as positive, negative, or neutral, and then performing statistical analysis to find significant trends can be time-consuming, subjective, and not scalable, especially when dealing with large volumes of reviews.

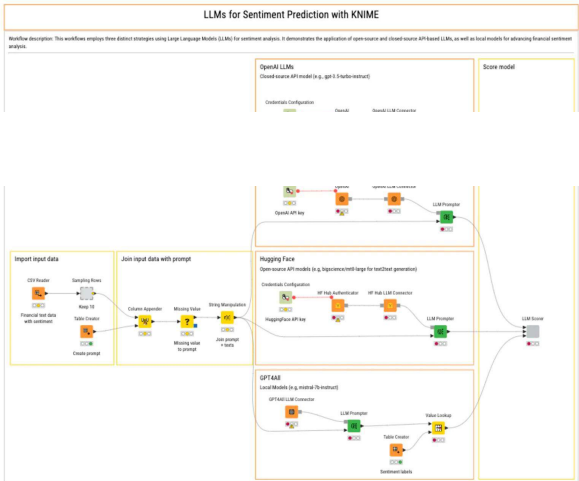
This process can be scaled up to classify the sentiment of large volumes of text in a fast and efficient way by using a Gen AI workflow. Using KNIME, you can input text data from multiple sources and use a prompt that will guide the LLM in effectively classifying the sentiment.

Example prompts

- "Assign a sentiment label, positive or negative, to the following text:".
- "Identify the sentiment regarding shipping/delivery in this review: positive, negative, or neutral."
- "Determine if the sentiment in this review suggests an intention to repurchase, switch brands, or seek alternatives."

By extracting the sentiment of the text using GenAI, you can analyze large volumes of customer reviews in a fraction of the time it would take manually. In this example, as a data scientist in the e-commerce industry, you can use this to identify patterns and trends in customer sentiment and thereby drive improvements in products and services.

[Here](#) is a KNIME workflow to run sentiment analysis. Try it out!



4. Text summarization

Text summarization is condensing a large volume of text into a shorter version while retaining the key information and main ideas. It is useful because it lets you quickly grasp the essence of lengthy documents, articles, or reports without reading them fully. This can be really helpful for analyzing long legal contracts or product documentation.

Here's a step-by-step process of how to do text summarization at scale with GenAI in KNIME:

- 1. Data Collection:** Gather the text-based dataset that you want to summarize. This could be articles, reports, ebooks, policy documents, and so on.
- 2. Preprocessing:** Clean the text data by removing special characters, numbers, and stopwords (GenAI can help with this too!). Then, [tokenize the text data](#) and convert it into a format suitable for input into the language model.
- 3. GenAI Prompt Engineering:** Create prompts that guide the language model to generate the desired summary. You'll want to make sure your prompts include specific instructions on how you would like to summarize the data.
- 4. Retrieve Relevant Information:** Use [RAG techniques in KNIME](#) to identify key sentences or phrases in the text

that should be included in the summary.

5. **Generate Summary:** Input the preprocessed text data and prompts into the language model using KNIME. The language model will generate the text summary based on the prompts and retrieved information.
6. **Evaluation:** Evaluate the generated summary and refine the prompts or retrieval process to improve the summary quality.
7. **Export Summary:** Export the generated summary for further analysis or use.
8. **Iterate:** Iterate on the process to generate more accurate and concise summaries.

For example, consider a data scientist in the context of research and development (R&D) in the pharmaceutical industry. Here, extensive research papers, clinical trial reports, and regulatory documents need to be reviewed regularly. This can be a highly manual and time-consuming task to read hundreds of pages, highlight and note down the important sections, and then summarize.

This process can be scaled up to summarize large volumes of text in a fast and efficient way by using a Gen AI workflow. Using KNIME, you can input text data from multiple sources and use a prompt that will guide the LLM in creating the desired summary.

Example prompts

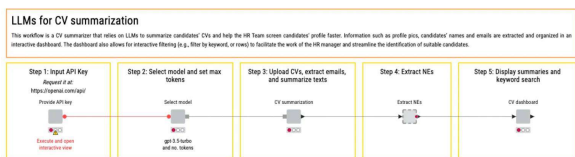
- “Summarize the key points of the regulatory compliance section in the clinical trial report for Drug XYZ.”
- “Summarize the statistical analysis methods used in the clinical trial for Drug XYZ.”

By breaking down lengthy texts into digestible paragraphs or sentences, text summarization can help extract vital information while preserving the meaning and context of the text. In this example, as a data scientist in the

pharmaceutical industry, you can use this text summarization to conduct an in-depth investigation.

You can now focus on the more important tasks like deriving data-driven insights such as finding correlation and demographic dependencies in your analysis and strategic decision-making like formulating recommendations for future research and development efforts.

[Here](#) is a KNIME workflow that relies on LLMs to summarize candidates' CVs and help the HR Team screen candidates' profiles faster. Try it yourself!



5. Question answering

Question answering automatically uses models to respond to questions based on a given context. This can be particularly useful for extracting information from documents or generating answers without explicit context. For example, if you have a company knowledge base, you can use GenAI to read the knowledge base and feedback answers. You can do this with any type of document, including legal contracts.

Here's a step-by-step process of how to do question answering with GenAI in KNIME:

1. **Data Collection:** Gather the text-based datasets that you want to extract information from. This could be articles, reports, ebooks, policy documents, and so on.
2. **Preprocessing:** Clean the text data by removing special characters, numbers, and stopwords (GenAI can help with this too!). Then, [tokenize the text data](#) and convert it into a

format suitable for input into the language model.

3. **Prompt Engineering:** Create prompts that provide context and that are specific. This will guide the model to generate accurate answers.
4. **Retrieval-Augmented Generation:** Using [RAG techniques in KNIME](#) to find relevant answers to your question can help you obtain a deep contextual understanding. This is because the model considers multiple sources and chooses the most appropriate answer based on the context of the question.
5. **Answering the Question:** Use KNIME's text processing and machine learning nodes to implement the question-answering system. Utilize KNIME's integration with external libraries for retrieval and generation tasks.

For example, let's say you are a data scientist working in the manufacturing industry and you have to manage and review multiple contract documents. These contracts might include agreements with suppliers, service providers, or customers, each containing critical information spread across numerous pages. Manually reading through each document to extract essential details can be a highly labor-intensive, time-consuming, and error-prone task.

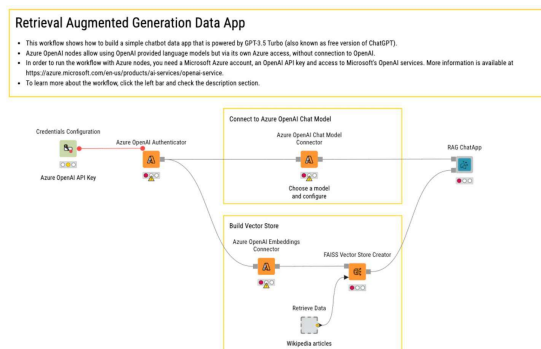
Building a question-answer workflow using GenAI can help you to automatically extract accurate and informative responses without having to go through copious amounts of text. Using KNIME, you can input text data from multiple sources and use a prompt that will guide the LLM to generate answers.

Example prompts

- “What is the effective date of the contract?”
- “Is there a clause for liability limitations?”
- “Are there any late payment penalties specified?”

Automating the extraction of key information saves time and effort, enabling quicker responses to critical queries. In this example, as a data scientist in the manufacturing industry, using a GenAI question-answer system can streamline processes and help you focus on higher-value tasks such as optimizing operations and supporting strategic decisions.

[Here](#) is a workflow that answers questions using Wikipedia data. Try it yourself and customize it to fit your needs!



6. Language translation

Language translation using [large language models](#) (LLMs) involves converting text from one language to another. This task is approached as a sequence-to-sequence problem, where the model learns to map a sequence of words in the original language in your data also known as source language to a sequence in the language that you would like as the output also known as target language.

Using LLMs for language translation automates the process of converting large volumes of text between languages, which would otherwise be time-consuming and require significant human effort. Here's a step-by-step

process of language translation with GenAI in KNIME:

1. **Data Collection:** Gather the text-based dataset that you want to translate. This could be articles, reports, ebooks, etc.
2. **Preprocessing:** Clean the text data by removing special characters, numbers, and stopwords (GenAI can help with this too!). Then, [tokenize the text data](#) and convert it into a format suitable for input into the language model.
3. **Prompt Engineering:** Design prompts that provide context and guidance to the language model for accurate translation. Include information such as source language, target language, and any specific instructions for translation.
4. **Retrieval Augmented Generation:** Use [RAG techniques in KNIME](#) to retrieve relevant information from the language model. Combine the retrieved information with the input prompt to generate the translated output.
5. **Implementation in KNIME:** Use KNIME nodes to implement the language translation process. Utilize KNIME's text processing and machine learning nodes for efficient translation.

Language translation is a powerful tool for businesses to scale up and break down barriers simultaneously. For example, consider you are a data scientist in the banking and finance industry and want to conduct a global market analysis. This involves analyzing financial news, reports, and market data from multiple languages to understand global financial trends and make informed decisions. Having translators convert these documents into comprehensible language can be expensive, time-consuming and limits the ability to quickly react to global financial events.

Building a language translation workflow using GenAI can help you quickly translate your text data and perform real-time analysis. Using KNIME, you can aggregate your text data from multiple sources and use a prompt that will

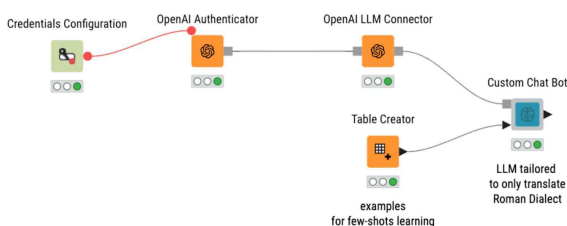
guide the LLM in translating your data into the desired language.

Example prompts

- "Translate this Italian regulatory update on new banking regulations into English."
- "Translate this French news article about the European stock market into English."
- "Translate this Japanese market analysis report on the Tokyo Stock Exchange into English."

In this use case, having a GenAI workflow for language translation can help you stay informed about global financial trends and make strategic decisions based on comprehensive and timely data. This not only allows you to respond swiftly to global market changes but also frees up time to analyze critical financial information from multiple languages, providing valuable insights into global market trends.

Here is an example workflow of language translation:
[Roman Translator App via LLM](#)



7. Topic modeling

Topic modeling identifies hidden themes or topics within a large collection of texts. It helps in organizing, understanding, and summarizing large datasets by

clustering similar words and phrases together into coherent topics. It helps automate the analysis of vast amounts of text data, enabling quick identification of key themes without manual reading. Here's a step-by-step process of topic modeling with GenAI in KNIME:

1. **Data Collection:** Gather the text-based dataset that you want to translate and the corresponding contexts or documents. This could include Wikipedia articles, research papers, books, and so on.
2. **Preprocessing:** Clean the text data by removing special characters, numbers, and stopwords (GenAI can help with this too!). Then, [tokenize the text data](#) and convert it into a format suitable for topic modeling.
3. **Topic Modeling with Large Language Models:** Use a pre-trained large language model for topic modeling to generate embeddings for the text data. Clustering algorithms such as K-means or DBSCAN can be applied to group similar documents based on the embeddings.
4. **Prompt Engineering:** Create prompts that are specific to the task of topic modeling. Design prompts that guide the language model to generate relevant topics from the text data. Experiment with different prompt structures to optimize topic generation.
5. **Retrieval Augmented Generation (RAG):** Use [RAG techniques in KNIME](#) to retrieve relevant documents based on the generated topics. Use the retrieved documents to refine the generated topics and improve the overall quality of topic modeling results.

6. **Iterate:** Iterate the retrieval and generation process to enhance the accuracy of the topics generated.

For example, consider you are a data scientist in the e-commerce industry, and you want to use topic modeling to get insights into product categorization, improve recommendation systems, and understand consumer behavior based on reviews and product descriptions. Topic modeling involves extracting themes or topics from a

collection of documents, such as customer reviews and product descriptions, to uncover hidden patterns and trends.

Relying on the categorization of products based on predefined categories or tags using basic keyword matching to suggest products to customers can lead to limited personalization and effectiveness. This also limits the understanding of consumer behavior.

Building a topic modeling workflow using GenAI can help you automate and improve the accuracy of product categorization based on semantic similarities and themes. It can also create a personalized recommendation system for customers based on their preferences and behavior. Using KNIME, you can input your text data from multiple sources and use a prompt that will guide the LLM in extracting topics from your data.

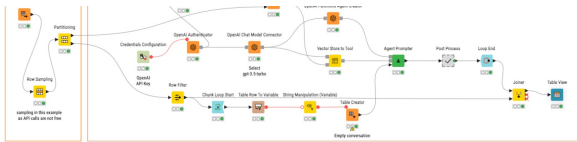
Example prompts

- "Cluster similar products based on their descriptions and highlight the common themes."
- "Suggest complementary products to customers based on the common themes found in their recent purchases."
- "Flag products with inconsistent or misleading descriptions based on topic analysis."

In this use case, having a GenAI workflow for extracting topics from a collection of documents can help you uncover nuanced insights that may not be immediately apparent. It also helps in understanding large volumes of text data and extracting key features or topics. Topic modeling can uncover hidden patterns and relationships within the data enabling personalized recommendations.

[Here](#) is a KNIME workflow for topic modeling. Try it out!





Disclaimer: It is important to note that the data given to the data app is shared with Azure OpenAI (or whichever LLM provider is used). Internal documents **should not** be used as input unless approved by the organization that owns the documents.

GenAI-powered workflows for data scientists

Incorporating Generative AI into your KNIME workflows can significantly help you by automating repetitive and time-consuming tasks. By leveraging these use cases, data scientists and analysts can focus on higher-level analysis and decision-making, accelerating the path to valuable insights.

Explore these practical applications on [KNIME's Community Hub](#) and integrate GenAI to transform your workflows.