The AI Product Builder's Toolkit; From Concepts to Agents

Dhara Mungra
University of Mannheim; SimPPL

Building Blocks for What You'll Build

Last session: WHY we're building social listening products
Today: WHAT tools and concepts you need to build them

Format:

- I will ask questions. Answer honestly, because this is not a test.
- If you know it, share. If you do not, that is fine, because why we're here.
- By the end, we will all know where we stand.

Computers don't understand words. So how do we make them?

Embeddings: Teaching Machines to "Understand" Language

The idea: Convert every word, sentence, or document into a list of numbers (a vector) that captures its MEANING.

Imagine every sentence gets a GPS coordinate in "meaning space."

- "The vaccine is dangerous" → [0.8, -0.3, 0.5, ...]
- "The shot is unsafe" → [0.79, -0.31, 0.48, ...]
- "I love chocolate cake" → [-0.2, 0.9, -0.1, ...]

The first two are CLOSE together. The third is FAR away.

Embeddings: Teaching Machines to "Understand" Language

How it works:

1. Models like BERT, GPT, or Sentence-BERT read the text
2. They output a vector (e.g., 768 numbers)
3. Similar meanings → similar vectors

Read more:

- Jay Alammar — "The Illustrated Word2Vec"
- Sentence embeddings: SBERT.net

Semantic Similarity: Measuring "Closeness" in Meaning

Where might you use this in a social listening product?

Semantic Similarity: Measuring "Closeness" in Meaning

Social listening applications:

- Deduplication: Finding near-duplicate posts across platforms
- Narrative tracking: "Is this new post saying the same thing as yesterday's viral tweet?"
- Search: "Show me all posts SIMILAR to this example"
- Clustering: Group thousands of posts by meaning → topic modeling

Quick Check: What Models Have You Used?

Name ONE model, library, or tool you have used for any NLP or ML task. Anything counts, including ChatGPT, scikit-learn, a Hugging Face model, NLTK, even Excel.

Three Ways to Understand What People Are Saying

What is the difference between sentiment, emotion, and stance? Are they the same thing?

Three Ways to Understand What People Are Saying

What is the difference between sentiment, emotion, and stance? Are they the same thing?

For example - 'I can't believe they approved that drug.'

Topic Modeling: What Are People Talking About?

If you had 100,000 tweets about public health, how would you figure out the main topics people are discussing?

Topic Modeling: What Are People Talking About?

Key tools:

- BERTopic — the standard for modern topic modeling
- Nomic Atlas — visual topic exploration at scale, auto-labels clusters with LLMs
- LLM layer — use GPT-4/Claude to turn clusters into narrative descriptions

Sources

- BERTopic Documentation
- Nomic Atlas — Topic Modeling
- BERTopic + LLMs Tutorial

How Do You Know If Your Model Is Good?

If your sentiment model says 85% accuracy, is that good? What if 90% of your data is positive?

Quick Check: Experience with These Tasks

Raise your hand if you have:

- Built or used a sentiment analysis model
- Heard of stance detection before today
- Done any form of topic modeling
- Evaluated a model beyond accuracy (precision, recall, F1)
- Manually annotated data for an NLP task

Large Language Models: When to Use Them, When Not To

You all use ChatGPT. But what IS it, technically? What makes it 'large'?

Large Language Models: When to Use Them, When Not To

What LLMs are:

- Neural networks trained on massive text corpora
- They predict the next token (word) based on everything before it
- "Large" = billions of parameters (GPT-4: ~1.8T, Llama 3: 405B, Apple FM: ~3B on-device)

Large Language Models: When to Use Them, When Not To

When TO use LLMs for social listening:

- Theme generation from topic clusters (qualitative interpretation)
- Zero-shot classification when you have no labeled data
- Summarization of large volumes of content
- Generating human-readable reports from data

Large Language Models: When to Use Them, When Not To

When NOT to use LLMs:

- High-volume classification (too slow, too expensive at scale)
- When you need consistent, reproducible outputs
- When a fine-tuned BERT model does the job faster and cheaper
- When you can't afford hallucinations

Large Language Models: When to Use Them, When Not To

When NOT to use LLMs:

- High-volume classification (too slow, too expensive at scale)
- When you need consistent, reproducible outputs
- When a fine-tuned BERT model does the job faster and cheaper
- When you can't afford hallucinations

Prompting: Structure & Output Schemas

When you use ChatGPT, do you just type a question? Or do you structure your prompts?
Show me — what does your typical prompt look like?

Prompting: Structure & Output Schemas

Output schemas matter for products:

- Free text responses → impossible to process programmatically
- Structured JSON → can be parsed, stored, aggregated, visualized
- This is the difference between a chatbot and a product pipeline

Key techniques:

- Zero-shot: No examples, just instructions
- Few-shot: 1–5 examples included in prompt
- Chain-of-thought: "Think step by step before answering"

Vibe Coding: The Elephant in the Room

How many of you have asked ChatGPT or Copilot to write code for you? Be honest.

Vibe Coding: The Elephant in the Room

Describing desired functionality in natural language, letting AI generate code.

The good:

- Rapid prototyping — test ideas fast
- Learning tool — see how something is implemented
- Boilerplate generation — save time on repetitive code

The dangerous:

- Security vulnerabilities: AI-generated code has 2.74x more security issues than human-written code (CodeRabbit, Dec 2025)
- Technical debt: inconsistent patterns, no documentation, "vibe-coded messes"
- False confidence: "it runs" ≠ "it works correctly"
- No architectural vision: each prompt solves ONE problem without seeing the system

Vibe Coding: The Elephant in the Room

Vibe code to PROTOTYPE. Engineer to SHIP

What Is an AI Agent?

What is the difference between ChatGPT and an AI agent?

What Is an AI Agent?

An agent = Model + Tools + Memory

Component	What It Does	Example
Model (the brain)	Reasons about what to do next	GPT-4, Claude, Llama
Tools (the hands)	Takes actions in the real world	Search the web, query a database, call an API, send an email
Memory (the context)	Remembers past interactions and state	Conversation history, user preferences, session data

Model Context Protocol (MCP): Why It Matters

If you had 10 different data sources – Twitter API, a database, a web scraper, an email system – how would you connect them all to your AI agent?

Model Context Protocol (MCP): Why It Matters

MCP = USB-C for AI agents. Understand what the dataset is
One standard protocol. Any server. Any client.

- Introduced by Anthropic, November 2024
- Adopted by OpenAI, Google DeepMind, and others
- Open standard — anyone can build MCP servers

Why it matters for your products:

- You build ONE interface, connect to MANY data sources
- Give Data Understanding to your AI Agents
- Your social listening product can ingest Twitter, Reddit, news feeds — all through MCP servers

Making Agents Smarter: Skills & Orchestration

Skills library: A collection of reusable capabilities an agent can invoke.

- "Analyze sentiment of these posts" → calls the right model, returns structured results
- "Generate a topic summary" → runs BERTopic + LLM theme generation
- "Search for related content" → uses semantic similarity
- Think of it as the agent's toolkit — modular, reusable, composable

Making Agents Smarter: Skills & Orchestration

For this course: LangChain is likely your starting point.

Data Infrastructure: Quick Diagnostic

Name different databases you have used

Data Infrastructure: Quick Diagnostic

Service	What It Is	Use When
S3	Object storage (files, images, raw data)	Storing raw datasets, exports, backups. Cheap, unlimited, slow for queries
DynamoDB	NoSQL key-value database	Fast reads/writes by a known key. User profiles, session data, metadata
Elasticsearch	Full-text search & analytics engine	Complex text queries, filtering, aggregation. "Find all posts about X from Y time period"

Data Infrastructure: Quick Diagnostic

For social listening:

- Raw data (tweets, articles) → S3
- User/project metadata → DynamoDB
- Searchable post index → Elasticsearch

Git & GitHub: The Non-Negotiable

How many of you have used Git solo or in a team project ?

Git & GitHub: The Non-Negotiable

Why this matters:

- You will work in teams of 3–4
- You will write code together
- If you don't use Git, you WILL lose work, overwrite each other, and waste hours

Minimum you need:

- git clone, git add, git commit, git push, git pull
- Branching: git checkout -b feature-name
- Pull requests: code review before merging
- .gitignore: don't commit API keys or data files

Git & GitHub: The Non-Negotiable

If you're new to Git:

- GitHub: Git Handbook
- Oh Shit, Git!?!
- Atlassian Git Tutorial

Group Formation

Teams of 3–4 members.

What makes a good team:

- Mix of skills (technical + design + communication)
- At least one person comfortable with code
- At least one person who enjoys talking to users
- Everyone willing to learn what they don't know

Process:

1. I'll read out the skills map from today (2 min)
2. Mingle and talk to people you haven't talked to yet (8 min)
3. Form your groups and register on [form/sheet] (5 min)
4. Each group: tell me your team name and one sentence about what you're curious to build (5 min)

Before Next Class

To do this week:

- If new to Git: complete the GitHub Git Handbook
- Jay Alammar — "The Illustrated Word2Vec"
- Explore: Nomic Atlas Demo — play with a topic map
- In your new team: set up a group chat and share one social listening problem you'd like to explore

Next session:

- Hands-on workshop: from raw data → embeddings → topics → themes
- Bring your laptops

Feedback!

Ask Me Anything!

Questions?
Please reach out to
dhara.atul.mungra@uni-mannheim.de

Thank you!