

- Intent In simple terms, **intent** means the purpose or goal behind a user's action or message.
- In everyday language: if someone says, "*Book me a ticket to Pokhara*", their **intent** is not just saying words—it's *wanting to book a ticket*.
-
- **TF-IDF score** = $TF \times IDF$
 - High score → The word is frequent in one document but rare across others (likely meaningful).
 - Low score → The word is common everywhere (like "the", "is", "and") — usually not helpful for distinguishing meaning.

TF-IDF stands for **Term Frequency – Inverse Document Frequency**.

It's a numerical statistic that measures how important a word is in a document compared to all other documents in the dataset.

- **TF (Term Frequency)** → How often a word appears in a document.
Formula:
- **IDF (Inverse Document Frequency)** → How unique the word is across all documents.

Instead of just one score, we calculate TF-IDF for **every word** in the vocabulary for a given text, producing a **vector** (list of numbers).

This **vector** is the numerical representation of the text that can be fed into machine learning models.

🔍 **Collect training data:** A set of example sentences for each intent (e.g., *Book_Flight*, *Get_Weather*, *Cancel_Ticket*).

🔍 **Preprocess** the text: Lowercase, remove stopwords, maybe stem or lemmatize.

🔍 **Convert to TF-IDF vectors:** Each sentence becomes a numerical vector.

🔍 **Train a classifier:** Feed vectors into an ML algorithm (e.g., Logistic Regression, SVM, Naive Bayes) to learn patterns for each intent.

🔍 **Prediction:**

- A new user query is transformed into a TF-IDF vector.

- The classifier predicts the most likely intent based on training.

Data splitting

- We split the dataset into:
 - **Training set** (e.g., 80%): Used to train the model.
 - **Validation set** (e.g., 20%): Used to check performance on unseen data.
- This ensures we measure how well the model **generalizes**, not just memorizes.

What is Random Forest?

- **Random Forest** is a **supervised machine learning algorithm** used for classification and regression tasks.
- It's an **ensemble method**, meaning it combines the output of multiple models (here, **decision trees**) to make a better overall prediction.
- The "forest" = many decision trees, each trained on a slightly different version of the data.

Model initialization (Random Forest)

- A **Random Forest** is an **ensemble of decision trees**.
- Each tree is trained on a random subset of data and features.
- Hyperparameters you might set:
 - `n_estimators` → number of trees (e.g., 100)
 - `max_depth` → maximum depth of each tree
 - `random_state` → seed for reproducibility
- Why Random Forest?
 - Handles **non-linear relationships**
 - Good for **small to medium datasets**
 - Works well with **sparse vectors** like TF-IDF output

Model training

- The Random Forest is trained using **training data TF-IDF vectors** as inputs and **intent labels** as outputs.
- Each decision tree learns **word patterns** that indicate specific intents.
- Example:

- If "book" and "flight" have high TF-IDF scores → Decision: Intent = Book_Flight.

Model evaluation

- We use the **validation set** to see how accurate the model is.
 - Common metrics:
 - **Accuracy** → % of correct predictions
 - **Precision** → How often the predicted intent is correct
 - **Recall** → How many correct intents the model catches
 - **F1-score** → Balance between precision and recall
 - This step tells us if the model is ready or needs improvement.
-

g) Save model and vectorizer

- After training, save:
 - **Trained Random Forest model**
 - **Fitted TF-IDF vectorizer** (so new queries are transformed the same way as training queries)
 - Why save? So you don't have to retrain every time — you can just load and use.
-

h) Predict intent for new queries

- When a new user query arrives:
 - Load the saved **TF-IDF vectorizer**.
 - Transform the query into a vector (same process as training).
-

i) Intent prediction

- Feed the vector into the **saved Random Forest model**.
- Model outputs a predicted **intent label**.
- Example:

vbnet

CopyEdit

Input: "Show me hotels in Kathmandu"

Output: Find_Hotel

j) Return result

- Send the predicted intent to the **chatbot** or **application**.
- The system then decides **what action to take**:
 - If intent = Get_Weather → Fetch weather API
 - If intent = Book_Flight → Start booking flow