UnCodemy

NLP Mini Project Ideas PDF

An NLP mini project should teach one text skill in one to four weeks: cleaning and tokenising, TF-IDF classification, named entities or simple text generation. A spam filter, product review sentiment analyser, keyword extractor and FAQ chatbot are good first projects using NLTK, spaCy and scikit-learn.

Key takeaways

- Learn text cleaning and TF-IDF before moving to transformers.
- NLTK and spaCy cover most mini project needs.
- Keep a small test set of sentences you write yourself.
- Show word clouds or top-word charts to explain results.
- Each idea fits in one to four weeks.

Who this PDF is for

This PDF is for 2nd and 3rd year students who want a first natural language processing project. The ideas use small public datasets or text you collect, and run on any laptop. They move from basics (tokens, stop words, TF-IDF) to entities, similarity and small models.

How to choose your project

- **Pick text you understand** such as reviews, messages or news.
- **Start with English**, then try Hindi as an extension.
- **Keep the vocabulary small** at first so training is quick.
- **Plan a simple input box** for the demo.

Beginner mini projects (1 to 2 weeks)

1. Text Preprocessing Toolkit

- **Problem:** Build a reusable cleaner for any text.
- **Dataset:** Any sample text, such as 20 Newsgroups from scikit-learn.
- **Tech stack:** Python, NLTK, spaCy.
- **Main modules:** Lowercasing, tokenising, stop words, stemming versus lemmatising, report.
- **Steps:** Write functions for each step, compare stemming and lemmatising outputs.
- **Show in the viva:** Why lemmas are often better than stems.
- **Extensions:** Package it as a small library.

2. Email Spam Filter

- **Problem:** Classify emails as spam or not.
- **Dataset:** UCI "Spambase" or the SMS Spam Collection.
- **Tech stack:** Python, scikit-learn.
- **Main modules:** Features, Naive Bayes, metrics, test box.
- **Steps:** Build TF-IDF or use the given features, train Naive Bayes, report precision and recall.
- **Show in the viva:** Why precision matters for spam (good mail must not be lost).
- **Extensions:** Compare logistic regression.

3. Product Review Sentiment Analyser

- **Problem:** Label reviews as positive, negative or neutral.
- **Dataset:** A public product review set on Kaggle or Hugging Face, checked for licence.
- **Tech stack:** Python, NLTK VADER, scikit-learn.
- **Main modules:** Lexicon method, trained model, comparison table.
- **Steps:** Score with VADER, train TF-IDF plus logistic regression, compare on labelled reviews.
- **Show in the viva:** When a lexicon fails (sarcasm, negation).
- **Extensions:** Find most common complaint words.

4. Keyword Extractor

- **Problem:** Pull key phrases from an article.
- **Dataset:** Any news articles you copy for study, or 20 Newsgroups.
- **Tech stack:** Python, scikit-learn, spaCy, RAKE or YAKE.
- **Main modules:** TF-IDF keywords, RAKE or YAKE keywords, noun-phrase filter.
- **Steps:** Run two methods, compare top 10 phrases on five articles.
- **Show in the viva:** Differences between the two methods.
- **Extensions:** Generate tags for a blog.

Intermediate mini projects (2 to 3 weeks)

5. FAQ Chatbot for a College

- **Problem:** Answer common student questions.
- **Dataset:** 40 to 80 question-answer pairs you write from your college notices.
- **Tech stack:** Python, scikit-learn or sentence-transformers, Flask.
- **Main modules:** Question bank, similarity search, fallback reply, chat UI.
- **Steps:** Embed questions, match user input by cosine similarity, reply with the closest answer above a threshold.
- **Show in the viva:** How the threshold avoids wrong answers.
- **Extensions:** Add Hindi questions.

6. Named Entity Highlighter

- **Problem:** Highlight people, places and organisations in text.
- **Dataset:** Any news text; spaCy's pretrained English model.
- **Tech stack:** Python, spaCy, displaCy, Streamlit.

- **Main modules:** Entity extraction, visual highlighting, entity count chart.
- **Steps:** Run spaCy NER, render with displaCy, count entity types.
- **Show in the viva:** Mistakes the model makes on Indian names.
- **Extensions:** Add custom entity rules.

7. News Headline Category Classifier

- **Problem:** Classify headlines into topics.
- **Dataset:** AG News dataset or 20 Newsgroups.
- **Tech stack:** Python, scikit-learn.
- **Main modules:** TF-IDF, linear SVM, confusion matrix.
- **Steps:** Train, test, list headlines that were misclassified.
- **Show in the viva:** Which categories overlap.
- **Extensions:** Try word embeddings.

8. Plagiarism Checker

- **Problem:** Measure similarity between two documents.
- **Dataset:** Your own sample assignments.
- **Tech stack:** Python, scikit-learn, sentence-transformers.
- **Main modules:** Cosine similarity on TF-IDF, embedding similarity, highlighted overlaps.
- **Steps:** Compare documents with both methods, show matching sentences.
- **Show in the viva:** Why paraphrases fool TF-IDF but not embeddings.
- **Extensions:** Batch-check a whole folder.

Stretch mini projects (3 to 4 weeks)

9. Text Summariser

- **Problem:** Summarise an article to a few sentences.
- **Dataset:** Any news articles; CNN/DailyMail samples for testing.
- **Tech stack:** Python, NLTK, a pretrained summarisation model from Hugging Face.
- **Main modules:** Frequency-based summary, TextRank, pretrained model, side-by-side view.
- **Steps:** Build the simple methods yourself, then compare with a pretrained model.
- **Show in the viva:** Extractive versus abstractive summaries.
- **Extensions:** Add a length slider.

10. Autocomplete with N-grams

- **Problem:** Suggest the next word as a user types.
- **Dataset:** Public-domain books from Project Gutenberg.
- **Tech stack:** Python, NLTK.
- **Main modules:** Tokenising, n-gram counts, smoothing, suggestion box.
- **Steps:** Build bigram and trigram models, add smoothing, suggest top 3 next words.
- **Show in the viva:** What smoothing solves.
- **Extensions:** Compare with a small LSTM.

11. Language Detector

- **Problem:** Detect whether text is English, Hindi, Marathi and so on.
- **Dataset:** Short public sentences per language, for example from Tatoeba.
- **Tech stack:** Python, scikit-learn.
- **Main modules:** Character n-gram features, Naive Bayes, confusion matrix.
- **Steps:** Build char n-grams, train, test on short and long sentences.
- **Show in the viva:** Why character n-grams work for language ID.
- **Extensions:** Detect romanised Hindi.

12. Resume Keyword Matcher

- **Problem:** Score how well a resume matches a job post.
- **Dataset:** Your own resume and sample job posts.
- **Tech stack:** Python, spaCy, scikit-learn.
- **Main modules:** Text extraction, skill list, similarity score, missing-skill report.
- **Steps:** Extract skills, compute overlap and TF-IDF similarity, list missing skills.
- **Show in the viva:** How you built the skill list.
- **Extensions:** Suggest learning resources for missing skills.

Mini project report structure

A mini project report is usually 10 to 20 pages. Keep it short and clear:

1. **Title page and certificate** in your department's format.
2. **Introduction:** the problem in one paragraph and your objectives as a list.
3. **Tools used:** Python, NLTK, spaCy or scikit-learn versions, and the dataset source.
4. **Design:** one block diagram or flowchart, plus the text cleaning steps.
5. **Implementation:** what each module does, with 3 to 6 screenshots.
6. **Results:** metrics or example outputs with a short error list.
7. **Conclusion and future scope,** then references.

Ask your guide for the exact format before you start writing.

Common mistakes to avoid

- Removing stop words for sentiment and losing words like "not".
- Testing on sentences that were in the training data.
- Using copyrighted text without checking terms.
- Not showing sample inputs and outputs in the report.

Frequently asked questions

What is a good NLP mini project for beginners?

A spam filter or a review sentiment analyser. Both use small datasets and teach text cleaning and TF-IDF.

NLTK or spaCy for a mini project?

NLTK is good for learning basics such as tokenising and stemming. spaCy is faster and better for entities and pipelines. Many projects use both.

Can I build a chatbot as a mini project?

Yes. A retrieval FAQ chatbot that matches questions by similarity is a good scope for two to three weeks.

Do I need a GPU for NLP mini projects?

No. The ideas here run on a normal laptop. Use Colab only if you try a transformer model.

Does this PDF include full source code?

No. It gives the problem, data, stack, modules and steps for each project. You write the code yourself, which is what examiners test in the viva.

Is this PDF free?

Yes. Fill in the short form on this page and the PDF downloads right away.

Sources and further reading

- NLTK documentation
- spaCy usage documentation
- UCI Machine Learning Repository
- Project Gutenberg

About this PDF

Prepared by the Uncodemy training team as practice project ideas. Datasets and tools named are public; check each dataset's licence before use. We update it every January. Corrections: info@uncodemy.com.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550