Natural Language Processing Tasks

Note:

Participants must complete at least
3 tasks for the 2-week internship and
4 tasks for the 1-month internship
— from any level.

Level 1

Task 1: Sentiment Analysis on Product Reviews

Description:

- Dataset (Recommended): IMDb Reviews or Amazon Product Reviews (Kaggle)
- Analyze product reviews to determine whether the sentiment is positive or negative
- Clean and preprocess text (e.g., lowercasing, removing stopwords)
- Convert text to numerical format using TF-IDF or CountVectorizer
- Train a binary classifier (e.g., logistic regression) and evaluate its performance

Tools & Libraries:

Python | Pandas | NLTK/spacy | Scikit-learn

Covered Topics

Text classification | Sentiment analysis

Bonus:

Visualize the most frequent positive and negative words
Try using a Naive Bayes classifier and compare accuracy

Task 2: News Category Classification

Description:

- Dataset (Recommended): AG News (Kaggle)
- Classify news articles into categories such as sports, business, politics, technology, etc.
- Perform standard text preprocessing (tokenization, stopword removal, lemmatization)
- Vectorize the text using TF-IDF or word embeddings
- Train a multiclass classifier (e.g., Logistic Regression, Random Forest, or SVM)

Tools & Libraries:

Pandas | Scikit-learn | Optionally: XGBoost or LightGBM

Covered Topics

Multiclass classification | Text preprocessing | Feature engineering with text

Bonus:

Visualize the most frequent words per category using bar plots or word clouds
Try training the model using a neural network (e.g., simple feedforward NN with Keras)

Natural Language Processing Tasks

Level 2

Task 3: Fake News Detection

Description:

- Dataset (Recommended): Fake and Real News Dataset (Kaggle)
- Classify news articles as real or fake based on their text content
- Preprocess title and content (remove stopwords, lemmatize, vectorize)
- Train a logistic regression or SVM classifier
- Evaluate using accuracy and F1-score

Tools & Libraries:

Python | Pandas | NLTK/spacy | Scikit-learn

Covered Topics

Binary classification | TF-IDF and preprocessing

Bonus:

Use a word cloud to visualize common terms in fake vs. real news

Task 4: Named Entity Recognition (NER) from News Articles

Description:

- Dataset (Recommended): CoNLL003 (Kaggle)
- Identify named entities (like people, locations, and organizations) from article content
- Use rule-based and model-based NER approaches
- Highlight and categorize extracted entities in the text

Tools & Libraries:

Python | SpaCy | Pandas

Covered Topics

Sequence labeling | NER (Named Entity Recognition)

Bonus:

Visualize extracted entities with displacy

Compare results using two different spaCy models

Natural Language Processing Tasks

Level 2

Task 5: Topic Modeling on News Articles

Description:

- Dataset (Recommended): BBC News Dataset (Kaggle)
- Discover hidden topics or themes in a collection of news articles or blog posts
- Preprocess the text: tokenization, lowercasing, stopword removal
- Apply Latent Dirichlet Allocation (LDA) to extract dominant topics
- Display the most significant words per topic

Tools & Libraries:

Python | Gensim | pyLDAvis | NLTK/spacy | Scikit-learn

Covered Topics

Topic modeling | Unsupervised NLP

Bonus:

Compare LDA vs. NMF performance

Use pyLDAvis or word clouds to visualize topic-word distributions

Natural Language Processing Tasks

Level 3

Task 6: Question Answering with Transformers

Description:

- Dataset (Recommended): SQuAD v1.1 - Stanford Question Answering (Kaggle)
- Build a system that answers questions based on a given context or passage
- Use pre-trained transformer models (e.g., BERT or DistilBERT) fine-tuned for question answering
- Feed the model both the context and the question, and extract the correct answer span
- Evaluate with exact match and F1 score

Tools & Libraries:

Hugging Face Transformers | Tokenizers | Pandas

Covered Topics

Question answering | Span extraction | Transformer-based NLP

Bonus:

Try different base models (e.g., BERT, RoBERTa, ALBERT) and compare performance

Build a simple command-line or Streamlit interface to input a passage and a question

Task 7: Text Summarization Using Pre-trained Models

Description:

- Dataset (Recommended): CNN-DailyMail News (Kaggle)
- Generate concise summaries from long documents using NLP models
- Use an encoder-decoder architecture (T5, BART, or Pegasus)
- Preprocess long texts and truncate to model input limits
- Evaluate summary quality using ROUGE scores

Tools & Libraries:

ROUGE | Hugging Face Transformers | Pandas

Covered Topics

Abstractive summarization | NLP with deep learning

Bonus:

Try extractive summarization using TextRank or Gensim

Fine-tune a pre-trained summarizer on a custom dataset

Natural Language Processing Tasks

Industry Level

Task 8: Resume Screening Using NLP

Description:

- Dataset (Recommended): Resume Dataset + Job Dataset (you will need both) (Kaggle)
- Develop a system to screen and rank resumes based on job descriptions
- Preprocess resumes and job descriptions using embeddings
- Match resumes using cosine similarity or classification
- Present top-ranked resumes with brief justifications

Tools & Libraries:

Sentence Transformers | Pandas | Scikit-learn

Covered Topics

Document similarity | Semantic search in NLP

Bonus:

Create a simple front-end to upload a resume and get matching results

Try named entity extraction from resumes (skills, experience)

Bonus Explanation:

User uploads a resume file (usually a .txt, .pdf, or .docx)

The system reads and processes the text from the resume

It compares that resume to a job description or a set of job requirements using NLP (e.g., cosine similarity with embeddings)

The user then sees a matching score or result like:

- “Match Score: 85%”
- “Strong match for: Junior Data Analyst”
- or a list of highlighted skills that matched the job description