Named Entity Recognition (NER)

(University Admission Query System)

1. Introduction

Named Entity Recognition (NER) is a foundational component of Natural Language Processing (NLP) designed to automatically identify and classify meaningful pieces of information called *entities* within text. In practical applications, NER helps convert unstructured user input into structured data that a machine can understand. For a university admission inquiry system, NER is essential because users express their queries in an informal, ungrammatical, and highly variable style, especially when using Roman Urdu.

For example, students commonly send messages such as:
"sir admission kb open honge?",
"meray matric ke marks 850 hain, kya me BSCS apply kr sakta hn?",
"fee structure bata dein BS Psychology ka", or
"last date kab hai form submit krne ki?"

In these queries, important entities such as DATE, PROGRAM NAME, MARKS, PERSON, DEADLINE, and DOCUMENT REQUIREMENT must be accurately extracted so the AI agent can respond correctly. NER enables the system to automatically detect that *"BSCS"* belongs to the PROGRAM entity, *"850 marks"* relates to ACADEMIC_SCORE, and *"kb open honge / last date kab hai"* represent DATE/TIME-related queries. Without NER, the agent would only understand these messages as plain text and would be unable to provide correct answers or trigger the appropriate workflow.

Thus, NER plays a critical role in turning messy, user-generated Roman Urdu text into actionable knowledge for the AI inbound calling agent.

2. Dataset Requirements for Roman Urdu NER

Building an effective NER system requires a high-quality, domain-specific dataset that reflects the way real students communicate during the admission process. Unlike standard corpora used for NLP in English, Roman Urdu is highly inconsistent; there is no standard spelling, grammar, capitalization, or punctuation.

For example, students may write the same message in many different forms:

- *"admission kab start?"*
- *"admsn kb shuru hoga?"*
- *"kb sy apply krna h?"*
- *"sir form kab milay ga?"*

All of these expressions refer to the DATE/PROCESS entity. The dataset must include such variation to teach the model how Roman Urdu behaves in real-life admission conversations.

In addition, the dataset needs to be annotated at the token level, meaning each word or subword is tagged with a category such as PROGRAM, DATE, MARKS, NAME, CITY, DOCUMENT, or OTHER. A properly annotated dataset might look like:

- *"BSCS ka admission kab start hoga?"*
  → BSCS = PROGRAM
  → admission = PROCESS
  → kab = TIME
  → start hoga = OTHER

Since this project focuses on a university admission agent, the dataset must be shaped to include entities that are specific to this domain. These may include:

- **Program names** (BSCS, BBA, BS English)
- **Admission process terms** (admission, merit list, registration)
- **Dates and deadlines**
- **Fee-related expressions**
- **Student marks and grades**
- **Required documents** (cnic, result card, domicile)

3. Approaches to NER

Over the years, several computational approaches have been developed to perform NER. The earliest systems relied on **rule-based techniques**, which used manually created dictionaries, patterns, and grammar rules to detect entities. Although useful in controlled or highly formal domains, such systems quickly break down when exposed to inconsistent informal text like Roman Urdu used by students. A rule-based model cannot interpret spelling variations such as *admission/admsun/admsn/admishan*, nor can it handle code-mixing such as *"sir admission info share krden"*.

Later, **machine learning approaches** such as Support Vector Machines (SVM), Logistic Regression, Hidden Markov Models (HMM), and Conditional Random Fields (CRF) became popular. These methods require explicit feature engineering such as word shapes, prefixes, suffixes, capitalization, POS tags, and handcrafted patterns. However, Roman Urdu offers almost none of these features because there is no capitalization, no standard POS tagging system, and no fixed spelling conventions. As a result, machine learning–based NER becomes extremely limited in this domain.

Deep learning models like CNNs, LSTMs, and BiLSTM-CRF became the next stage of progress. These models learn features automatically and do not require manual engineering. However, deep learning architectures depend heavily on high-quality word embeddings (such as word2vec or fastText). Since Roman Urdu has no standard pretrained embeddings, the model struggles with unseen words, spelling variations, and transliteration differences.

Also, deep learning models require large datasets, which can be costly and time-intensive to develop.

The most advanced and currently dominant method is **Transformer-based NER**, introduced by models like BERT, RoBERTa, and XLM-RoBERTa. Transformers use a mechanism called self-attention, which allows them to analyze entire sentences at once and understand relationships between words based on context. Unlike older models, transformers do not depend on handcrafted features or rigid word embeddings—they rely on contextual embeddings that adapt to new vocabulary and sentence structure. This makes transformer-based NER far superior for multilingual, noisy, and informal text like Roman Urdu.

4. Why Transformer-Based NER Is the Best Choice for a University Admission AI Agent

A university admission inquiry system must understand complex, messy queries by students Transformer models are uniquely suited for this because they are trained on massive multilingual datasets and excel at learning patterns in unstructured text.

For example, inquiries such as:

- *“ bs english ki last date konsi hai?”*
- *“sir bscs ki fees kya hai?”*
- *“admission open hai kya fall wale?”*
- *“test kb hoga entry test ka?”*

These messages mix English, Roman Urdu, and domain-specific terms. Transformer models process these naturally, breaking the input into subwords and identifying entities even when the spelling varies.

There are several reasons why transformers, especially XLM-RoBERTa are the ideal choice for this use case.
First, these models are trained on billions of multilingual sentences, including languages that share vocabulary with Urdu, Hindi, and English. This multilingual pretraining gives the model an excellent understanding of the linguistic structure and lexical patterns commonly found in Roman Urdu.

Second, transformers use subword tokenization, which helps them interpret different spellings and noisy variations. For example, *“admission, admishan, admsn, admsion”* are all broken into understandable fragments, enabling the model to classify them correctly even if it has not seen that exact spelling before. This is essential for real student queries, where spelling inconsistency is the norm, not the exception.

Third, transformers are extremely effective at handling code-mixed language. Roman Urdu often includes both English and Urdu words, such as *“sir admission open hai?”* or *“fee structure of bba bata dein”*. Because transformers were trained on mixed datasets, they handle such content effortlessly.

Fourth, transformers do not require manual feature engineering. This is important for Roman Urdu because the language lacks capitalization, diacritics, and standardized grammar. Instead of relying on these absent features, transformers extract context-based meaning directly from the sequence, making them highly adaptable and scalable for real conversational AI.

Finally, transformer models are highly suitable for real-time AI agents because they offer extremely high accuracy with relatively small fine-tuning datasets. Even with a moderate amount of labeled Roman Urdu admission queries, a fine-tuned transformer can identify entity types with precision levels close to human annotation.

5. Why XLM-RoBERTa Is Specifically the Best Choice

Among all transformer architectures, XLM-RoBERTa stands out as the most suitable model for Roman Urdu NER in a university admission agent. It was trained on 2.5TB of text across more than 100 languages, many of which share roots, vocabulary, or structure with Urdu and Hindi. This massive multilingual exposure enables XLM-R to naturally understand the hybrid structure of Roman Urdu even before fine-tuning.

If embedded into a university admission query system, XLM-RoBERTa can recognize entities such as program names (*BSCS, BBA, BS English*), deadlines (*last date, due date, kb tk*), student marks (*900 marks, 75%*), documents (*cnic, domicile, result card*), and fee-related phrases (*fees kitni hai, per semester charges*). The model's contextual understanding also allows it to differentiate between similar sentences with different meanings, such as:

- *"bs English ki fee"* (Program → Fee inquiry)
- *"English test kb hai?"* (Subject → Test inquiry)

This level of nuance is crucial for an inbound admission system, where mistakes can lead to incorrect automated responses.