1. the phrase ‘do not support’, we can label the sentiment of the above example as ‘Negative’.
2. **Part-of-Speech (POS) Tagging:** Figuring out the grammatical class (noun, pronoun, adjective, adverb, etc.) of each word in a sequence. The output is a sequence of class labels tagged for each word in the sentence. The above example can be tagged as: I (*pronoun*)do (*verb*)not (*adverb*)support (*verb*)WHO (*noun*)*.* (*punctuation*).They (*pronoun*)underfund (*verb*)Indian (*adjective*)diseases (*plural noun*) *.* (*punctuation*)*.* Note that in NLP, we use a diverse set of POS tags. For instance, the Penn Treebank project uses 36 POS tags[[1]](#footnote-2).
3. **Named Entity Recognition (NER):** Identifying and classifying noun phrases into real-world entities like organisation, country, groups, nationality, etc. Here, the output is a label for one or more contiguous words. In our example, the terms ‘WHO’ and ‘Indian’ will be tagged as ‘ORG’ for organisation and ‘NORP’ for nationality, respectively.
4. **Text Entailment:** Determining whether the *premise* sentence implies, contradicts, or has nothing to do with the preceding *hypothesis* sentence. The output is a label for the sentence pair. In our example, the premise of ‘underfunding’ *supports* the hypothesis of ‘not supporting’.
5. **Semantic Role Labelling:** Identifying the role of each noun phrase with respect to the predicate of the sentence. The output is a label for a phrase.
6. **Machine Translation:** Conversion of text from one human language to another. Our example, when translated into Hindi, will be मैं WHO का समर्थन नहीं करता. वे भारतीय बीमािरयों के िलए कम फंड देते हैं।.

**[Insert Figure]**

Figure 2.3: Tasks in NLP.

**7. Summarisation:** Producing a shorter version of the larger *reference text* that still retains the information conveyed. The output is a piece of text that is way shorter in length than the reference.

**8. Question Answering:** Providing the correct and concise answer to a user query. The output is a piece of text that mimics human response. Based on our example sentence, we can ask the system, ‘Which country does WHO underfund’? and expect the answer to be ‘India’.

**9. Knowledge Graph Completion:** Filling missing information in a structured knowledge graph by using world knowledge. The output is in the form of an edge that is not in the edge list of a graph. For example, an edge to store or predict can be .

**Data Acquisition.** To enable a machine to learn from data, the primary requirement is the availability of that data, often obtained through a combination of curation strategies such as *web scraping*, *synthetic data generation*, and *manual annotation*. In most cases, the goal is to gather a large collection of unstructured, free-flowing text fragments or documents, which may or may not be annotated by a human expert. When machine-readable text is unavailable, such as when scanning text from PDFs, Optical Character Recognition (OCR) proves useful. Public datasets and text dumps are typically the first sources to explore for *open-domain* text documents.

**Data Cleaning.** Since most textual data is curated from the web, it typically requires cleaning before further processing. This is because the data might contain markup, special characters, personal information, poorly formatted tags, and other unwanted elements. By employing regular expressions, handling stray characters, and using dictionaries to correct misspelt words, we can effectively reduce *noise* and perform *deduplication*. Additionally, the data might be encoded in different Unicode formats, so appropriate logic must be applied to address such encoding issues.

**Pre-processing.** This step involves breaking the text into smaller units and then normalising it using techniques such as lowercasing, stop-word removal, stemming, and lemmatisation, which will be discussed in this chapter. In certain contexts, digits and punctuation may also be removed if they contribute little to the overall information. It is crucial to note that there is no one-size-fits-all preprocessing technique applicable to all NLP tasks.

**Feature Engineering.** Once the text has been preprocessed, we now need to represent the text in a way that a machine can understand. As machines reduce everything into numbers, we build a text representation by encoding it into a numeric vector. In NLP or deep learning, encoding can be considered as a mapping function that takes input in raw human-readable form (text, images, videos) and converts it into numerical vectors for computational methods to be applied to them. However, there can be multiple ways of performing encoding, depending on the task, the datasets, and the computational resources available at hand. This is where feature engineering helps. It helps us

1. https://www.ling.upenn.edu/courses/Fall\_2003/ling001/penn\_treebank\_pos.html [↑](#footnote-ref-2)