

Learning and Understanding Concepts of Core Natural Language Processing (NLP) Techniques Through the Microsoft Learn Platform

Jomari Manto
College of Computer Studies & Engineering
Jose Rizal University
Mandaluyong City, Philippines
jomari.manto@my.jru.edu

Abstract—This academic reflection documents the study of core natural language processing techniques through a structured self-paced learning module. The objective was to establish a foundational understanding of natural language processing as a technology, to trace how it has evolved over time, and to examine how it is applied today across business and other domains. The exercise was completed through an interactive module covering text preprocessing, statistical text analysis, and semantic language models, followed by a hands-on text analytics activity and a knowledge assessment. The material progressed from tokenization and text normalization, through statistical term weighting using term frequency and inverse document frequency, to high-dimensional vector embeddings and contextual representation. Taken together, these techniques bridge the gap between the language people use to communicate and the numerical form in which machines represent and process information, which is the foundation on which modern conversational systems are built. The exercise established that the apparent fluency of such systems rests on a sequence of preprocessing and representation decisions, each carrying trade-offs that affect the results obtained further along the pipeline. Understanding these core concepts establishes the foundation required for more advanced study within the field of natural language processing. (*Abstract*)

Keywords—*natural language processing, tokenization, term frequency-inverse document frequency, word embeddings, context vectors (keywords)*

I. INTRODUCTION

Natural language processing traces its roots to the 1950s, when the question of whether machines could think was examined by Alan Turing in what became known as the Turing Test, opening the research into machine understanding of human language that followed [1]. In the decades since, the field has moved out of the research laboratory and into ordinary use. Chatbots and virtual assistants, large language models such as ChatGPT, Claude, and Gemini, spam filtering in electronic mail, and the profanity filters built into mobile games all depend on it, as do business applications including sentiment analysis, keyword matching, and the extraction of information from unstructured records [1], [3].

That ubiquity creates a gap between use and understanding. These techniques operate beneath the surface of the products that contain them, and Oracle describes natural language processing as the cornerstone technology behind the services people rely on daily [3]. Familiarity with the applications therefore does not imply familiarity with the methods that produce them.

The author's position before this exercise was of that kind. The course readings had established what natural language

processing does and where organizations apply it [1] and had introduced the neural network structures underlying current systems [2], but neither addressed how raw text is converted into a form that a model can compute over. Industry practice treats that conversion as the first two stages of the implementation pipeline, namely collecting and preparing text data, then extracting features and representing them numerically [3]. This exercise addresses those two stages, and this report documents what was learned from it.

II. EXPERIENTIAL AND PRACTICAL INSIGHTS

The module is a self-paced course of approximately thirty minutes, of which fifteen are allocated to the hands-on exercise in Unit 5. Units 1 through 4 present the concepts and techniques, and the exercise then allows the learner to observe those techniques operating within an application [4]. The environment is a learning aid modeled on the interface of the Microsoft Foundry portal, although it invokes no cloud services. It runs Microsoft's Phi 3.5-mini locally through WebAssembly, which allows it to operate on any device without a dedicated graphics processor, and it provides a fallback option that skips local inference entirely on lower-specification machines.

The exercise itself presented little that was unfamiliar. The author uses language models of considerably greater capability in daily work, and the outputs returned by the application held no surprises at the level of use. That observation is itself informative. The model deployed here is deliberately small, and the interface exposes its constraints openly through an initial download measured in minutes, slower responses on modest hardware, and an explicit option to disable inference altogether. Oracle identifies computational cost as one of the principal current challenges in the field [3], and the design of this learning environment renders that constraint visible rather than abstract.

The value of the exercise therefore lay elsewhere. Prolonged familiarity with such applications had concealed the mechanism beneath them. Placing the four preceding units alongside a running application made it apparent that what presents itself as conversation is the product of a sequence of deliberate operations, in which text is segmented into tokens, reduced to base forms, weighted according to statistical importance, and finally represented as a position within a vector space. The fluency is an outcome of that pipeline rather than a property of the system.

Viewed from sufficient distance, the field remains an application of mathematics. Human language is converted into numerical form so that it can be operated upon, and the results are converted back into language so that a person can interpret

them. This exercise made the first of those conversions legible.

III. CORE LEARNING OUTCOMES

The techniques presented in the module are not alternatives but a chain, in which each stage constrains everything that follows. Tokenization and the normalization accompanying it determine what survives into later processing, and nothing downstream can recover what is discarded there. Statistical weighting then establishes which of the surviving terms carry significance, although it treats words as symbols and leaves equivalent meanings unrelated to one another. Vector embeddings replace that arrangement with geometry, and contextual representation makes the position of a word depend upon the sentence containing it rather than upon the word alone [3].

Considered separately, none of these operations resembles understanding. Segmenting, counting, weighting and measuring distance are mechanical procedures. What the module made clear is that the comprehension a user perceives in an application such as a large language model is a property of the composition rather than of any component within it. A question typed in ordinary words is segmented, normalized, weighted and positioned; a response is produced from those positions and rendered back into language. The user perceives a conversation, while the system has performed arithmetic upon representations.

The same structure accounts for failure. A system that misreads a negated statement has commonly lost the negation during preprocessing, and one that answers the wrong sense of an ambiguous word is working from a representation insufficiently conditioned upon context. Behavior observed at the surface is traceable to a stage within the chain, which is knowledge that familiarity with such applications does not supply.

IV. COURSE APPLICATION AND IMPACT

These concepts are introductory by design, and their value lies in what they render accessible next. The immediate progression is architectural. Zhong and Zhuang describe recurrent neural networks as retaining data entered previously in order to predict what follows [2], and Bahja records that neural machine translation conducts its entire process through such a network [1]. Beyond these lie the pre-trained contextual models that Bahja identifies, among them Bidirectional Encoder Representations from Transformers, since advanced by XLNet [1], from which current large language models descend. Each of these developments extends the representation problem examined in this module rather than replacing the preprocessing that precedes it, which is why the sequence had to be understood first.

For a student of information technology in business analytics, that progression maps onto the work ahead. Both readings establish how widely these techniques are applied, with Bahja documenting deployment across commerce, electronic governance, healthcare, education and hospitality [1], and Zhong and Zhuang across customer service, sales, marketing, daily operations and risk management [2]. What connects those cases is that each begins with unstructured text and ends with a decision. Subsequent laboratory work and the capstone project will require that path to be constructed rather than merely described, and any project drawing upon text data must account for how that text is segmented, normalized and represented before questions of modeling arise. The exercise supplies the vocabulary needed to specify such work, to evaluate the tools available for it, and to justify the resulting choices.

V. CONCLUSION AND FUTURE WORK

The principal outcome of this exercise concerned mechanism rather than application. Familiarity with natural language processing tools had not conveyed how those tools are constructed, and the module established that what presents itself as comprehension is the accumulated result of segmentation, weighting and representation, with each stage constraining what the next is able to do.

The wider significance is a matter of scope. Prior coursework completed by the author in predictive modeling addressed traditional machine learning applied to structured data. The techniques examined here extend that work into deep learning and, with it, into data that no tabular method can reach. Continued study of natural language processing and of the large language models descended from it is directed toward a single objective, namely the capacity to build analytical systems that operate upon structured, semi-structured and unstructured data alike rather than upon the structured portion alone.

REFERENCES

- [1] M. Bahja, "Natural language processing applications in business," in *E-Business: Higher Education and Intelligence Applications*. London, U.K.: IntechOpen, 2020.
- [2] Z. Zhong and X. Zhuang, "Deep learning applications in business activities," *American Journal of Management Science and Engineering*, vol. 3, no. 5, pp. 38–43, 2018.
- [3] J. Erickson, "What is natural language processing (NLP)?" Oracle, Sep. 22, 2025. [Online]. Available: <https://www.oracle.com/asean/artificial-intelligence/natural-language-processing/>
- [4] Microsoft, "Introduction to natural language processing concepts," Microsoft Learn, 2025. [Online]. Available: <https://learn.microsoft.com/en-us/training/modules/introduction-language/>