CHAPTER II
THEORY AND METHOD

2.1 Literature Review

1.1.1 Theoretical Framework

1.1.1.1 Systemic Functional Linguistic

Systemic Functional Linguistics, abbreviated as SFL, is a method of linguistic analysis that seeks to comprehend how the meaning of a text emerges from its specific context (Halliday, 2013). In the past two decades, the SFL has been broadened thanks to the efforts of numerous scholars all over the world (Achugar & Colombi, 2008). Language, as described by Halliday (1994), serves social functions, which are made visible in the development of functional grammar of modern English. Systemic functional linguistics, in particular, serves as a descriptive framework for interpreting language as a social semiotic system (Eggins 2004: 2). According to Halliday and Matthiessen (2004), context can be considered one of the core issues because it plays such an essential role in the entire process of meaning creation. In point of fact, when language is used in a particular setting, it will reference The Context of Culture and The Context of the Situation.

1.1.1.2 Transitivity System

According to Halliday, Transitivity is a system that reduces the universe of experience to a manageable set of process categories (2004:170). The English transitivity system distinguishes between six distinct types of processes: material, mental, verbal, logical, behavioral, and existential. Each process has three components: process, participant, and circumstance. To sum up, Transitivity is the study of what individuals do, mainly focusing on who does what to whom (Machin & Mayr, 2012).

The author found several studies that implement the same methods by manually recognizing, collecting, and analyzing linguistic data, such as Waridatika (2016), Asad (2019), and Hastuti (2017). The tools used in the studies are only used to help organize and display the findings.

1.1.2 Tools

1.1.2.1 UAM Corpus Tool

UAM (Universidad Autonoma de Madrid) Corpus Tool, is a text corpus annotation software that enables for manual and automatic annotation of text collections at various linguistic layers. It was created by a computational linguist, Mick O'Donnell, who discovered that while many annotation tools have been created, they are not easily adaptable to varied annotation issues. They have been constrained to some extent in that they only enable specific forms of annotation to occur. (O’Donnell, 2008).

The UAM Corpus Tool was created to address a gap in automatic annotation, which is often done by 'Human-Annotate' or the traditional method. It allows the user to apply tags to portions of the text and assign them to the tag hierarchy based on the user's needs and preferences. This tool can be used for bulk processing, also known as multiple file processing with the same annotation system. Users can also add many layers depending on the type of linguistic analysis they prefer to implement, such as semantic-pragmatic, clause, phrases, or at a lexical level. However, UAM has a restriction on the size of text that can be processed. At certain point, we'll need to break a large text to smaller chunks, or reduce the content.

The author discovered a number of studies that have already implemented UAM as the corpus analysis tool, such as Bhatti, Azher, and Abbas (2019), Ammara and Anjum (2019), Haroon and Arslan (2021), and Iqbal (2023). The UAM corpus tool is used in these studies to help annotate the transitivity analysis.

1.1.2.2 CQPweb

According to Hardie (2012) CQPweb is a web-based graphical user interface (GUI) for several CWB parts, specifically the CQP query processor. CQPweb is intended to mimic the user interface of the well-known BNCweb tool, which also uses CQP as a back-end. CQPweb, like BNCweb, employs a database alongside the CWB to provide additional functionality beyond those integrated into the CWB/CQP. CQPweb, on the other hand, can be used with any corpus, unlike BNCweb.

CQPweb is ideal for students, non-linguists, and anyone who are intimidated by a Unix-like command-line. (Hardie, 2012). As a corpus analysis tool, CQPweb provides several advantages. It offers a simple yet powerful query language. The technology is capable of indexing any corpus. It offers a user-friendly interface that allows any user to utilize it with only little adaptions. The technique is scalable, allowing it to cope with corpora containing hundreds of millions of words. CQPweb has some limitations as well. On a 64-bit system, the maximum corpus size is 2.1 billion words.

Despite its ability to deal with large-scale files, CQPweb is an indexer tool that does not include any analytic tools. At present, a number of third-party software have been added to CQPweb, most of which are TreeTagger variants (Schind, 1994:1995). However, UAM has not been acquired due to the merging complexities.

The author discovered several studies utilizing CQPweb as their indexing tool, such as Luo and Liao (2015), and Tootalian and Jacob (2017). Thus, in order to do transitivity analysis on large-scale files, (1) it is necessary to have the files been analyzed using UAM, (2) adapt the format so that it is readable by CQPweb, and (3) index the files to CQPweb.

However, it might be challenging for a linguist without any programming background to convert the format. In this research, the author offers a ready to use program that allows non-technical users without a background in computers to automatically convert UAM analysis to CQPweb-readable format.

1.2 Research Methodology

To provide solutions to the issues with the research that were brought up in the first chapter. The author will discuss the process in developing the converter. This study begins with collecting the text that will serve as the primary object of analysis for the transitivity process. It consists of ten contemporary novels that have different publication year, author, and, publisher. Three chapters from every novel used to populate the data set, so that the data sample will consist large of text written from various perspectives.

The following step is doing the automatic analysis with UAM and retrieve the output. Then, inspect the output structure and compare it to the CQPweb-readable format. After understanding the information from the output, next step is developing the converter. The author creates the converter step by step, and discuss together with thesis advisor. This step is called trial and error. The goal is to restructure the UAM output to match with CQPweb format. After the converter is developed, convert every UAM output, do the xml format checking, and populate the metadata (information about the data such as authors, years of publication, publisher, and etc.) that will be used as a filter in CQPweb. Once completed the data & metadata are indexed to CQPweb, the final step is to perform SFL analysis through the CQPweb, using CQPweb full functionalitites.