

# Lexical Analysis of Computer Studies Terms Translation from English to Filipino Language

**Elimar Alupay Ravina**

Language Learning Area, University of Santo Tomas, Manila, Philippine.

**How to cite this paper:** Elimar Alupay Ravina. (2022) Lexical Analysis of Computer Studies Terms Translation from English to Filipino Language. *Journal of Humanities, Arts and Social Science*, 6(1), 113-128.

DOI: 10.26855/jhass.2022.01.012

**Received:** February 3, 2022

**Accepted:** February 27, 2022

**Published:** March 14, 2022

\***Corresponding author:** Elimar Alupay Ravina, Language Learning Area, University of Santo Tomas, Manila, Philippine.

**Email:** earavina@ust.edu.ph;  
elimar\_a\_ravina@yahoo.com

---

## Abstract

This study analyzed 530 terms in computer studies based on their lexical classifications namely lexical type, lexical category, and lexical formation. These were from printed books, journals, documents, and academic discourses of students and professors in the field. The terms were analyzed to find the relationship between the lexical classification and translation techniques used. The analysis revealed that the terms are either simple or complex (lexical type), and they can be categorized as nouns, adjectives and verbs (lexical category). The noun sub-categories were common words, brands or trademarks, and keywords. The terms were lexically formed by derivation, blending, compounding, clipping, and simple root word. From this, structural characteristics of the terms were revealed. In the translation techniques used, simple terms were found to be translated by purification, systematic borrowing, lexical convergence, and semantic translation. Complex terms were translated by borrowing, owning, or adoption. The analysis further revealed that the lexical classifications and other structural characteristics had no relationship with the translation techniques used. Thus, there was no translation decision patterns found. This proves the inconsistencies observed in translating the terms, and it signifies the absence of clear standards on how English terms in computer studies are translated in Filipino.

## Keywords

Computer Studies Terms, Lexical Analysis, Lexical Category, Lexical Formation, Lexical Type, Translation, Translation Technique

---

## 1. Introduction

English, Filipino, and other Philippine languages are all mediums of instruction in the Philippine educational system (Sicat, Zafra, & Geronimo, 2017). This language policy in the implemented K-12 education system eventually effected the revisions of curriculums in the higher education and neutralized the dominance of English across the system (Sicat, 2019). In the pursuit of intellectualizing the Filipino language in order to be at par with the use of English language in the different academic disciplines, the Commission on Filipino Language conducted a series of academic seminars for the higher learning institution administrators and academics (KWF, 2016). This endeavor led to the certification programs of translators and the translation of documents, books, and other essential texts in different disciplines from English to Filipino.

However, the approaches in translating English terms into the Filipino language as stated in the principles expounded by Santos in 1951, Del Rosario in 1969, Paz in 1995, Alcantara in 2006, and Enriquez and Marcelino in 1984 (Batnag, Petras, & Antonio, 2009) cannot be taken as standards because they are not consistent with one another. The absence of standard guide created the problem in the translation process. The principles propagated by

the aforementioned proponents reflect the guide given by the Commonwealth Act No. 184, Sec.8 Par. 2 (in Santiago, 1984). In the translation work on Science textbooks of the Philippine Normal College-Language Study Center, these popular principles were followed. These translation principles cited in the study *The Technical Lexicon of Filipino* (Santiago, 1984), include the following: a) using Tagalog Terms; b) borrowing from English; c) borrowing from Spanish; and d) using the *maugnayin principles*. In the more recent translation techniques used by prominent translators like Almario (1997) and Sevilla (2001), almost the same principles were followed but additional specifications like borrowing from English but respelled in Filipino, and creating new words were added. These principles were only assumed to be most suited guidelines, but there were no explanations to what linguistic structures these were based from.

In the analysis of different texts in computer studies that were written in Filipino language, it was found out that different translation techniques were used (Alviar & Roxas, 2000; Macatangay & Macatangay, 1998; De Castro, 2008; Madrid, 2005). This finding prompted the researcher to investigate further on this phenomenon. Thus, this paper sought to analyze lexically the computer studies terms and the techniques used in translating them. Specifically, this study aimed to:

- 1) identify *terms* in computer studies that were used in Filipino academic discourses;
- 2) analyze the *lexical classification* of the *back translated terms* according to their:
  - a. *lexical type*,
  - b. *lexical formation*, and
  - c. *lexical category*; and
- 3) analyze the translation techniques used on the terms considering their *lexical classification*.

## 2. Literature Review

The conduct of this analysis aims to contribute to the identification of standard translation techniques based on the linguistic characteristics of words. Furthermore, there is a need for this study in the pursuit of intellectualizing the Filipino language; thus, balancing the hegemony of English language in the neo-colonial Philippine education system (Ferguson, 2006). In the field of computer studies, this study is hoped to contribute to the identification of linguistic structure-based decision in adopting a translation technique. This is important for several reasons:

Firstly, *the Filipino language is an intellectual language that can be used as medium in technical discourses*. This is in refute of the popular notion that only the English language is capable of delivering technical information, but as Gonzalez (2005) posited, the breadth and depth of the intellectualization of people is the obligation of the state and the academic community that use a language- in this case the Filipino language. And in as long as an intellectual discipline has not rooted in that society, the language of that discipline may not be cultivated (p. 235).

Secondly, *Filipino is a modern and global language*. This means that this language, like any other major languages of the world like English, German, Mandarin, Spanish or French, has its available counterpart words for new and technical concepts like that of the computer technology. This coincides with the statement of Zafra (2003) that the Filipino language because of its corpus inter-translatability has the capacity to be at par with languages that are modern, intellectualized, and standardized like English.

Thirdly, *the Filipino language is a lingua franca; thus, it is useful in enriching the academic and professional skills of the learners*. This is supported by Tauli (1968) who claimed that, "...because language is an instrument; therefore, it can be assessed, altered, engineered, controlled, and developed. Thus, a new form of that language can be planned." When the lingua franca is used as the language of education, people can efficiently develop their skills better.

Fourthly, *the Filipino language should be used in computer studies, because it is mandated by the law*. The DECS Order No. 54 s. 1987 which is commonly known as the *Implementing Guidelines for the 1987 Policy on Bilingual Education* intended not only to make Filipino as a medium of instruction, but as a medium of intellectual discourses and scholarship (Fortunato, 2003, pp. 82-83). The use of the common language to more efficiently master a content was further strengthened by researches which became the basis of the Department of Education Order 16, s. 2012 that mandates the implementation of the *Mother Tongue-Based Multilingual Education* or MTB-MLE in Philippine schools (Sicat, Zafra, & Geronimo, 2017).

Lastly, the clear guidelines in translating *the terms used in the field of computer studies into Filipino will, thereby, help promote the intellectualization of the language*. This was based on the idea that the availability of parallel vocabularies that equate to the concepts in the field makes it easier for writers and translators to give equivalent terms in a way that is consistent among each other.

Students' use of identity language in a discipline like computer studies is more effective in their learning (Benson, 2004), and having that language produce its pedagogical idioms and technical terms in that language is a must. This can be achieved with translation. The learners' utilization of their own language, a language that is well-understood, promotes high and deep knowledge of a discipline. This can be seen in the context of educational systems that promote student's use of native language in higher education like Russia, Germany, Japan, and South Korea (Ravina, 2017). Russia, for instance, has an advanced knowledge of space technology, and Germany is known in transport engineering ("Airbus Defense and Space", 2015; "Germany Trade and Invest", 2015). In the article *Russia Remains Leader in Space Launches*, Russia was deemed to be the world's leader in space science because it remains to have the most number of explorations and experiments in this field (2015). On the other hand, GTAI published a report entitled *Industry Overview: the Automotive Industry in Germany* which claims that the country has remained to be the number one producer of automobiles in the last 125 years (2015). In the Asian context, Japan is known to robotics and computer technology while South Korea prides itself in the fields of medical technology and in mobile computing technology. *The New York Times* reported that *K Computer*, a Japan-made computer, is the most powerful computer product. Furthermore, it reported that the country's computer products remained to be the best in the global market in terms of its quality (Kopytoff, 2011). The same scenario was observed in South Korea, as emphasized in the article *Why South Korea will Always be the Leader in Mobile Technology*, as the strongest and the best when it comes to mobile technology (David, 2015). On the same manner, the county is also the leader in the production of quality device and in the actualization of processes in the field of medical technology ("Quality Medical Education", 2013).

These experiences of countries with the education system's language policy favorable on their own language have direct impact on their technical and scientific knowledge production. And in the case of the Philippines, the wider use of the lingua franca of the people as medium of instruction and intellectual discourses in higher education may contribute to a similar knowledge production state. But since the historical context of the Philippines is different as to those previously mentioned, translation of texts in the English dominated fields can be a jumpstart into this state. This is highly possible if there is a standard guide that will make translation of terms consistent.

### 3. Research Methodology

#### 3.1. Research Design

This research used descriptive-analytical design. It generally described the state of English and Filipino as languages in computer studies discourses. Also, it involved micro analysis of the terms focusing on the linguistic features, and the specific techniques of translation applied on to them. Furthermore, this research used triangulated data sources and analysis for a more credible data representation (Bryman, 2010).

#### 3.2. Participants

The participants of this study were the data source and the data analyst. The data source included students and professors in the field of computer studies. They were additional data sources aside from the pile of written materials such as books, documents, and journals in the field. The data analyst composed of professors, practitioners, and language experts. They were chosen based on the *purposive sampling* technique, and the technique considered context inclusion, expertise and professional experience as the criteria for selection.

The participants are specifically described as:

##### 3.2.1. Data Source

- a) Eighteen (18) students were chosen purposively. These students must be taking any program from the area of computer studies. They must be enrolled in any higher education learning institution that specializes in computer studies or that have been declared as Center of Development (COD) or Center of Excellence (COE) in computer studies by the Commission on Higher Education (CHED).
- b) Fifteen (15) professors in the field of computer studies who used or allowed the use of Filipino language in their classes in a CHED identified COD, COE, or specialized institutions were chosen.

##### 3.2.2. Data Analyst

- a) Two (2) professors in computer studies and one industry practitioner served as experts to validate the collected data which were the computer studies terms.
- b) One (1) technical translator who holds a doctoral degree in translation studies validated the back translation of the terms being studied.

- c) Three (3) professors of English language were chosen to serve as counter-checker/reviewer of the lexical analysis of the data.

### 3.3. Instrumentation

In the data gathering process, these instruments were utilized:

- a) *Observation sheet*: This was where the researcher recorded the data he gathered from observing classes in computer studies.
- b) *Questionnaire*: This is a survey questionnaire that was used to gather computer studies terms in Filipino from both professors and students. It also contained a list of English jargons in the field that was translated by the participants in Filipino.
- c) *Literature survey note*: This is a form similar to the *observation sheet* where the terms culled from printed sources such as books, journals, and other documents were recorded.

### 3.4. Data Gathering Procedure

In gathering the data (computer studies terms) needed for this study, the following procedures were done:

- 1) Professors who use or allow the use of Filipino language in their computer studies classes were identified and were formally requested to be observed from their classes. Terms used in the observed classes both by professors and by students were recorded using the observation sheet.
- 2) The professors and selected students were asked to answer the questionnaires. This was intended to collect more terms that were used in their classes and to know their translations of the terms listed in the questionnaire. Small group discussions were also done to discuss their answers in the questionnaires.
- 3) Resource materials in computer studies written in Filipino language such as journals, books, theses and the like were identified and surveyed. From these materials, computer studies terms were collected and recorded using the literature survey notes.
- 4) A preliminary list of terms was developed which were based from the observation sheets, questionnaires, and literature survey notes.

### 3.5. Data Analysis Methods

The methods that follow were done to analyze the data gathered.

- 1) The preliminary list of terms that contained the computer studies terms was revalidated separately by two professors and an industry practitioner. Revalidation was a process by which the terms were checked if they could be considered as (technical) terms in computer studies. In this process, 530 terms were included in the final list of terms.
- 2) The terms in the final list were back translated in English by the researcher since he is both a language and computer science professional and practicing translator at the same time. This process produced the list of back translated terms in English.
- 3) The back translated terms were validated by a technical translator. The translation was corrected, refined, and finalized. The validation was done alone by the technical translator expert, but transmitted the refined list with annotations and comments. From these comments, refinements were applied.
- 4) The validated back translated terms were analyzed lexically by the researcher and reviewed by three English language professors who acted as expert panel. The analysis was done by grouping the terms according to their classifications: lexical category, lexical formation, and lexical type (Paz, Hernandez, & Peneyra, 2010, pp. 64-69; Yule, 2003, pp. 63-70; Santiago & Tiangco, 2003, p. 125). The results were discussed in a focus group discussion with the panel of experts.
- 5) Based from the lexical analysis, the terms were analyzed and grouped accordingly based on the translation techniques of the proposed guidelines of Santos, del Rosario, Sytangco, Paz, Ramos, Alcantara, and Enriquez and Marcelino (in Batnag, Antonio, & Petras, 2009, pp. 253-254). This was to analyze the relationship between the lexical characteristics of the terms and the techniques used in translation from which generalizations and conclusions were made.

## 4. Results and Discussion

There were 530 terms gathered from the data sources of this study. This was a reduced count after the validation

of the listed terms was done by experts composed of two professors in computer studies and an industry practitioner. The validated list became the basis of the three processes done: back translation, lexical analysis, and translation technique analysis.

#### 4.1. Analysis of the Lexical Type and Lexical Category of the Terms

The tables that follow show the samples from the 530 terms presented in their back translated form (English) and their form in Filipino discourses, respectively. They were organized according to the lexical type and lexical category of the words. *Lexical type* is a lexical classification that identifies whether the term is *simple* or *complex*. Simple terms are terms taken as one word, while complex terms are terms composed of more than one word. Lexical category, on the other hand, is the grouping that classifies the term as noun, verb, or adjective (Paz, Hernandez, & Peneyra, 2010; Yule, 2003; Santiago & Tiangco, 2003).

##### 4.1.1. Simple Terms

The lexical type analysis revealed that there were 278 simple terms out of the 530 terms in the list. These terms were found out to be English content words specifically nouns, verbs, and adjectives. The word frequency revealed that there were 207 nouns, 34 verbs, and 37 adjectives. Samples terms from this finding are reflected in the following tables.

##### Nouns

**Table 1. Samples of simple terms with lexical category noun and with sub-category common words**

| <i>Common Words</i>                |   |
|------------------------------------|---|
| 1. account: account                | 11. prompt: prompt                            |
| 2. adapter: adapter                | 12. pseudo-code: pseudocode                   |
| 3. addition: adisyon               | 13. record: record, rekord, tala              |
| 4. address: address                | 14. recursion: pag-uulit                      |
| 5. algorithm: algorithm, algoritmo | 15. register: register                        |
| 6. loader: loader                  | 16. subscript: paglalagay ng pananda sa ibaba |
| 7. loop: loop, pag-uulit           | 17. subtract: subtraksiyon, pagbabawas        |
| 8. manipulation: manipulasyon      | 18. symbol: simbolo                           |
| 9. memory: memory, memorya         | 19. syntax: syntax, sintaktiko                |
| 10. microchip: microchip           | 20. system: system, sistema                   |

**Table 2. Samples of simple terms with lexical category noun and with sub-category keywords**

| <i>Keywords</i> |                |
|-----------------|----------------|
| 1. div          | 11. dir        |
| 2. mod          | 12. char       |
| 3. xor          | 13. del        |
| 4. and          | 14. dir        |
| 5. or           | 15. cls        |
| 6. boolean      | 16. mkdr       |
| 7. ABS          | 17. not: hindi |
| 8. cd, chdir    | 18. integer    |
| 9. char         | 19. LEFT\$()   |
| 10. del         | 20. LEN()      |

**Table 3. Samples of simple terms with lexical category nouns and with sub-category brand and trademark**

| <i>Brand or Trademark</i> |                |
|---------------------------|----------------|
| 1. Apple Macintosh        | 7. IBM         |
| 2. Basic                  | 8. Intel       |
| 3. Cobol                  | 9. iOS         |
| 4. Facebook               | 10. Lotus      |
| 5. Fortran                | 11. Pascal     |
| 6. Google                 | 12. PowerPoint |

**Verbs**

**Table 4. Samples of simple terms with lexical category verb**

| <i>Verbs</i>                                      |   |
|---|---|
| 1. execute: mag-execute, paganahin                | 10. italicize: paggamit ng italiko                                      |
| 2. initialize: i-initialize                       | 11. load: mag-load  |
| 3. input: i-input, nag-input                      | 12. merge: mapagsama  |
| 4. install: install, i-install                    | 13. overwrite: i-overwrite  |
| 5. italicize: paggamit ng italiko                 | 14. print: ilimbag, maimprenta, i-print, mag-print                      |
| 6. load: mag-load                                 | 15. programming: programming, pagprograma, i-program, pagbuo ng program |
| 7. merge: mapagsama                               | 16. reactivate: i-reactivate  |
| 8. overwrite: i-overwrite                         | 17. retrieve: nababawi  |
| 9. print: ilimbag, maimprenta, i-print, mag-print | 18. save: naitatago, isine-save, mag-save                               |

**Adjectives**

**Table 5. Samples of simple terms with lexical category adjectives**

| <i>Adjectives</i>                      |  |
|--|--|
| 1. binary: binary                      | 11. real-time: real-time, walang takdang paggamit  |
| 2. Boolean: Boolean                    | 12. recursive: recursive                           |
| 3. character: character                | 13. reliable: maaasahan                            |
| 4. compatible: compatible              | 14. systematic: sistematiko                        |
| 5. cryptic: cryptic                    | 15. string: string                                 |
| 6. decimal: decimal                    | 16. timesharing: timesharing, salo-salong paggamit |
| 7. effective: epektibo                 | 17. touch-screen: touch-screen                     |
| 8. efficient: mahusay                  | 18. true: tama                                     |
| 9. electronic: electronic, elektroniko | 19. user-friendly: user-friendly                   |
| 10. false: mali                        | 20. volatile: volatile                             |

The data revealed that simple terms under the category noun do still have sub-categories namely common words (207), keywords (26), and brands or trademarks (21). Verbs and adjectives had no other sub-categories identified.

#### 4.1.2. Complex Terms

Terms that were identified in the lexical type complex had 252 frequency counts. Looking at their lexical composition, each of the term's component words can be grouped according to how simple terms are categorized. Moreover, the analysis of the complex terms revealed two distinct forms namely *acronyms* and *non-acronyms*.

Complex terms in the form of *acronyms* are the terms in computer studies that are commonly used with the initial letter of the component words or the combination of several letters of the words. The *non-acronyms* on the other hand are words used in their full form and do not create coined word out of the initial letters of the component words.

Samples of the complex terms are shown in the tables that follow.

#### *Acronyms*

**Table 6. Samples of complex terms in acronym form**

| <i>Acronyms</i>   |  |
|---|--|
| 1. Base Pointer: BP, base pointer   | 11. Computer Science: CS, computer science, agham computer   |
| 2. Base Register: BX, base register   | 12. Personal Computer: PC, personal computer, personal kompyuter                                   |
| 3. Basic Input-Output System: BIOS, basic input-output system                             | 13. Point of Sale: POS, point of sale  |
| 4. Binary Digit: Bit, binary digit  | 14. Program Counter: PC, program counter   |
| 5. Bus Unit: BU, bus unit   | 15. Read Only Memory: ROM, read only memory  |
| 6. Central Processing Unit: CPU, central processing unit                                  | 16. Source Index: SI, source index   |
| 7. Code Segment: CS, code segment   | 17. Stack Segment: SS, stack segment   |
| 8. Compact Disc: CD, compact disc   | 18. Systems Analysis and Design: SAD, systems analysis and design                                  |
| 9. Complementary Metal-Oxide Semiconductor: CMOS, complementary metal-oxide semiconductor | 19. Universal Asynchronous Receiver Transmitter: UART, universal asynchronous receiver transmitter |
| 10. Computer-Aided Design: CAD, computer-aided design                                     | 20. Universal Automatic Computer: UNIVAC, universal automatic computer                             |
|   | 21. Video Graphic Adapter: VGA, video graphic adapter  |

#### *Non-Acronyms*

**Table 7. Samples of complex terms in non-acronym form**

| <i>Non-Acronyms</i>                                |  |
|--|--|
| 1. array element: element ng array                 | 14. temporary storage: pansamantalang lalagyan   |
| 2. assembly language: assembly language            | 15. terminal symbol: simula/katapusan na simbolo |
| 3. basic command: basic command                    | 16. time complexity: time complexity             |
| 4. binary system: binary system, binary na sistema | 17. toggle key: toggle key (caps lock, num lock) |
| 5. built-in application: built-in application      | 18. vacuum tube: vacuum tube                     |
| 6. case command: case command                      | 19. variable declaration: variable declaration   |
| 7. case statement: case statement                  | 20. video call: video call                       |
| 8. close function: close function                  | 21. video card: video card                       |
| 9. compound statement: compound statement          | 22. web design: pagdisenyo ng websayt            |
| 10. computer circuit: circuit ng computer          | 23. while-do command: while-do command           |
| 11. computer games: computer games                 | 24. white space: puting espasyo                  |
| 12. computer language: computer language           | 25. word processing: word processing             |
| 13. computer memory: memory ng computer            | 26. word processor: word processors              |

Tables 6 and 7 show samples of acronyms (21 out of 59 terms), and of non-acronyms (26 out of 193 terms) respectively. The two identified forms are vital, because they can be considered when deciding to translate these terms. The complex terms were analyzed differently from the simple terms, because they have these two distinct forms; however, individual component words of the complex terms could still be looked at in the way simple terms were lexically analyzed.

## 4.2. Analysis of the Lexical Formation of the Terms

The simple terms and the component words of the complex terms can be analyzed based on how they are formed. The field of linguistics provides several word formation processes that include acronymy, blending, compounding, derivation, clipping, and simple root word, and from which the lexical formation classifications were derived. Derivation is a process that uses affixation to create a new word. Clipping, on the other hand, is a formation that shortens the word by removing some syllable or letters. This can be associated with the processes of elision. Simple root words refer to words in its base form, and from root words or from stems, combining two of these is known as compounding. As oppose to compounding, blending forms words by taking syllables and/or letters/sounds from different words. This is in contrast with acronymy that makes use of the initial letter of the component words to form a new word. (Paz, Hernandez, & Peneyra, 2010; Yule, 2003).

### 4.2.1. Simple Terms

Analysis of the simple terms in relation to their lexical category, as validated by three professors of English language, revealed that all lexical categories may be formed from any of the lexical formations mentioned. Derivation, blending, compounding, clipping, and simple root word were the lexical formations that contained structural features that may be considered in analyzing the translation techniques applied on to the terms. The other formations found like back formation, borrowing, or conversion can still be classified in one of the previously stated lexical formations. Acronymy was disregarded as this was a form for complex terms.

There were several findings as to the analysis of the simple terms lexical category in relation to their lexical formations. These are summed up as follow:

#### *Nouns*

- a) For *common word nouns* formed from derivation, there are four structural characteristics found: verb + suffix, verb + -ing, adjective + suffix, and prefix + root + suffix. Examples of these are *assembler*, *encoding*, *portability*, and *debugger* respectively.
- b) For *common word nouns* formed by compounding, blending, or simple root words, they can be characterized based from their orthographical and phonological structures: term which every letter has equivalent letter or sound in Filipino (letter-to-letter correspondence); term with unsounded letter (silent sounds); term with no silent sound except for double letters, and the sound of the double letter reciprocates Filipino sound; and terms with complex sounds. Sample terms for these include *port*, *firmware*, *loop*, and *algorithm* respectively.

The only nouns being referred here are the common word nouns, because the other sub-categories are brands and keywords and in the context of computer studies keywords are special words, and in usage brands are proper nouns. This makes the case of the two sub-categories special. Their English forms were kept in the translated texts, thus no further lexical analysis is necessary.

#### *Adjectives*

- a) For *adjectives* formed by derivation, they are structurally characterized as follow: verb adjective, root + suffix, prefix + root, and prefix + root + suffix: Sample terms for these structures are *nested*, *cryptic*, *non-volatile*, and *monochromatic* respectively.
- b) For *adjectives* formed by compounding or just simple root words, the following was found: noun adjective, and natural adjective. Examples for these are *string*, and *user-friendly*.

Based from the data, there are no adjectives formed by blending and clipping.

#### *Verbs*

- a) For *verbs* formed by derivation, their structures are as follow: noun + suffix, prefix + verb, prefix + root + suffix, and verb + inflection. Sample terms for these structures are *initialize*, *debug*, *reactivate*, and *programming* respectively.
- b) For *verbs* formed by compounding and blending, there were no special structural or lexical characteristics found.

The analysis shows that there are structures and characteristics of these words that are common to the terms considering their lexical categories and lexical formations. These can be used as bases if the identified characteristics and structure were considered by writers, translators, professors, and students who in their own ways translate

terms in Filipino as they engage in different discourses in the field of computer studies.

#### 4.2.2. Complex Terms

It was previously mentioned that the component words of complex terms can be analyzed in the same ways as the simple terms. Thus, the complex terms will be looked at not on their structures but on their existing forms namely acronyms and non-acronyms.

As regard to how they were used in Filipino discourses, most of the acronyms were used in their original forms in English. Of the 59 identified terms under acronyms, only two had a translation that is not their English forms. One was *Computer Science* (CS) that in some cases were translated as *agham kompyuter*, and the other one was *personalcomputer* (PC) with translation as *personal kompyuter*.

Looking at the lexical type, lexical formation, and lexical category of the terms, generally, they are used in the discourses only as English terms (*boot, octal*), only as Filipino terms (*element, condition*), or in very few instances as Filipino or English (*program, memory*). All of these findings led to analyzing the translation techniques in relation to the findings of the lexical analysis.

#### 4.3. Analysis of the Term Translation Techniques in Filipino Discourses

Newmark (1988) defines translation as an, "... attempt to replace a statement in one language by the same statement in another language." In the context of translating a term, it is giving the most reliable equivalent word of that term in a foreign language in the target language, and in this study are the English terms in computer studies with their translation in Filipino language.

The data revealed that in Filipino discourses, a term was generally translated using the following ways: a) using only the original term in English, b) using only the Filipino equivalent of the term, and c) using the original term in English and the Filipino equivalent of it. This analysis focuses on the relation of the translation techniques used, and of the lexical type, lexical category, lexical formation, and a great deal of consideration on the identified characteristics and structures of the terms.

The sample data revealed four translation techniques used: a) purification (*ambag-dalisay*), b) systematic borrowing, c) lexical convergence, and d) semantic translation. *Ambag-dalisay* refers to giving computer studies terms equivalent word in the Philippine languages most often in Tagalog. The term *ambag* or contribution negates the idea of borrowing as this is a share of the other Philippine languages to the Filipino national language. *Systematic borrowing*, on the other hand, refers to a clear manner of loaning word. This process involved *direct borrowing* (*full borrowing*), and *nativisation* (*Anglicism, Hispanism*). *Direct borrowing* or *full borrowing* refers to the use of the term without any changes. *Nativisation* is a kind of borrowing indirectly as it may adopt a term in computer studies, but respelled based on its pronunciation in Filipino (Anglicism), or assimilated in Spanish and written in Filipino (Hispanism). *Lexical convergence* as another technique refers to mixing Filipino and English in the translated terms. And, semantic translation happens when the term is translated using the contextual meaning of it. (Santos, 1951; del Rosario, 1969; Sytangco, 1977; Paz, 1995; Ramos, 2000; Batnag, Antonio, & Petras, 2009).

The tables that follow give a glimpse of how the terms were organized considering their lexical classification and the applied translation techniques.

Looking at these data, nouns are variedly translated as all techniques were used in the process. There is no adjective term translated through lexical convergence, but there are some forms of adjective in computer studies which are not found in the data that can be translated using this technique. Example of these are adjectives with construction verb + -able like *printable* and *installable* which can be translated in Filipino as *napi-print*, and *nai-install* respectively. As to the verbs, the record shows that verbs were not translated semantically or with the use of Anglicism. Most of the verbs were prefixed with Filipino affixes, thus making lexical convergence the most commonly used technique in this lexical category. It is not surprising, too, to see nouns sub-categorized as trademarks or brands, and keywords make use of English only translation or full borrowing as these cannot be altered or changed in their disciplinary context.

Considering the lexical type, lexical category, and even the lexical formation of the terms, there is no evident pattern associated to these three factors that suggest translation decision making. However, it might be possible that looking at the lexical type and lexical category and considering the different structural characteristics of the terms previously identified, clearer translation guidelines can be made.

**Table 8. Translation techniques in nouns**

| Lexical Type:<br><b>Simple</b>    | Back Translated<br>Terms in Computer<br>Studies   | Translation Techniques  |   |   |   |  | Lexical Conver-<br>gence   | Semantic<br>Translation |
|-----------------------------------|---|---|---|---|---|--|--|-------------------------|
|                                   |   | <i>Purification/<br/>Ambag-Dalisay</i>  | Systematic Borrowing  |   |   |  |  |                         |
|                                   |   |   | Direct  | Nativisation  |   |  |  |                         |
| Lexical Cate-<br>gory <b>NOUN</b> |   |   | Full Borrowing  | Anglicism   | Hispanism   |  |  |                         |
| Common<br>Words                   | <p><b>Derivation</b></p> <ol style="list-style-type: none"> <li>1.animation</li> <li>2.automation</li> <li>3.adapter</li> <li>4.assembler</li> <li>5.controller</li> <li>6.editor</li> <li>7.identifier</li> <li>8.interpreter</li> <li>9.linker</li> <li>10.loader</li> <li>11.scanner</li> <li>12.scheduler,</li> <li>13.speaker</li> <li>14.processor,</li> <li>15.graphics</li> <li>16.signal</li> <li>17.encoding</li> <li>18.computability</li> <li>19.portability</li> <li>20.reliability</li> <li>21.scalability</li> <li>22.option</li> <li>23.compiler</li> <li>24.debugger</li> <li>25.disassembler</li> <li>26.addition</li> <li>27.compilation</li> <li>28.conversion</li> <li>29.decision,</li> <li>30.declaration</li> <li>31.division</li> <li>32.expression</li> <li>33.information,</li> <li>34.instruction</li> <li>35.manipulation</li> <li>36.multiplication</li> <li>37.connector</li> <li>38.maintenance</li> <li>39.procedure</li> <li>40.modeling</li> <li>41.testing</li> <li>42.security</li> <li>43.application</li> <li>44.recursion</li> <li>45.computer</li> <li>46.printer</li> <li>47.programmer</li> <li>48.translator</li> <li>49.antivirus</li> <li>50.interface</li> <li>51.microchip</li> <li>52.microcircuit</li> <li>53.micro-computer</li> <li>54.micro-processor,</li> <li>55.pseudocode</li> <li>56.submenu</li> <li>57.subprogram</li> <li>58.subroutine</li> <li>59.subscript</li> <li>60.superscript</li> </ol> | <ol style="list-style-type: none"> <li>1. tuntunin</li> <li>2. pandugtong</li> <li>3.pagpapanatili</li> <li>4 .katiwasayan</li> <li>5.paggagamitan</li> <li>6. limbagan</li> <li>7 .tagapagsalin</li> </ol> | <ol style="list-style-type: none"> <li>1.animation</li> <li>2.automation</li> <li>3.adapter</li> <li>4.assembler</li> <li>5.controller</li> <li>6.editor</li> <li>7.identifier</li> <li>8.interpreter</li> <li>9.linker</li> <li>10.loader</li> <li>11.scanner</li> <li>12.scheduler</li> <li>13.speaker</li> <li>14.processor</li> <li>15.graphics</li> <li>16.signal</li> <li>17.encoding</li> <li>18.computability</li> <li>19.portability</li> <li>20.reliability</li> <li>21.scalability</li> <li>22.option</li> <li>23.compiler</li> <li>24.debugger</li> <li>25.disassembler</li> <li>26.procedure</li> <li>27.applicat-ion, apps</li> <li>28. recursion</li> <li>29.computer</li> <li>30.program-mer</li> <li>31.translator</li> <li>32.antivirus</li> <li>33.interface</li> <li>34.microchip</li> <li>35.micro-circuit</li> <li>36.micro-computer</li> <li>37.micropro-cessor</li> <li>38.pseudo-code</li> <li>39.submenu</li> <li>40.sub-program</li> <li>41.subroutine</li> <li>42.subscript</li> <li>43.super-script</li> </ol> | <ol style="list-style-type: none"> <li>1. adisyon</li> <li>2.kumbersyon</li> <li>3. desisyon</li> <li>4. dibisyon</li> <li>5. ekspresyon</li> <li>6. kompyuter</li> <li>7. printer</li> </ol> | <ol style="list-style-type: none"> <li>1. deklarasyon</li> <li>2. impormasyon</li> <li>3.manipulasyon</li> <li>4.multiplikasyon</li> <li>5. seguridad</li> <li>6. aplikasyon</li> </ol> | <ol style="list-style-type: none"> <li>1.pag-compile</li> <li>2. pagmodelo</li> <li>3. pag-testing</li> <li>4.nagpo-program</li> </ol> | <ol style="list-style-type: none"> <li>1. pag-uulit sa sarili</li> </ol> |                         |

|  |  |  |   |   |   |  |   |
|--|--|--|---|---|---|--|---|
| <p><b>Simple Root Word</b></p> <ol style="list-style-type: none"> <li>1.pixel</li> <li>2.port</li> <li>3.set</li> <li>4.terminal</li> <li>5.window</li> <li>6.byte</li> <li>7.console</li> <li>8.diskette</li> <li>9.mouse</li> <li>10.feature</li> <li>11.stack</li> <li>12.account</li> <li>13.address</li> <li>14.array</li> <li>15.bus</li> <li>16.complement</li> <li>17.constant</li> <li>18.entity</li> <li>19.error</li> <li>20.function</li> <li>21.loop</li> <li>22.prompt</li> <li>23.register</li> <li>24.screen</li> <li>25.solution</li> <li>26.variable</li> <li>27.bold</li> <li>28.element</li> <li>29.result</li> <li>30.design</li> <li>31.hierarchy</li> <li>32.condition</li> <li>33.data</li> <li>34.process</li> <li>35.speed</li> <li>36.symbol</li> <li>37.user</li> <li>38.version</li> <li>39.algorithm</li> <li>40.program</li> <li>41.file</li> <li>42.image</li> <li>43.technology</li> <li>44.bug</li> <li>45.character</li> <li>46.memory</li> <li>47.syntax</li> <li>48.system</li> </ol> |  | <ol style="list-style-type: none"> <li>1. pagpapating-kad</li> <li>2. bilis</li> <li>3. gumagamit</li> <li>4. talaan</li> <li>5. kamalian</li> <li>6. titik</li> </ol> | <ol style="list-style-type: none"> <li>1.pixel</li> <li>2.port</li> <li>3.set</li> <li>4.terminal</li> <li>5.window</li> <li>6.byte</li> <li>7.console</li> <li>8.diskette</li> <li>9.mouse</li> <li>10.feature</li> <li>11.stack</li> <li>12.account</li> <li>13.address</li> <li>14.array</li> <li>15.bus</li> <li>16.comple-ment</li> <li>17.constant</li> <li>18.entity</li> <li>19.error</li> <li>20.function</li> <li>21.loop</li> <li>22.prompt</li> <li>23.register</li> <li>24.screen</li> <li>25.solution</li> <li>26.variable</li> <li>27.algo-rithm</li> <li>28.program</li> <li>29.file</li> <li>30.image</li> <li>31.techno-logy</li> <li>32.bug</li> <li>33.charac-ter</li> <li>34.memory</li> <li>35.syntax</li> <li>36.system</li> </ol> | <ol style="list-style-type: none"> <li>1.kondisyon</li> <li>2.bersiyon</li> </ol> | <ol style="list-style-type: none"> <li>1.elemento</li> <li>2.resulta</li> <li>3.disenyo</li> <li>4.herarkiya</li> <li>5.datos</li> <li>6.proseso</li> <li>7.simbolo</li> <li>8.algoritmo</li> <li>9.programa</li> <li>10.imahe(n)</li> <li>11.teknolohiya</li> <li>12.memorya</li> <li>13.sintaktiko</li> <li>14.sistema</li> </ol> |  |   |
| <p><b>Compounding</b></p> <ol style="list-style-type: none"> <li>1.flowcharting</li> <li>2.firmware</li> <li>3.hardware</li> <li>4.keyboard</li> <li>5.smartphone</li> <li>6.sourcefile</li> <li>7.spreadsheet</li> <li>8.textfile</li> <li>9.joystick</li> <li>10.laptop</li> <li>11.desktop</li> <li>12,password</li> <li>13.bluetooth</li> <li>14.software</li> <li>15.motherboard</li> <li>16.underline</li> <li>17.flowchart</li> <li>18.database</li> <li>19.website</li> </ol>  |  | <ol style="list-style-type: none"> <li>1.salungguhit</li> <li>2.daluyang linya</li> </ol>  | <ol style="list-style-type: none"> <li>1.flowchart-ing</li> <li>2.firmware,</li> <li>3.hardware,</li> <li>4.keyboard</li> <li>5.smart-phone,</li> <li>6.sourcefile</li> <li>7.spread-sheet</li> <li>8.textfile</li> <li>9.joystick</li> <li>10.laptop</li> <li>11.desktop</li> <li>12.pass-word</li> <li>13.blue-tooth</li> <li>14.software</li> <li>15.mother-board</li> <li>17.flow-chart</li> <li>18.database</li> <li>19.website</li> </ol>   | <ol style="list-style-type: none"> <li>1.websayt</li> </ol>                       |   |  | <ol style="list-style-type: none"> <li>1.tipunan ng ulat</li> </ol> |

|                    |  |  |  |  |  |  |  |
|--------------------|--|--|--|--|--|--|--|
|                    | <p><b>Blending</b><br/>                 1.modem<br/>                 2.internet<br/>                 3.wifi<br/>                 4.soft-eng</p>  |  | <p>1.modem<br/>                 2.internet<br/>                 3.wifi<br/>                 4.soft-eng</p>   |  |  |  |  |
| Brand or Trademark | <p><b>Simple Root</b><br/>                 1.Apple<br/>                 2.Google<br/>                 3.iOS<br/>                 4.Lotus<br/>                 5.Macintosh<br/>                 6.Unix<br/>                 7.Windows<br/>                 8.Yosemite</p>   |  | <p>1.Apple,<br/>                 2.Google,<br/>                 3.iOS,<br/>                 4.Lotus,<br/>                 5.Macintosh,<br/>                 6.Unix, 7.Windows,<br/>                 8.Yosemite</p>   |  |  |  |  |
|                    | <p><b>Compounding</b><br/>                 1.Facebook<br/>                 2.Microsoft<br/>                 3.PowerPoint<br/>                 4.Wordperfect<br/>                 5.Wordstar</p>  |  | <p>1.Facebook<br/>                 2.Microsoft<br/>                 3.PowerPoint<br/>                 4.Wordperfect<br/>                 5.Wordstar</p>  |  |  |  |  |
|                    | <p><b>Blending</b><br/>                 1.Fortran<br/>                 2.Prolog</p>  |  | <p>1.Fortran<br/>                 2.Prolog</p>   |  |  |  |  |
| Keyword            | <p><b>Clipping</b><br/>                 1.div (division)<br/>                 2.mod (modulus division)<br/>                 3.xor (exclusive OR)<br/>                 4.ABS ( ) (absolute value function)<br/>                 5.chdir (change directory)<br/>                 6.char (character (programming key-word))<br/>                 7.del (delete)<br/>                 8.dir (directory)<br/>                 9.cls (clearscreen)<br/>                 10.mkdr (make directory)<br/>                 11. LEN ( ) (get character length)<br/>                 12. MID\$ ( ) (get middle character)<br/>                 13. SQR ( ) (get squareroot)<br/>                 14. ALT (alternate)<br/>                 15.ESC (escape)</p> |  | <p>1.div<br/>                 2.mod<br/>                 3.xor<br/>                 4.ABS ( )<br/>                 5.chdir<br/>                 6.char<br/>                 7.del<br/>                 8.dir<br/>                 9.cls<br/>                 10.mkdr<br/>                 11.LEN ( )<br/>                 12.MID\$ ( )<br/>                 13.SQR ( )<br/>                 14.ALT<br/>                 15.ESC</p> |  |  |  |  |

Table 9. Translation techniques in adjectives

| Lexical Type:<br><b>Simple</b>            | Back Translated<br>Terms in Com-<br>puter Studies   | Translation Techniques                    |  |              |  |  | Lexical<br>Convergence   | Semantic<br>Translation |
|---|---|---|--|--------------|--|--|--|-------------------------|
|   |   | Purification/<br>Am-<br>bag-Dalisay       | Systematic Borrowing   |              |  |  |  |                         |
|   |   |   | Direct   | Nativization |  |  |  |                         |
| Lexical Cate-<br>gory<br><b>ADJECTIVE</b> |   |   | Full Borrowing   | Anglicism    | Hispanism  |  |  |                         |
| Adjective                                 | <b>Derivation</b><br>1.nested<br>2.automatic<br>3.cryptic<br>4.installable<br>5.portable<br>6.global<br>7.hexadecimal<br>8.nonvolatile<br>9.mono-chromatic<br>10.multi-processor<br>11.automated<br>12.authorized<br>13.systematic<br>14.effective<br>15.reliable<br>16.retrieveable,<br>17.electronic,<br>18.recursive | 1.maaasahan<br>2.nababawi                 | 1.nested<br>2.automatic<br>3.cryptic<br>4.installable<br>5.portable<br>6.global<br>7.hexadecimal<br>8.nonvolatile<br>9.mono-chromatic<br>10.multi-processor<br>17.electronic<br>18.recursive | 1.awtomeyted | 1.awtorisado<br>2.sistematiko<br>3.epektibo<br>4.elektroniko |  |  |                         |
|   | <b>Simple Root Word</b><br>1.binary<br>2.boolean<br>3.character<br>4.decimal<br>5.integer<br>6.octal<br>7.real<br>8.string<br>9.compatible<br>10.volatile<br>11.efficient<br>12.false<br>13.true<br>14.batch<br>15.local  | 1.mahusay<br>2.mali<br>3.tama<br>4.tumpok | 1.binary<br>2.boolean<br>3.character<br>4.decimal 5.integer<br>6.octal<br>7.real<br>8.string<br>9.compatible<br>10.volatile<br>14.batch  |              |  |  | 1.di-global  |                         |
|   | <b>Compounding</b><br>1.floating-point<br>2.touchscreen<br>3.user-friendly<br>4. realtime<br>5. timesharing   |   | 1.floating-point<br>2.touchscreen,<br>3.user-friendly<br>4. realtime<br>5.timesharing  |              |  |  | 1.walang<br>takdang pag-<br>gamit<br>2.salo-salong<br>paggamit |                         |

**Table 10. Translation techniques in verbs**

| Lexical Type:<br><b>Simple</b>       | Back Translated<br>Terms in Computer<br>Studies   | Translation Techniques  |                      |              |              |  | Lexical Conver-<br>gence | Semantic<br>Transla-<br>tion |
|--------------------------------------|---|---|----------------------|--------------|--------------|--|--------------------------|------------------------------|
|                                      |   | Purifica-<br>tion/Ambag-D<br>alisy  | Systematic Borrowing |              |              |  |                          |                              |
|                                      |   |   | Direct               | Nativisation |              |  |                          |                              |
| Lexical Cate-<br>gory<br><b>VERB</b> |   |   | Full Borrowing       | Anglicism    | Hispanism    |  |                          |                              |
| Verbs                                | <b>Derivation</b><br>1.customize<br>2.initiaialize<br>3.debug<br>4.decode<br>5.unload<br>6.reactivate<br>7.programming  |   | 1.programming        |              |              | 1.i-customize<br>2.i-initialize<br>3.i-debug<br>4.i-decode<br>5.i-unload<br>6.i-reactivate<br>7.nagpo-program  |                          |                              |
|                                      | <b>Simple Root Word</b><br>1.add<br>2.boot<br>3.compare<br>4.merge<br>5.delete<br>6.document<br>7.edit<br>8.evaluate<br>9.execute<br>10.load<br>11.retrieve<br>12.copy<br>13.print<br>14.scan<br>15.install<br>16.compile<br>17. save<br>18.show<br>19.store<br>20.declare<br>21.search | 1.magdagdag<br>2.mapag-sama<br>3.magbawas<br>4.bawiin<br>5.kopyahin<br>6.ilimbag<br>7.pagsama-hin<br>8.itago<br>9.magsiwalat<br>10.iniiimbak<br>11.magtug-ma<br>12.maghanap | 1.copy<br>2.scan     |              | 1.idineklara | 1.mag-boot<br>2.ikokompara<br>3.i-dokumento<br>4.ma-edit<br>5.ma-evaluate<br>6.mag-execute<br>7.mag-load<br>8.i-print, mag-print,<br>mag-imprenta<br>9.i-install<br>10.i-compile<br>11. mag-save, i-save |                          |                              |
|                                      | <b>Compounding</b><br>1.download<br>2.overwrite<br>3.update<br>4.input<br>5.upload  |   | 1.input              |              |              | 1.mag-download<br>2.i-overwrite<br>3.mag-update<br>4.mag-input<br>5.i-upload,<br>mag-upload  |                          |                              |

Complex terms that are in the forms of acronyms or non-acronyms were not analyzed based on the individual word components, because each component word can go through similar analysis as the simple terms, but according to all their component words and the grammatical construction of the complex terms. Both forms underwent either borrowing by owning or by borrowing by adoption. Borrowing by owning is similar to the full borrowing in translating simple terms. In here, all the component words as well as the grammatical construction of term are retained. Examples are *read only memory* or ROM (acronym), and *procedure declaration* (non-acronym). Borrowing by adoption is a process where one or more of the component words were translated using the simple term translation techniques, and/or they have been retained but underwent grammatical shift. Full adoption retains the grammatical structure, while partial adoption does not. Examples of full adoption are *systems analysis and design* (SAD) which can be translated as *system analisis at desayn* (acronym), and *algorithmic design* translated as *disenyong al-goritmiko* (non-acronym).

Both acronyms and non-acronyms were translated by borrowing by owning. This may be attributed to the easy process, and this is essentially practical and efficient especially for acronyms, because it is good to retain the acronym form of the term to maintain its recall power. This is in contrast with non-acronyms as they can be better translated by borrowing by adoption. However efficient it is to retain the acronym of the terms, there were several complex terms which acronyms were not retained in their translation. Some examples of this case include *local area network* or LAN (*network ng lokal na lugar*), *human-computer interaction* or HCI (*ugnayang tao-kompyuter*), and *data flow diagram* or DFD (*daloy ng datos na naka-diagram*). In the case of non-acronyms, the most evident translation technique for terms translated by borrowing by adoption is the partial adoption or the process that shifts grammatical construction.

## 5. Conclusion and Recommendation

The study found out that the translation from English of computer studies terms used in Filipino discourses did not reveal a pattern based on the lexical characteristics of the terms. This implies that translators, writers, and users of Filipino language in the field of computer studies have no consistencies among each other on the principles they followed in translating these terms. Thus, it can be said that there is no existing standard method of translation as revealed by the data being studied. However, using the lexical analysis conducted in this study, it is foreseen that a lexical, structure-based translation principles and model can be developed so that consistency or standards can be established in translating computer studies terms. Having term translation standardized in computer studies, this study may be significant in improving bilingual discourses and materials development in Philippine education that are perceived to be helpful in teaching and learning.

This study can be further explored considering the following recommendations:

- 1) Refine the lexical analysis and derive translation principles that can be applied in computer studies. The data can be used to determine commonalities among terms that will be basis for deciding what term translation is most suited.
- 2) When principles are already in place, an algorithm can be formulated that can be used in improving machine translation.
- 3) The analysis can be replicated in different fields that will help establish principles in technical translation.

## References

- Airbus defense and space leader in radar technology in Germany*. (2015). Retrieved May 15, 2016, from <http://airbusdefenseandspace.com>.
- Almario, V. S. (1997). *Tradition and Filipino language*. Quezon City, Philippines: UP, Center for Filipino Language.
- Alviar, S. M. and Roxas, R. (2000). *Introduction to computer science*. Quezon City, Philippines: UP, Center for Filipino Language.
- Batnag, A. E., Petras, J., and Antonio, L. F. (2009). *Textbook in translation*. Quezon City, Philippines: C & E Publishing, Inc.
- Benson, C. (2004). *The importance of mother tongue-based schooling for educational quality*. United Nations Educational and Scientific and Cultural Organization. Retrieved April 15, 2020, from <http://www.un.org>.
- Bryman, A. (2010). *triangulation and measurement*. Retrieved April 15, 2020, from <http://google.com>.
- David, A. (2015). *3 reasons why South Korea will always be the world leader in mobile technology*. Retrieved April 30, 2020, from <http://www.thestreet.com>.
- De Castro, I. P. (2008). Bilingual dictionary of computer terms: A need for the intellectualization of Filipino language. *Unitas Journal*, 81(1), 167-180.
- Del Rosario, G. (1969). *Maugnayin science wordlist (English-Pilipino)*. Malabon: Gregorio Araneta University Foundation.
- Ferguson, G. (2006). *Language planning and education*. United Kingdom: Edinburgh. University Press Ltd.
- Fortunato, T. F. (2003). The Filipino language in the bilingual education policy. *SangFil Sourcebook*, 1, 78-83.
- Gonzalez, A. B. (2005). Language planning and intellectualization. *SangFil Sourcebook*, 2, 232-243.

- Industry overview: the automotive industry in Germany.* (2016). Retrieved May 15, 2020, from <http://www.qtai.de>.
- Kopytoff, V. (2011). *Japanese 'K' computer is ranked most powerful.* Retrieved April 15, 2020, from <http://www.nytimes.com>.
- KWF. (2016, February 26-28). Intellectualization of the Filipino language in various disciplines. Manila: Komisyon sa Wikang Filipino.
- Lacuesta, G. (1967). *Filipino versus Pilipino.* Quezon City, Philippines: Katas Magazine.
- Macatangay, J. E. and Macatangay, L. (1998). *Introduction to computer organization and low-level programming.* Quezon City, Philippines: UP, Center for Filipino Language.
- Madrid, A. M. (2005). *Translating the industrial automation module into Filipino: towards the modernization of the Filipino language.* Unpublished doctoral dissertation, De La Salle University, Manila.
- Paz, C. J. (1995). *The Filipino language is ours.* Quezon City, Philippines: UP, Center for Filipino language.
- Paz, C. J., Hernandez, V., and Peneyra, I. U. (2010). *The study of language.* Diliman, Quezon City: University of the Philippines Press.
- Ramos, J. F. (2000). *Eclectic approach in the elaboration of mathematics register: towards a theory in language planning.* Unpublished doctoral dissertation, University of the Philippines, Diliman.
- Ravina, E. A. (2017). *Lexilaboration: Towards a model in translating computer studies terms in Filipino.* Unpublished doctoral dissertation, University of the Philippines, Diliman.
- Russia remains world leader for space launches.* (2015). Retrieved May 15, 2020, from <http://www.spacedaily.com>.
- Santiago, A. O. (1984). *The technical lexicon of Pilipino.* Unpublished doctoral dissertation, Philippine Normal University, Manila.
- Santiago, A. O. and Tiangco, N. G. (2003). *Modern Filipino grammar.* Quezon City, Philippines: Rex Bookstore.
- Santos, L. (1951). *Modern grammar: Critic and recommendations of Cirio H. Panganiban of the Institute of National Language.* Maynila: Institute of National Language.
- Sevilla III, F. B. (2001). Using Filipino in the sciences. *Daluyan*, 10 (1-4), 95-101.
- Sicat, C. P., Zafra, B., and Gernimo, J. V. (2017). *Talaban: Communication, reading, and research in the Filipino.* Quezon City, Philippines. Rex Bookstore.
- Sicat, C. P. (2019). *Reading and analysis of texts towards research.* Quezon City, Philippines. Rex Bookstore.
- South Korea emerging as medical device leader.* (2013). Retrieved March 18, 2020, from <http://www.qmed.com>.
- Sytangco, J. (1977). *Maunlarin science wordlist: English-Pilipino.* Manila: UST Research Center.
- Tauli, V. (1968). *Introduction to a theory of language planning.* Uppsala: Almqvist and Wiksell.
- Yule, G. (2003). *The study of language.* United Kingdom: Cambridge University Press.
- Zafra, G. S. (2006). Translating technological knowledge. *Daluyan*, 13 (1-2), 29-43.